باسمه‌تعالی

وزارت علوم، تحقیقات و فناوری
پژوهشگاه علوم و فناوری اطلاعات ایران

طرح پژوهشی

به‌کارگیری فنون و ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ‌مقیاس

مجری

گلناز وکیلی

بهار ۱۳۹۳

فهرست مطالب

عنوان **شماره صفحه**

حقوق: پژوهشگاه علوم و فناوری اطلاعات ایران

طرح پژوهشی «به‌کارگیری فنون و ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ مقیاس» پیرو قرارداد شماره ۳۳/۱۳۸۱ مورخ ۱۳۹۲/۲/۲ میان پژوهشگاه علوم و فناوری اطلاعات ایران (کارفرما) و خانم دکتر گلناز وکیلی (مجری) اجرا شده است. گزارش حاضر مستند حاصل از این طرح است.

این گزارش و تمامی حقوق مادی آن بر اساس «قانون حمایت حقوق مولفان و مصنفان و هنرمندان، مصوب سال ۱۳۸۴ و اصلاحیه‌های بعدی آن و همچنین آیین‌نامه‌های اجرایی این قانون» متعلق به پژوهشگاه علوم و فناوری اطلاعات ایران است و هر گونه استفاده از تمامی یا پاره‌ای از آن، شامل: نقل قول، تکثیر، انتشار، کاربرد نتایج، تکمیل و مانند آن‌ها به صورت چاپی، الکترونیکی یا وسایل دیگر فقط با اجازه کتبی پژوهشگاه امکان‌پذیر است. نقل قول در حد هزار واژه در انتشارات علمی مانند کتاب و مقاله با درج اسلاعات کامل کتاب‌شناختی، نیازی به مجوز پژوهشگاه ندارد.

صحت مندرجات گزارش بر عهده مجری طرح پژوهشی است.

چکیده

با توجه به گسترش روزافزون بسترهای رایانشی بزرگ مقیاس، در آینده‌ای نزدیک بسیاری از خدمات رایانشی پژوهشگاه علوم و فن‌آوری ایران، مانند «ارایه‌ی طرح‌ها و پایان‌نامه‌های الکترونیکی»، «آموزش و یادگیری الکترونیکی» و «انجام محاسبات علمی و پربازده»، تنها با ارایه بر فراز چنین بسترهایی می‌توانند پاسخ‌گوی نیازمندی‌های روزافزون کاربران خود با کیفیت مطلوب باشند. با در نظر گرفتن این نیازمندی، در این طرح پژوهشی قصد داریم به چگونگی به‌کارگیری فنون و ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ مقیاس، به عنوان نخستین گام برای دست‌یابی به این هدف نهایی که تحقق پیاده‌سازی انواع خدمات رایانشی پژوهشگاه بر اساس فن‌آوری‌های رایانشی بزرگ مقیاس است بپردازیم. به این منظور پس از بررسی فنون و ابزارهای شبیه‌سازی بزرگ‌مقیاس، به دسته‌بندی کلی انواع این فنون و ابزارها می‌پردازیم. در ادامه گزارش پس از شناسایی انواع نیازمندی‌های مختلف برای استفاده از فنون و ابزارهای هر دسته، با تمرکز بر روی حوزه‌ی آموزش و یادگیری الکترونیکی، به چگونگی به‌کارگیری این فنون و ابزارها در برخی سناریوهای متداول در این حوزه می‌پردازیم.

فهرست جدول‌ها

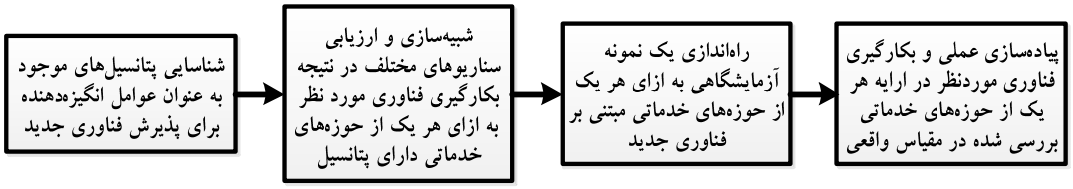
فهرست شکل‌ها

۱- مقدمه

با توجه به اهداف و ماموریت‌های اصلی پژوهشگاه علوم و فن‌آوری اطلاعات ایران، بر اساس بررسی-های انجام گرفته در طرح پژوهشی پیشین، فعالیت‌های این پژوهشگاه در حوزه‌هایی مانند «ارایه‌ی طرح‌ها و پایان‌نامه‌های الکترونیکی»، «آموزش و یادگیری الکترونیکی» و «انجام محاسبات علمی و پربازده»، پتانسیل بالایی برای گسترش بر روی بسترهای رایانشی بزرگ‌مقیاس را دارند. باید در نظر داشت که توسعه‌ی فناورانه‌ی فعالیت‌هایی از این دست، یا به صورت آشکارا و یا به صورت پس-زمینه‌ای، مبتنی بر حجم بالایی از عملیات رایانشی است. انجام این «عملیات رایانشی بزرگ‌مقیاس» اغلب به میزان زیادی از منابع پردازشی و ذخیره‌سازی و همچنین به پهنای باند بالای ارتباطی نیاز دارد. تامین این منابع پردازشی، ذخیره‌سازی و ارتباطی برای گسترش فعالیت‌ها و خدماتی که دارای پتانسیل‌های رایانشی هستند، صرف هزینه‌های بالایی را به همراه خواهد داشت. محدودیت تامین بودجه از یک‌سو و در اختیار نداشتن تخمین مناسبی از منابع مورد نیاز از سوی دیگر، مساله‌ی تامین منابع لازم را دو چندان مشکل می‌سازد. روی‌کرد منطقی برای پرداختن به چالش‌هایی از این دست، به‌کارگیری ابزارها و فنون شبیه‌سازی در انجام عملیات رایانشی بزرگ‌مقیاس است که در این طرح به چگونگی آن پرداخته خواهد شد.

به‌کارگیری فنون و ابزارهای مدل‌سازی و شبیه‌سازی، سابقه‌ای طولانی در توسعه‌ی الگوریتم‌ها و فناوری‌های جدید دارد و همواره به عنوان راه‌کاری که امکان گسترش سیستم‌های توزیع‌یافته و بزرگ مقیاس را فراهم می‌آورد، مورد توجه قرار داشته است. این فنون و ابزارها به خصوص در برخورد با مسایلی که ارزیابی تحلیلی آن‌ها، به علت ماهیت خاصی که دارند، با محدودیت منابع روبه-رو است از اهمیت بالایی برخوردارند. از این‌رو در سال‌های اخیر، تعداد و پیچیدگی شبیه‌سازهایی که در حوزه‌های علمی و صنعتی مورداستفاده قرار گرفته‌اند، افزایش چشمگیری داشته است. این افزایشِ به‌کارگیری، تنها محدود به دانشگاه‌ها و پژوهشگاه‌ها نبوده است، و شرکت‌ها وکسب‌وکارهای مختلف نیز استفاده از فنون و ابزارهای مدل‌سازی و شبیه‌سازی را در برنامه‌ی کاری خود قرار داده‌اند. در یک نگاه کلی‌تر، انتظار می‌رود که هر مجموعه‌ی دانشگاهی یا صنعتی در پذیرش و به‌کارگیری فناوری‌های جدید در ارایه‌ی خدمات و انجام فعالیت‌های خود، مسیر مشخصی را دنبال نماید. این مسیر با مطرح شدن ایده‌ی به‌کارگیری فناوری مورد نظر آغاز می‌شود و در صورتی که به درستی

طراحی و دنبال شود، در انتها منجر به تحقق و پیاده‌سازی عملی ایده‌ی اولیه و به‌کارگیری موفق آن فناوری خواهد شد. طرح کلی نقشه‌ی چنین مسیری در شکل ۱ نشان داده شده است. با در نظر گرفتن این نقشه‌ی کلی، اهمیت فنون و ابزارهای مدل‌سازی و شبیه‌سازی برای پرداختن به مساله‌ای که در گام دوم این مسیر کلی طرح شده است، به خوبی مشخص می‌شود. به بیان دقیق‌تر، انتخاب بسترهای رایانشی بزرگ‌مقیاس به عنوان زیرساختی برای گسترش و ارایه‌ی فناورانه‌ی خدمات، یکی از اهداف بزرگی است که تحقق آن نیازمند صرف هزینه‌های کلان است و از این‌رو لازم است تا با به‌کارگیری مناسب فنون و ابزارهای شبیه‌سازی تا حد امکان از درست پیمودن مسیری که به این هدف منتهی می‌شود، اطمینان پیدا کرد.



شکل ۱- اهمیت فنون و ابزارهای مدل‌سازی و شبیه‌سازی در به‌کارگیری فناوری‌های رایانشی بزرگ‌مقیاس

با توجه به این‌که انجام عملیات رایانشی بزرگ مقیاس، اغلب به حجم قابل ملاحظه‌ای از منابع پردازشی و ذخیره‌سازی نیازمند است، برای رویارویی با مساله‌ی افزایش روبه‌رشد تقاضاها برای دریافت توان پردازشی، نخستین روی‌کرد استفاده از اَبَرکامپیوترها است. برای نخستین بار در حدود دهه‌ی ۷۰ میلادی، ابرکامپیوتری که توسط شرکت «سی‌بی‌سی»[1] طراحی و ساخته شده بود، در دسترس متقاضیان قرار گرفت که بیشینه‌ی بازده آن در حدود ۳ در ۱۰ به توان ۶ «فِلاپس»[2] (عملیات نقطه شناور بر ثانیه) بود. در سال ۲۰۰۸ میلادی نیز شرکت آی‌بی‌اِم محصولی با بیشینه‌ی بازدهی بیشتر از ۱۰ به توان ۱۵ فِلاپس را برای سرویس‌دهی ارایه کرد که تا مدت‌ها به عنوان یکی از برترین ابرکامپیوترها باقی ماند[۳].

امروزه نیز ابرکامپیوترها همچنان به منظور اجرای شبیه‌سازی‌های پیچیده در مدت زمان قابل قبول مورد استفاده قرار می‌گیرند، ولی مدت‌هاست که دیگر توان پاسخ‌گویی به تقاضاهای روزافزون منابع پردازشی را در بسیاری از مسایل ندارند. البته باید توجه داشت که هزینه‌های بسیار بالایی که برای تهیه

---

[1] *CBC (Control Data Corporation)*
[2] *flops (floating point operations per second)*

و نگهداری این سیستم‌ها مورد نیاز است، همواره یکی از موانع اصلی در به‌کارگیری گسترده‌ی آن‌ها بوده است.

با قوی‌تر شدن سخت‌افزار (افزایش ظرفیت منابع محاسباتی، ذخیره‌سازی و شبکه‌ای) سیستم‌ها و ماشین‌های پردازشی متداولی که با هزینه‌های به نسبت پایین در دسترس عموم قرار گرفت، پژوهش-گران و موسسات پژوهشی که توان استفاده از ابرکامپیوترها را نداشتند به سمت خوشه‌بندی این ماشین‌ها روی آوردند تا به این طریق نیازهای پردازشی خود را برآورده سازند. در حال حاضر حتی در شرایطی که دسترسی به ابرکامپیوترها نیز امکان‌پذیر باشد، باز در بسیاری موارد استفاده از خوشه-های کامپیوتری ترجیح داده می‌شود، و اغلب تنها در مواردی که به دلیل نیازمندی‌های خاص نیاز به استفاده از ابرکامپیوترها باشد، عملیات پردازشی بر روی آن‌ها اجرا می‌گردد.

در صورت استفاده از خوشه‌های کامپیوتری به منظور پردازش و ذخیره‌سازی، باید علاوه بر تقاضاهای کنونی، پتانسیل افزایش حجم تقاضاها در آینده را نیز در نظر گرفت. از این‌رو معمولا خوشه‌های کامپیوتری برای حالت بیشینه‌ی بار تامین منابع می‌شوند. در نتیجه در بیشتر مواقع، منابع آن‌ها کم‌تر از بیشینه‌ی ظرفیت خود مورد بهره‌برداری قرار می‌گیرند.

بسیاری از تامین‌کنندگان منابع چه در صنعت و چه در دانشگاه، برای افزایش بهره‌وری منابع خود، ظرفیت آزاد آن‌ها را برای استفاده‌ی دیگران در دسترس قرار می‌دهند. به این منظور لازم است تا امکان دسترسی محدود و از راه‌دور مصرف‌کنندگان به منابع محلی فراهم‌کنندگان ایجاد شود. رایانش توری و رایانش ابری از جمله فناوری‌هایی هستند که به این منظور مورد استفاده قرار می‌گیرند. مفهوم رایانش توری از تحقیات دانشگاهی در سال ۱۹۹۰ نشات می‌گیرد [۱]. در شبکه‌ی توری منابع مربوط به چندین دامنه‌ی مدیریتی در قالب یک زیر ساخت یا محیط پردازشی به اشتراک گذاشته می‌شوند. به بیان دقیق‌تر، می‌توان شبکه‌ی توری را به عنوان سیستم توزیع‌یافته‌ای از منابع در نظر گرفت که این منابع در مکان‌های متفاوتی قرار گرفته‌اند و در مجموع به پردازش حجم بالایی از بارکاری می-پردازند. ویژگی مشخص رایانش توری که آن را از سیستم‌های متداول رایانش پربازده متمایز می‌سازد این است که در شبکه‌ی توری منابع ناهم‌گون هستند، با آزادی بیشتری به هم پیوند خورده‌اند[۳] و از توزیع و پراکندگی جغرافیایی برخوردارند [۲۸].

در مقابل، فراهم‌کنندگان صنعتی در سال‌های اخیر منشا پیدایش رایانش ابری بوده‌اند. تاکید مفهوم ابر بر تامین دسترسی آسان به منابعی است که تحت مالکیت یک فراهم‌کننده‌ی مشخص قرار دارند [۲]. ویژگی‌های اصلی و خاصی که در رایانش ابری مطرح می‌شود عبارتند از[۲۹]:

---

[3] *Loosely coupled*

- دسترسی به منابع محاسباتی نامحدود در صورت نیاز: این جنبه، کاربران رایانش ابری را از طرح‌ریزی مبتنی بر پیش‌بینی برای تامین منابع بی‌نیاز می‌سازد.
- حذف از پیش عضو شدن کاربران ابر: به این ترتیب شرکت‌ها می‌توانند با منابع سخت‌افزاری کمی شروع به‌کار کنند و تنها با افزایش نیاز خود به میزان این منابع بیفزایند.
- امکان پرداخت هزینه‌ی استفاده از منابع محاسباتی بر مبنای میزان نیاز و همچنین آزادسازی منابع در صورت عدم نیاز : به این ترتیب می‌توان برای مثال هزینه‌ی منابع پردازشی را بر حسب مقدار ساعت استفاده پرداخت و هنگامی که استفاده‌ای ندارند با رهاسازی آن‌ها صرفه‌جویی نمود.

به طور کلی ساخت و به‌کاراندازی مرکزهای داده بسیار بزرگ در مکان‌های کم‌هزینه، یکی از عوامل اصلی توسعه‌ی رایانش ابری بوده است.

در حال حاضر رایانش توری و رایانش ابری از جدیدترین و اصلی‌ترین فناوری‌های مطرح برای انجام عملیات رایانشی بزرگ مقیاس به‌شمار می‌روند [۳]. از این‌رو، همان‌گونه که در بخش بعد بررسی خواهیم کرد، بسیاری از فنون و ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ مقیاس، بر مبنای شبیه-سازی محیط‌های مبتنی بر این دو فناوری توسعه داده شده‌اند.

در ادامه‌ی این گزارش قصد داریم تا ابتدا در بخش ۲ به بررسی فنون و ابزارهای مطرح در عملیات رایانشی بزرگ مقیاس بپردازیم. سپس در بخش ۳ ویژگی‌های کلی این فنون و ابزارها را استخراج و بر اساس نحوه‌ی به‌کارگیری، یک دسته‌بندی کلی از این فنون و ابزارها ارایه می‌نماییم. در بخش ۴ و ۵ با تمرکز بر روی حوزه‌ی آموزش و یادگیری الکترونیکی به عنوان یکی از حوزه‌های خدمات رایانشی بزرگ‌مقیاس، ابتدا انواع نیازمندی‌های مختلف برای استفاده از فنون و ابزارهای شبیه‌سازی در هر دسته را به تفکیک شناسایی می‌نماییم و سپس به تبیین چگونگی به‌کارگیری هر دسته ای این فنون و ابزارها در برخی از سناریوهای متداول و مطرح در حوزه‌ی موردنظر می‌پردازیم. در نهایت پس از جمع‌بندی گزارش، پیشنهادهایی را برای پژوهش‌های آتی ارایه می‌نماییم.

۲- فنون و ابزارهای شبیه‌سازی مناسب در عملیات رایانشی بزرگ مقیاس

امروزه، مدل‌سازی و شبیه‌سازی سیستم‌های رایانشی بزرگ مقیاس یکی از مراحل جاافتاده در حوزه‌ی تحقیق و توسعه‌ی کاربردها و الگوریتم‌ها به‌شمار می‌رود. با افزایش پیچیدگی چنین محیط‌هایی، فناوری‌های مربوط به شبیه‌سازی آن‌ها نیز تغییرات قابل ملاحظه‌ای را تجربه کرده است تا توان رویارویی با چالش‌هایی مانند «چندبستری»[۴] بودن محیط‌های توزیع‌یافته و یا چالش‌های دیگری از این دست را بیابد. برای مثال، انتظار می‌رود که شبیه‌سازهای پربازده و کارا توانایی بررسی سیستم در شرایط مختلف پیکربندی را داشته باشند، و همچنین بتوانند سناریوهای مربوط به کاربردهای متفاوت را قبل از این‌که در عمل و در سیستم‌های رایانشی بزرگ مقیاس واقعی اتفاق بیفتند، در محیط شبیه‌سازی ایجاد نمایند.

به طور کلی، فنون و ابزارهای شبیه‌سازی باید امکان مطالعه و ارزیابی روش‌ها و سیاست‌های توسعه داده شده‌ی موردنظر پژوهش‌گران را در شرایط تکرارپذیر و قابل کنترل فراهم کنند، به گونه‌ای که پژوهش‌گران بتوانند پیش از به‌کارگیری این روش‌ها و الگوریتم‌ها بر روی سیستم‌های عملیاتی به تنظیم کارایی آن‌ها بپردازند. علاوه بر این، ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ مقیاس باید توانایی مدل‌سازی مولفه‌های پردازشی گوناگون، و امکان ایجاد منابع، توپولوژی شبکه و کاربران را داشته باشند. همچنین باید امکان پیاده‌سازی انواع مختلف الگوریتم‌های زمان‌بندی و توزیع کار، مدیریت سیاست‌های قیمت‌گذاری، و تولید نتایج آماری مناسب در خروجی خود را داشته باشند.

در ادامه‌ی این بخش به برخی از فنون و ابزارهای شبیه‌سازی مطرح در حوزه‌ی عملیات رایانشی بزرگ مقیاس اشاره می‌کنیم. با توجه به گستردگی تنوع این فنون و ابزارها، به منظور مطالعه و بررسی دقیق‌تر آن‌ها، ابتدا به ارایه‌ی یک طبقه‌بندی کلی در این حوزه می‌پردازیم. سپس بر اساس طبقه‌بندی ارایه شده، نمونه‌های اصلی و مطرح در هر دسته را مورد بررسی قرار می‌دهیم. لازم به توضیح است که برخی از این شبیه‌سازها به دلیل این‌که در کاربرد گسترده‌ای یافته‌اند به نسبت از اهمیت بالاتری

---

[4] *multiplatform*

برخوردارند، از این‌رو در ادامه، این فنون و ابزارها با تاکید و تفصیل بیشتری مورد بحث قرار می-گیرند.

با توجه به تنوع کاربردهای فنون و ابزارهای شبیه‌سازی در حوزه‌های مختلف (به خصوص در حوزه-هایی که فرآیندها و سیستم‌های مورد ارزیابی از پیچیدگی بالایی برخوردارند)، مدل‌های طبقه‌بندی گوناگونی برای دسته‌بندی آن‌ها ارایه شده است [۳۰] ، [۳۱] و [۳۲]. برای نمونه در مرجع [۳۰] ابزارهای شبیه‌سازی بر اساس «حوزه‌ی کاربردی» آن‌ها به چهار دسته‌ی کلی تقسیم شده‌اند: ۱) فرآیندهای صنعتی، ۲) منابع محیطی، ۳) سیستم‌های موازی و توزیع‌یافته و ۴) سایر حوزه‌های کاربردی. سپس بر اساس این طبقه‌بندی، در ادامه‌ی کار این مرجع تاکید خود را بر روی سیستم‌های موازی و توزیع‌یافته قرار داده و به دسته‌بندی جزیی‌تر فنون و ابزارها در این حوزه‌ی کاربردی پرداخته است. همچنین مرجع [۳۱] بر اساس «نحوه‌ی اجرا»، ابزارهای شبیه‌سازی را به دو دسته‌ی ۱) مبتنی بر روی‌داد[5] و ۲) مبتنی بر موجودیت[6] تقسیم‌بندی می‌کند، که تفاوت این دو دسته در عاملی است که در اجرا و پیش‌روی فرآیند شبیه‌سازی نقش اصلی را بر عهده دارد (این عامل می‌تواند روی‌دادها و یا موجودیت‌های شبیه‌سازی شده در فرآیند باشد).

در این طرح پژوهشی، تاکید اصلی بر روی ابزارهای شبیه‌سازی است که به طور خاص در حوزه‌ی عملیات رایانشی بزرگ‌مقیاس به‌کار گرفته می‌شوند. از این‌رو ما ، فنون و ابزارهای شبیه‌سازی در این حوزه‌ی کاربردی را بر اساس «نحوه‌ی به‌کارگیری» به دو دسته‌ی کلی تقسیم می‌کنیم، ۱) فنون و ابزارهایی که بر مبنای شبیه‌سازی محیط‌های رایانشی بزرگ‌مقیاس توسعه داده شده‌اند و ۲) فنون و ابزارهای شبیه‌سازی که بر روی بسترهای رایانشی بزرگ‌مقیاس قابل به‌کارگیری هستند. به بیان دقیق‌تر دسته‌ی نخست در بر گیرنده‌ی فنون و ابزارهایی است که به شبیه‌سازی بسترهای رایانشی مناسب برای انجام عملیات رایانشی بزرگ‌مقیاس، مانند محیط‌های رایانش توری و رایانش ابری می‌پردازند (همان-طور که پیش‌تر بحث شد، بر اساس مطالعات انجام گرفته، رایانش توری و رایانش ابری از فناوری‌های اصلی و مطرح برای توسعه و انجام عملیات رایانشی بزرگ مقیاس به‌شمار می‌روند)، و دسته‌ی دوم ابزارهایی را در بر می‌گیرد، که به منظور انجام عملیات رایانشی بزرگ‌مقیاس، قابلیت به‌کارگیری و توسعه بر روی چنین بسترهایی را دارند و ما در این پژوهش این دسته را فنون وابزارهای شبیه‌سازی بزرگ‌مقیاس می‌نامیم. بر اساس این طبقه‌بندی در ادامه‌ی این بخش، با توجه به اهمیت نسبی فنون و ابزارهای دسته‌ی اول نسبت به دسته‌ی دوم، ابتدا در زیر بخش‌های ۲-۱ تا ۲-۷، نمونه‌هایی از این

---

[5] *Event-based*
[6] *Entity-based*

دسته مورد بررسی قرار می‌گیرند و سپس در زیر بخش ۲-۸ به تعدادی از فنون و ابزارهای دسته‌ی دوم پرداخته خواهد شد. لازم به تاکید است که فنون و ابزارهایی که در هر یک از این دو دسته قابل طرح و بررسی هستند از گستردگی بسیار بالایی برخوردارند، برای مثال [۴،۵،۶،۷،۹،۱۰،۱۱،۱۳،۱۴،۱۵، ۲۳،۲۲،۲۱،۲۰،۱۶،۳۷،۳۶،۳۵،۳۴] ، و ما در این بخش به تعداد محدودی از آن‌ها به عنوان نمونه‌هایی از هر دسته خواهیم پرداخت.

۲-۱- بریکس

بریکس[۷] [۴] یکی از اولین محیط‌های شبیه‌سازی است که برای بررسی مسایل مربوط به زمان‌بندی در شبکه‌های توری طراحی شده است. به بیان دقیق‌تر، بریکس به منظور مطالعه و مقایسه‌ی چارچوب‌ها و الگوریتم‌های زمان‌بندی در شرایط مختلف بارکاری و ساختار شبکه، و با هدف فراهم کردن نتایج تکرارپذیر طراحی شده است. پژوهش‌هایی که برای ارزیابی خود به استفاده از این ابزار می‌پردازند، بیشتر بر روی شبیه‌سازی سناریوهای شبکه‌های توری با چند سرور و چند کلاینت تاکید دارند.

بریکس از جمله شبیه‌سازهای مطرح در حوزه‌ی عملیات رایانشی بزرگ مقیاس است که امکان ارزیابی کارایی چارچوب‌ها و الگوریتم‌های زمان‌بندی در سیستم‌های پردازشی پربازده را فراهم می-آورد. بریکس امکان شبیه‌سازی شرایط و رفتارهای متفاوتی در سیستم مانند الگوریتم‌های زمان‌بندی و ماژول‌های برنامه‌نویسی مربوط به آن در سیستم، توپولوژی عمومی کلاینت‌ها و سرورها در سیستم پردازشی، و شماهای پردازشی سرورهای شبکه را ایجاد می‌نماید. این ابزار اساساً یک شبیه‌ساز «مبتنی بر رویداد»[۸] است که به عنوان چارچوب و مجموعه‌ای از مولفه‌های قابل جایگزین طراحی و پیاده-سازی شده است.

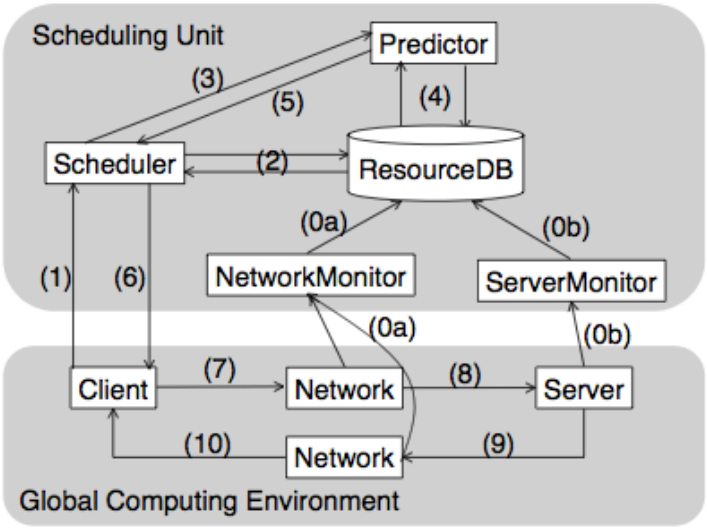
معماری بریکس از دو بخش سیستم پردازشی عمومی و واحد زمان‌بندی تشکیل یافته است که این دو بخش با یک‌دیگر تعامل دارند. سیستم پردازشی عمومی شامل تقاضاهای (کارهای) ارسال شده‌ی کاربران، سرورهایی که این تقاضاها را اجرا می‌کنند و شبکه‌ی ارتباطی آن‌هاست. مشخصاتی که برای سرورها و شبکه‌ها تعریف می‌شود، بازده، بار کاری و میزان واریانس آن‌ها در طول زمان است. در واقع سرورها و شبکه‌ها با سیستم‌های صف مدل‌سازی می‌شوند. تعریف تقاضاها (کارها) در سیستم با سایز پارامترها و نتایج خروجی آن‌ها و همچنین تعداد عملیات محاسباتی مورد نیاز آن‌ها مشخص می-

---

[7] *Bricks*
[8] *event-driven*

شود. بریکس امکان «وارد کردن»[9] سیستم‌های پردازشی عمومی واقعی و اعتبارسنجی آن‌ها را نیز فراهم می‌کند. شمای کلی معماری بریکس و مولفه‌های آن در شکل ۲ نشان داده شده است.

واحد زمان‌بندی این شبیه‌ساز، از یک مونیتور شبکه، یک مونیتور سرور، یک پایگاه داده‌ی منابع، یک ماژول پیش‌بینی، و یک ماژول زمان‌بندی تشکیل یافته است. مونیتور شبکه تاخیر و پهنای باند شبکه را اندازه‌گیری می‌کند. مونیتور سرور بازده، بار کاری و میزان دسترسی سرورها را اندازه‌گیری می‌کند. پایگاه داده‌ی منابع تمام نتایج اندازه‌گیری‌ها را در خود نگهداری می‌کند. ماژول پیش‌بینی با خواندن اطلاعات اندازه‌گیری شده، به پیش‌بینی میزان منابع در دسترس می‌پردازد، و در نهایت ماژول زمان-بندی بر اساس اطلاعات پایگاه داده‌ی منابع و ماژول پیش‌بینی، تخصیص کارها به سرورها را انجام می‌دهد.



شکل ۲- شمای کلی معماری بریکس و مولفه‌های آن [۴]

پیاده‌سازی بریکس با استفاده از جاوا انجام گرفته است. به این ترتیب که کاربرها می‌توانند انواع توپولوژی‌های شبکه، معماری‌های سرور، مدل‌های ارتباطات و مولفه‌های چارچوب زمان‌بندی را در محیط شبیه‌سازی مشخص کنند. علاوه بر این، همان‌طور که پیشتر به آن اشاره شد، مولفه‌ها نیز به نوبه-ی خود قابل جایگزین شدن هستند و این مساله بر انعطاف‌پذیری بریکس می‌افزاید. به طوری که این شبیه‌ساز در زمینه‌های گوناگونی از کاربردهای عملیات رایانشی بزرگ مقیاس مانند سرویس‌های پیش‌بینی هوا مورد استفاده قرار گرفته است.

---

[9] *import*

۲-۲- سیم‌گرید

سیم‌گرید[10] [۵،۶ و۷] یکی دیگر از شبیه‌سازهایی است که در حوزه‌ی عملیات رایانشی بزرگ مقیاس مورد استفاده قرار گرفته است. انگیزه‌ی طراحی این شبیه‌ساز، لزوم توسعه‌ی ابزارهایی برای مطالعه‌ی مسایل زمان‌بندی چندسروری در محیط‌های پیچیده، پویا، توزیع‌یافته و ناهمگون بوده است. از آن-جایی که پیچیدگی این مساله در فرم اصلی خود از مرتبه‌ی اِن‌پی[11] است، بیشتر الگوریتم‌های زمان-بندی که به عنوان راه‌حل مورد مطالعه قرار می‌گیرند از نوع هیوریستیک هستند. سیم‌گرید یک شبیه-ساز مبتنی بر رویداد است که با استفاده از زبان «سی»[12] طراحی شده است و مجموعه‌ای از توابع مختلف را برای شبیه‌سازی مشخصات کاربردها و زیرساخت‌های رایانشی فراهم می‌نماید.

سیم‌گرید کارکردهای هسته و اصلی برای ارزیابی الگوریتم‌های زمان‌بندی در انجام عملیات رایانشی بزرگ مقیاس متناظر با اجرای کاربردهای توزیع‌یافته در محیط رایانشی ناهمگون را فراهم می‌کند. هدف این شبیه‌ساز مدل‌سازی و تامین سطح مناسبی از انتزاعی‌سازی برای مطالعه‌ی الگوریتم‌های زمان‌بندی و در نهایت تولید نتایج دقیق به عنوان خروجی است. در این شبیه‌ساز، الگوریتم‌های زمان-بندی با استفاده از عامل‌هایی تصمیم‌گیرنده که بر روی نحوه‌ی زمان‌بندی تصمیم‌گیری می‌کنند، توصیف می‌شوند. این عامل‌ها با ارسال و دریافت رویدادها از طریق کانال‌های ارتباطاتی با هم تعامل دارند. با توجه به اهمیت نسبی سیم‌گرید، این شبیه‌ساز با جزییات بیشتری مورد بررسی قرار داده می-شود.

در سیم‌گرید منابع رایانشی با مشخص کردن تاخیر و نرخ سرویسی که دارند مدل‌سازی می‌شوند. این مشخصات ممکن است به صورت مقادیر ثابت تنظیم شوند، و یا قابلیت پویایی داشته باشند و بر اساس وضعیت ثبت‌شده برای سیستم تغییر کنند. توپولوژی شبکه کاملا قابل پیکربندی است. به بیان دقیق‌تر، توصیف توپولوژی بر عهده‌ی کاربر گذاشته شده است و کاربر حتی می‌تواند از مجموعه‌ی پیچیده‌ای از لینک‌ها و مسیریاب‌ها به این منظور استفاده نماید. هر یک از عملیات ارتباطاتی و محاسباتی به عنوان یک کار[13] محسوب می‌شوند. مسولیت زمان‌بندی مناسب منابع برای انجان عملیات ارتباطاتی و محاسباتی بر عهده‌ی کاربر است.

---

[10] *SimGrid*
[11] *NP complete*
[12] *C*
[13] *task*

سیم‌گرید پیش‌بینی زمان اجرای عملیات را بر عهده می‌گیرد و میزان خطا در مدت زمان پیش‌بینی شده را به عنوان معیار ارزیابی در اختیار کاربر قرار می‌دهد. به این ترتیب کاربر می‌تواند رفتار الگوریتم را در شرایط پیچیده‌ای که زمان اجرای عملیات به طور دقیق قابل پیش‌بینی نیست، مورد مطالعه قرار دهد. در این محیط شبیه‌سازی، کاربر از طریق «واسط برنامه‌نویسی کاربرد»[14] امکان دست-کاری دو ساختار داده را دارد: منابع و کارها.

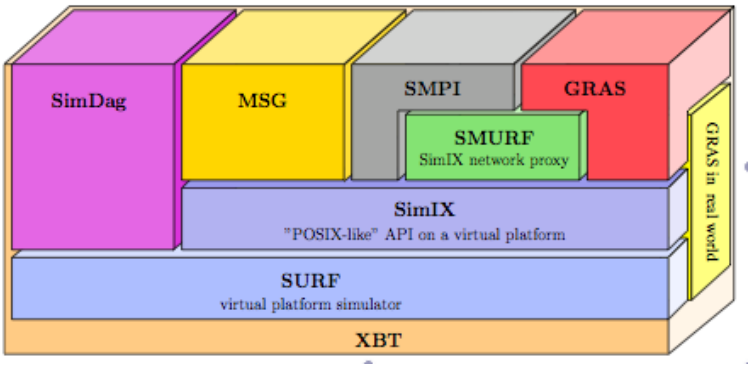
مشخصات منابع و کارها با جزییات بسیار زیادی قابل توصیف هستند. از جمله مشخصات منابع می-توان به سرعت و میزان دسترس‌پذیری آن‌ها اشاره کرد، که همان‌طور که اشاره شد می‌توانند ثابت یا پویا باشند. مشخصات کارها نیز با هزینه و «وضعیتی» که دارند مشخص می‌شود. کاربر می‌تواند با استفاده از واسط برنامه‌نویسی کاربرد به ایجاد کردن، از بین بردن و همچنین بازرسی کردن منابع و کارها بپردازد.

این شبیه‌ساز تنها با فراخوانی یک تابع قابل اجراست و هنگامی که فراخوانی می‌شود فهرستی از کارهایی که از زمان آخرین فراخوانی کامل شده است را باز می‌گرداند.این تابع آرگومانی را نیز به عنوان معیار توقف دریافت می‌کند که این معیار می‌تواند یک بازه‌ی زمانی (تاخیر) و یا یک رخ‌داد (تکمیل یک کار) باشد. به این ترتیب سیم‌گرید در شبیه‌سازی الگوریتم‌های زمان‌بندی هم در «زمان کامپایل» و هم در «زمان اجرا» کاربرد دارد. در دسته‌ی اول تمام تصمیم‌گیری‌های زمان‌بندی پیش از اجرا انجام می‌گیرد و در دسته‌ی دوم برخی از تصمیم‌ها در زمان اجرا گرفته می‌شوند. در حال حاضر، محیط‌های شبیه‌سازی گوناگونی (مانند شبکه‌ی توری، ابر و اِم‌پی‌آی[15]) بر مبنای سیم‌گرید توسعه داده شده‌اند [۸].

در نسخه‌های بعدی این شبیه‌ساز به تدریج ویژگی‌های جدیدی به آن افزوده شد و عمل‌کرد آن از جنبه‌های گوناگونی بهبود یافت. برای مثال در نسخه‌ی دوم امکان استفاده از مدل‌هایی واقعی‌تر از شبکه و همچنین شبیه‌سازی «عامل»های[16] توزیع‌یافته‌ی زمان‌بندی در مقیاس بزرگ فراهم شد. همچنین در نسخه‌ی سوم که در سال ۲۰۰۸ ارایه شد، هسته‌ی شبیه‌ساز به طور کامل بازنویسی شد تا بر سرعت و مقیاس‌پذیری آن افزوده شود. پشته‌ی نرم‌افزاری آخرین نسخه‌ی شبیه‌ساز با تمام مولفه‌های آن در شکل ۳ نشان داده شده است.

---

[14] *Application Programming Interface (API)*
[15] *MPI (Message Passing Interface)*
[16] *agent*



شکل ۳ – شمای کلی از پشته‌ی نرم‌افزاری و مولفه‌های سیم‌گرید [۷].

سیم‌گرید چهار واسط کاربری را ارایه می‌نماید: «سیم‌دَگ» به دنبال نسخه‌ی اول سیم‌گرید، به منظور بازرسی هیوریستیک‌های زمان‌بندی کاربردها، با مدل‌سازی آن‌ها به عنوان «گراف‌های کار»[۱۷]، طراحی شده است. «اِم‌اِس‌جی» واسطی است که در نسخه‌ی دوم سیم‌گرید به منظور مطالعه و بررسی «فرآیندهای متوالی هم‌زمان»[۱۸] معرفی شده است. «گرَس» این امکان را فراهم می‌کند که سیم‌گرید به عنوان محیط توسعه‌ی کاربردهای توزیع‌یافته‌ی واقعی مورد استفاده قرار گیرد. «اِس‌اِم‌پی‌آی» قابلیت شبیه‌سازی مستقیم کاربردهای اِم‌پی‌آی را ایجاد می‌کند.

ماژول «ایکس‌بی‌تی» در واقع جعبه‌ابزاری است که به پیاده‌سازی «کانتِینِر»های[۱۹] کلاسیک داده، مکانیزم‌های ورود به محیط و همچنین بروز خطا می‌پردازد و از قابلیت پیکربندی و انتقال‌پذیری[۲۰] پشتیبانی می‌کند. «سرف»[۲۱] موتور و هسته‌ی شبیه‌ساز است که بسیار ماژولار طراحی شده است تا امکان پیاده‌سازی انواع مختلفی از منابع را ایجاد نماید. «سیم‌آی‌ایکس» ماژولی داخلی است که واسطی مشابه «پُزیکس»[۲۲] را بر روی سرف ایجاد می‌کند تا توسعه‌ی واسط‌های شبیه‌سازی که به پیاده‌سازی چندین فرآیند هم‌زمان می‌پردازند را ساده‌تر نماید. هدف اصلی و اولیه از ماژول «اِسمرف»، ایجاد امکان توزیع فرآیندهای شبیه‌سازی شده بر روی کلاسترها و با استفاده از توان چندین کامپیوتر است. این قابلیت تا حد زیادی بر مقیاس‌پذیری سیم‌گرید می‌افزاید و آن را مناسب عملیات رایانشی بزرگ مقیاس می‌نماید.

---

[17] *task graph*
[18] *Concurrent Sequential Process (CSP)*
[19] *container*
[20] *portability*
[21] *SURF*
[22] *POSIX (Portable Operating System Interface)*

۲-۳- گریدسیم

یکی دیگر از شبیه‌سازهایی که در عملیات رایانشی بزرگ مقیاس به‌کار می‌رود، گریدسیم[۲۳] [۹] است که پس از سیم‌گرید و با اهداف و انگیزه‌هایی بسیار مشابه با آن طراحی و ارایه شده است. یکی از تفاوت‌های اصلی گریدسیم، تاکید آن بر روی جنبه‌ی اقتصادی رایانش توری است. بر همین اساس، مفاهیمی مانند تولید کنندگان (دارندگان منابع)، مصرف‌کنندگان (کاربران انتهایی)، «دلال»هایی[۲۴] که به کشف و اختصاص‌دهی منابع می‌پردازند، در مسایل زمان‌بندی دخالت می‌کنند. اساس کار این شبیه‌ساز نیز مبتنی بر رویداد است. همچنین در محیط شبیه‌سازی امکان وجود چندین دلال قابل لحاظ است که این قابلیت را می‌توان با قابلیت شبیه‌سازی عوامل توزیع‌یافته‌ی زمان‌بندی در سیم‌گرید مقایسه کرد.

گریدسیم با شبیه‌سازی موجودیت‌های مختلف در سیستم‌های محاسباتی موازی و توزیع‌یافته، امکان طراحی و ارزیابی الگوریتم‌های زمان‌بندی در انجام عملیات رایانشی را فراهم می‌آورد. در این شبیه-ساز امکان ایجاد انواع مختلفی از منابع ناهم‌گون وجود دارد که این منابع قابل تجمیع و استفاده برای اجرای کاربردهایی که حجم بالای داده و محاسبات دارند، هستند. به این ترتیب این شبیه‌ساز از انواع گوناگونی از منایع محاسباتی و همچنین از دامنه‌ی وسیعی از کاربردها در علوم مختلف، پشتیبانی می-کند.

به بیان دقیق‌تر، گریدسیم به مدیریت چندین موجودیت مختلف می‌پردازد: کاربر، دلال، منابع، سرویس اطلاعات توری، ورودی و خروجی. مشخصات هر کاربر با نوع کار (مانند زمان اجرا)، استراتژی بهینه‌سازی زمان‌بندی، نرخ فعالیت، زمان منطقه‌ی جغرافیایی[۲۵]، هزینه و مهلت انجام کار، و پارامترهای «سست‌سازی» متناظر با آن‌ها توصیف می‌شود. دلالان کارهایی که توسط کاربران ارسال می‌شود را دریافت کرده و الگوریتم‌های زمان‌بندی آن‌ها را پیاده‌سازی می‌کنند. از آن‌جایی که چندین کاربر برای دسترسی به مجموعه منابعی مشترک رقابت می‌کنند (فرض می‌شود که مجموعه‌ی منابع محدود است)، دلالان باید توازنی میان تقاضاهای کاربران بیابند. این توازن باید به گونه‌ای انجام

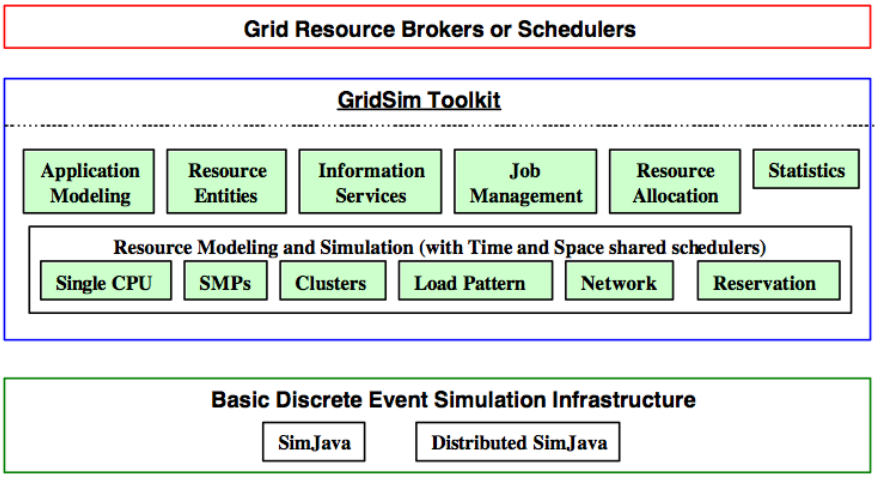
---

[23] *GridSim*
[24] *broker*
[25] *Time zone*

گیرد که برای تمام کاربران قابل پذیرش باشد و به خصوص مهلتی که آن‌ها برای انجام کار دارند را ارضا نماید.

مشخصات منابع با تعداد پردازنده‌ها، هزینه‌ی پردازش، بازده، سیاست زمان‌بندی داخلی، بار کاری و زمان منطقه‌ی جغرافیایی توصیف می‌شود. همچنین گریدسیم ورودی‌ها و خروجی‌ها را به دو شیوه‌ی مجزا و متفاوت مدیریت می‌کند و به این شیوه، راهی برای بیان اختلاف پارامترها و نتایج ارتباطات فراهم می‌نماید. به طور کلی در مقایسه با سیم‌گرید، می‌توان گریدسیم را شبیه‌سازی در سطوح بالاتر عملیات رایانشی دانست که اساسا برای بازرسی تعاملات و واسطهای بین دلال‌های توزیع‌یافته و تصمیم‌های زمان‌بندی انجام گرفته توسط آن‌ها، طراحی شده است. حتی در نسخه‌های ابتدایی توپولوژی شبکه در محیط شبیه‌سازی در نظر گرفته نمی‌شد.

گریدسیم با استفاده از فناوری جاوا طراحی شده است و پیاده‌سازی آن بر اساس هسته‌ی شبیه‌سازی زمان‌گسسته‌ای به نام «سیم‌جاوا»[۲۶] صورت گرفته است. سیم‌جاوا موجودیت‌های مختلف را در «رشته»های[۲۷] مجزا اجرا می‌کند. شبیه‌سازی با یک «شیء»[۲۸] مرکزی که کنترل صف روی‌دادها را بر عهده دارد، مدیریت می‌شود. لازم به ذکر است که تعاملات بین موجودیت‌ها در گریدسیم با استفاده از انواع مختلفی از روی‌دادها که شامل روی‌دادهای داخلی، خارجی، هم‌زمان و غیرهم‌زمان هستند، پیاده‌سازی می‌شود. شمای کلی معماری گریدسیم و مولفه‌های آن در شکل ۴ نمایش داده شده است.



شکل ۴ – شمای کلی معماری گریدسیم و مولفه‌های آن [۹]

[26] *SimJava*
[27] *thread*
[28] *object*

در گریدسیم قابلیت‌های گوناگونی مانند ایجاد کارهای متناظر با کاربردها، نگاشت کارها به منابع و مدیریت آن‌ها وجود دارد. از آن‌جایی که این شبیه‌ساز توانایی مدیریت منابع توزیع‌یافته، مبتنی بر جنبه‌ی اقتصادی و با لحاظ محدودیت‌های هزینه و مهلت زمانی را دارد، بیشتر به منظور مطالعه‌ی الگوریتم‌های بهینه‌سازی زمان و هزینه برای زمان‌بندی کارهای متناظر با کاربردها، روی توری‌های ناهم‌گون، به‌کار می‌رود.

۲-٤- گَنگ‌سیم

گنگ‌سیم[۲۹] [۱۰] را نیز می‌توان به عنوان یکی از شبیه‌سازهای عملیات رایانشی بزرگ مقیاس برشمرد که در اصل توسعه یافته‌ی ابزاری است که به منظور پایش «سازمان‌های مجازی»[۳۰] به‌کار می‌رود. در حوزه‌ی رایانش توری، سازمان مجازی به مجموعه‌ی پویایی از موسسات اطلاق می‌شود که حول مجموعه‌ای از شرایط و قوانین اشتراک منابع تعریف می‌شوند. سازمان‌های مجازی معمولا وجه مشترکی با یک‌دیگر دارند که می‌تواند برای مثال دغدغه‌ها و یا نیازمندی‌های مشترک باشد [۳۳]. گنگ‌سیم امکان شبیه‌سازی محیط‌هایی که در برگیرنده‌ی صدها موسسه و هزاران کاربر است (به عبارتی در مجموع ده‌ها یا صدها هزار سیستم پردازشی و فضای ذخیره‌سازی متناظر با آن‌ها)، را فراهم می‌کند. این شبیه‌ساز یک ساختار مدیریتی مبتنی بر سیاست را شبیه‌سازی می‌کند که در آن، چگونگی تخصیص منابع بر اساس تعاملات سیاست‌های اختصاص منابع درون و بین سازمان‌های مجازی، تعیین می‌شود.

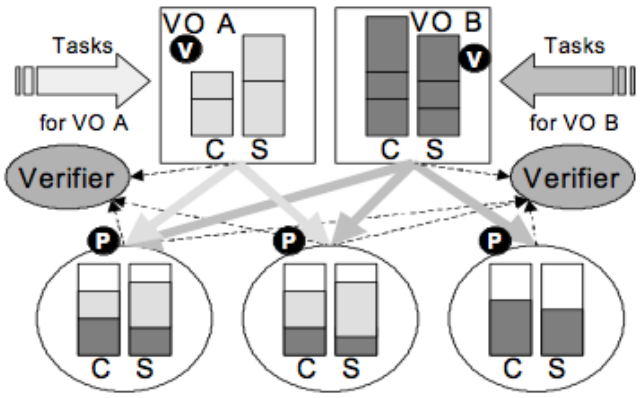
گنگ‌سیم یک شبیه‌ساز زمان‌گسسته و مبتنی بر رویداد است که به طور متناوب وضعیت تمام مولفه-های شبیه‌سازی شده را ارزیابی می‌کند. مولفه‌هایی که در این محیط شبیه‌سازی می‌شوند عبارتند از: سایت، سازمان مجازی، زمان‌بندهای داخلی، خارجی و داده، نقطه‌ی پایش داده، نقطه‌ی اجرای سیاست سایت و نقطه‌ی اجرای سیاست سازمان مجازی. مشخصات سایت با ظرفیت منابع آن (تعداد پردازنده‌ها و منابع شبکه‌ای درون و بین سایت‌ها و فضای ذخیره‌سازی) و سیاست زمان‌بندی آن (که احتمال تخصیص منابع به هر یک از سازمان‌های مجازی را بیان می‌کند) توصیف می‌شود. هر گروهی از کابران که مجموعه‌ای از کارها را ارسال می‌کنند، به عنوان یک سازمان مجازی در نظر گرفته می-

---

[29] *GangSim*
[30] *Virtual Organization (VO)*

شود. هر گروه با سیاستی نمایش داده می‌شود که سازمان مجازی برای تخصیص منابع در پیش می-گیرد.

در شکل ۵ شمای کلی معماری گنگ‌سیم با تاکید بر نحوه‌ی تخصیص منابع در آن نمایش داده شده است. در این شکل دو سازمان مجازی (که با کادر مربع شکل مشخص شده‌اند)، سه سایت (که با کادر دایره شکل مشخص شده‌اند) و نحوه‌ی ارتباطات آن‌ها نشان داده شده است. منابع محاسباتی و فضای ذخیره‌سازی متناظر با هر یک از این مولفه‌ها نیز به طور جداگانه (به ترتیب با حروف سی و اِس) مشخص شده‌اند.



شکل ۵ – شمای کلی معماری گنگ‌سیم و نحوه‌ی تخصیص منابع در آن [۱۰]

در گنگ‌سیم، زمان‌بندها نقاطی را نشان می‌دهند که در آن‌ها بر روی زمان‌بندی تصمیم‌گیری می‌شود. نقاط پایش داده، نقاطی هستند که به جمع‌آوری و توزیع اطلاعات مربوط به شاخص‌های مصرف منابع می‌پردازند. در نهایت نقاط اجرای سیاست، با استفاده از شاخص‌های پایش‌شده مانع از تخطی از سیاست‌های سایت‌ها و سازمان‌های مجازی شده و به اجرای آن‌ها می‌پردازند. در گنگ‌سیم تنها در شرایطی کار ارسال شده پذیرفته می‌شود که یا میانگین بهره‌وری منابع مربوط به سازمان مجازی در طول یک بازه‌ی بلند زمانی، از حد آستانه‌ی مشخصی پایین‌تر باشد و یا این که منابع در دسترس بوده و میانگین بهره‌وری منابع مربوط به سازمان مجازی در طول یک بازه‌ی کوتاه زمانی، از حد آستانه‌ی مشخصی کم‌تر باشد.

همان‌طور که اشاره شد، گنگ‌سیم توسعه یافته‌ی یکی از ابزارهای پایش سازمان‌های مجازی به نام گنگ‌لیا[31] است که علاوه بر جایگزین کردن برخی از ماژول‌های این ابزار با مولفه‌های مدل‌سازی شده، قابلیت‌هایی را برای توصیف بار کاری و ایجاد محیط رایانش توری به آن افزوده است. در گنگ‌سیم برای پیاده‌سازی مولفه‌های شبیه‌سازی از سیاست‌های تخصیص کار، شاخص‌های تجمیع، مولفه‌های توری و ثبات‌های حالت محیط استفاده شده است. این شبیه‌ساز وضعیت تمام مولفه‌های شبیه‌سازی شده را به طور متناوب ارزیابی می‌کند که این دوره‌ی تناوب دقت زمانی شبیه‌سازی را تعیین می‌کند و معمولا در بازه‌ی ۱۰ تا ۳۰ ثانیه قرار دارد. به این ترتیب گنگ‌سیم هم در ارزیابی بار کاری هم‌زمان که در آن تمام سازمان‌های مجازی کارهای خود را در زمان یکسانی ارسال می‌کنند و هم در ارزیابی بار کاری غیرهم‌زمان که در آن سازمان‌های مجازی کارهای خود را در زمان‌های متفاوتی ارسال می‌کنند، به کار گرفته می‌شود.

۲-۵- اُپتُرسیم

اپترسیم[32] [۱۱] یکی دیگر از شبیه‌سازهایی است که در عملیات رایانشی بزرگ مقیاس و به خصوص برای شبیه‌سازی استراتژی‌های تکرار[33] داده در توری‌های داده به‌کار برده می‌شود. انگیزه‌ی اصلی در طراحی اپترسیم را می‌توان، کمبود محیط‌های شبیه‌سازی برای پردازش کاربردهای توری با مجموعه دادگان بسیار بزرگ دانست. یکی از مفاهیم کلیدی در توری‌های داده، تکرار داده است که مساله‌ی ایجاد و مدیریت تکرار داده‌ها در نقاط مختلف جغرافیایی، به منظور بهینه‌سازی هزینه‌ی دسترسی به داده را در بر می‌گیرد. هدف اپترسیم بررسی رفتار گذرا و حالت پایدار شیوه‌های بهینه‌سازی تکرار داده است.

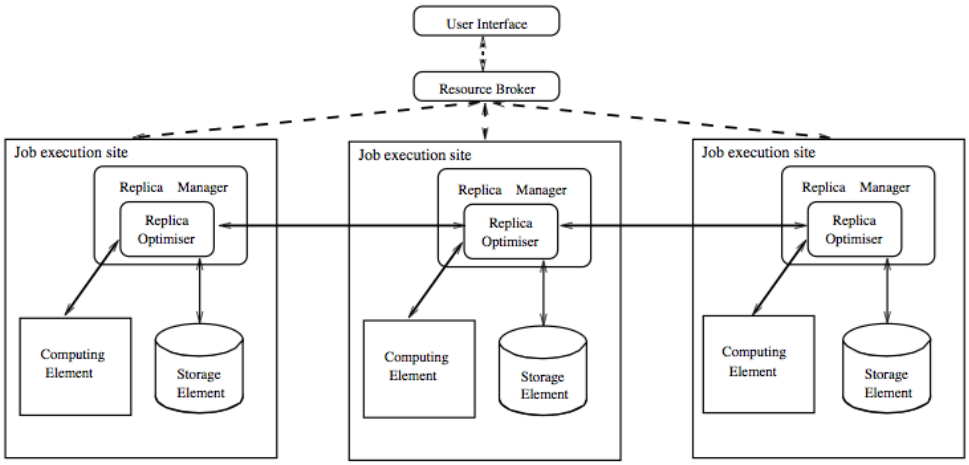
ساختار توری پذیرفته شده در اپترسیم بر گرفته از معماری ساده‌سازی شده‌ی است که در پروژه‌ی توری داده [۱۲] ارایه شده است. شمای کلی این معماری در

شکل ۶ نمایش داده شده است. بر اساس این مدل، ساختار توری از سایت‌هایی تشکیل یافته است که هر یک از آن‌ها می‌توانند به تامین منابع برای کارهای ارسالی بپردازند. این شبیه‌ساز با دریافت اطلاعات مربوط به توپولوژی شبکه‌ی توری و منابع آن، مجموعه‌ای از کارها که باید اجرا شوند، و

---

[31] *Ganglia*
[32] *OptorSim*
[33] *Replication*

استراتژی بهینه‌سازی به عنوان ورودی، مجموعه کارها را بر روی توری شبیه‌سازی شده اجرا می‌کند. همچنین با اندازه‌گیری مجموعه‌ای از شاخص‌ها، امکان ارزیابی کمی کارایی استراتژی‌های بهینه-سازی تحت شرایط آزمایش شده را فراهم می‌کند.



شکل ۶ – شمای کلی معماری اپترسیم و مولفه‌های آن [۱۱]

به بیان دقیق‌تر، اپترسیم تعاملات هر یک از مولفه‌های توری‌های داده را مدل‌سازی می‌کند. به این منظور این شبیه‌ساز مفاهیم زیر را در نظر می‌گیرد: سایت‌ها که اغلب تامین‌کننده‌ی منابع پردازشی و یا منابع ذخیره‌سازی داده هستند، دلال‌ها که زمان‌بندی کارها برای منابع مجاسباتی را بر عهده دارند، و مسیریاب‌ها که فاقد منابع محاسباتی و ذخیره‌سازی هستند. به این ترتیب با استفاده از داده‌ای که بر روی منابع ذخیره‌سازی ذخیره شده است، کارها بر روی منابع محاسباتی اجرا می‌شوند. توپولوژی شبکه در اپترسیم با بیان تعداد لینک‌های بین سایت‌ها و میزان پهنای باند در دسترس آن‌ها توصیف و مشخص می‌شود.

همچنین در این شبیه‌ساز می‌توان الگوهای گوناگونی را برای دسترسی به فایل‌ها (به بیان دقیق‌تر، ترتیبی که فایل‌ها در دسترس کارها قرار می‌گیرند) در نظر گرفت. این الگوها عبارتند از: متوالی (طبق یک لیست دارای ترتیب)، دسترسی بر اساس انتخاب تصادفی از لیست، حرکت تک‌واحدی تصادفی

روی لیست (یک گام در هر جهت)، حرکت تصادفی گوسی[۳۴] (انتخاب بر اساس توزیع گوسی حول فایل پیشین).

اپترسیم بر اساس معماری ماژولار طراحی شده است و از این‌رو امکان اتصال انواع مولدهای الگوهای دسترسی فایل (به منظور شبیه‌سازی بار کاری) و همچنین انواع مختلفی از الگوریتم‌های بهینه‌سازی به خود را فراهم می‌آورد. در این شبیه‌ساز، هر یک از منابع محاسباتی با رشته‌ای مجزا پیاده‌سازی می‌شود. زمان‌بندی کارها بر روی منابع توسط دلال منابع نیز با رشته‌ی هم‌زمان دیگری مدیریت می‌شود. در نهایت، هر یک از منابع تنها برای اجرای یک کار منفرد در نظر گرفته می‌شود. به این ترتیب، اپترسیم برای ارزیابی و مقایسه‌ی انواع گوناگونی از استراتژی‌های تکرار داده، مانند استراتژی عدم تکرار و یا استراتژی تکرار غیر مشروط که در صورت نیاز به فضا به پاک کردن قدیمی‌ترین فایل می‌پردازد، قابل به‌کارگیری است.

۲-٦- کلُدسیم

کلدسیم[۳۵] [۲۰،۲۱ و۲۲] یکی از مطرح‌ترین شبیه‌سازهایی است که در عملیات رایانشی بزرگ‌مقیاس و به طور خاص عملیات رایانش ابری به‌کار گرفته می‌شود. کلدسیم از مدل‌سازی و شبیه‌سازی محیط‌های مبتنی بر ابر مراکز داده و واسط‌هایی که به مدیریت ماشین‌های مجازی، حافظه، فضای ذخیره‌سازی و پهنای باند اختصاص می‌یابند، پشتیبانی می‌نماید. شمای کلی معماری لایه‌ای کلدسیم و مولفه‌های آن در شکل ۷ نمایش داده شده است.

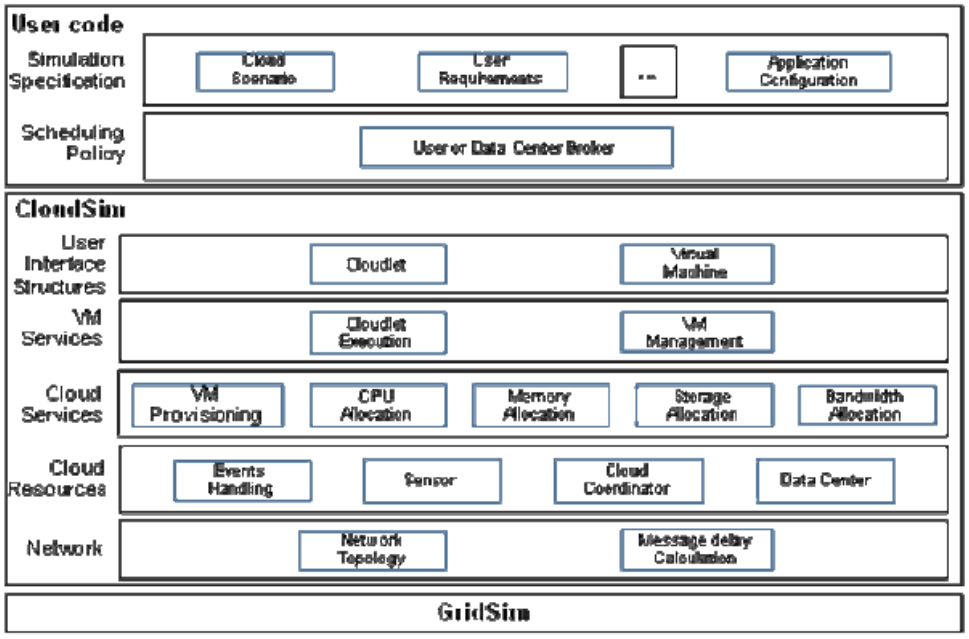
در پایین‌ترین لایه گریدسیم (که در بخش‌های پیشین مورد بررسی قرار گرفت) قرار می‌گیرد تا از قابلیت‌ها و مولفه‌های آن مانند مدل‌سازی منابع و بار کاری، و سرویس‌های اطلاعاتی استفاده شود. در لایه‌ی بعدی کلدسیم، با توسعه‌ی قابلیت‌های اصلی گریدسیم پیاده‌سازی می‌شود. لایه‌ی کلدسیم به مدیریت اجرای مولفه‌های هسته‌ای یعنی ماشین‌های مجازی، میزبان‌ها، مراکز داده و کاربردها در طول شبیه‌سازی می‌پردازد. مسایل اساسی مانند تدارک میزبان‌ها برای ماشین‌های مجازی و بر اساس تقاضاهای کاربران، مدیریت اجرای کاربردها و پایش پویا توسط این لایه حل می‌شود. بالاترین لایه در پشته‌ی معماری لایه‌ی کد کاربر است که عملیات مربوط به پیکربندی میزبان‌ها (تعداد ماشین‌ها و

---

[34] *Gaussian distribution*
[35] *CloudSim*

مشخصات فنی آن‌ها)، کاربردها (تعداد عملیات و نیازمندی‌های آن‌ها)، ماشین‌های مجازی، تعداد کاربران و نوع کاربرد مورد درخواست آن‌ها، و سیاست‌های زمان‌بندی را بر عهده دارد.

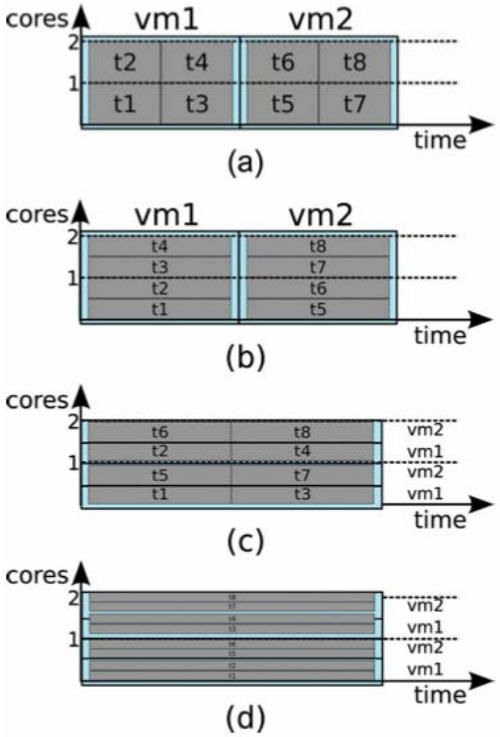
یکی از مولفه‌های اصلی که در این شبیه‌ساز مدل‌سازی می‌شود، مرکز داده است که به تقاضاهای دریافتی رسیدگی می‌کند. این تقاضاها مربوط به کاربردهایی هستند که ماشین‌های مجازی متناظر با آن‌ها باید به میزبان‌های واقع در مرکز داده اختصاص داده شده و سرویس‌دهی شوند. مرکز داده از مجموعه‌ای از میزبان‌ها تشکیل یافته است که مسوولیت مدیریت ماشین‌های مجازی در طول چرخه‌ی عمر (ایجاد شدن، مهاجرت[۳۶] کردن و از بین رفتن) آن‌هارا بر عهده دارند. میزبان‌ها، مولفه‌هایی هستند که گره‌های پردازشی مرکز داده را (به صورت تک‌هسته‌ای یا چند هسته‌ای) مدل‌سازی می‌کنند و با ظرفیت پردازشی (بر حسب میلیون دستورالعمل بر ثانیه یا میپس)، حافظه، فضای ذخیره‌سازی و سیاستی که در اختصاص‌دهی هسته‌های پردازشی به ماشین‌های مجازی دنبال می‌کنند مشخص می‌شوند.



شکل ۷ – شمای کلی معماری لایه‌ای کلدسیم و مولفه‌های آن [۲۰]

[36] *migration*

یکی از جنبه‌های اساسی که عملیات رایانشی ابری را از عملیات رایانشی توری متمایز می‌سازد، این است که اجرای عملیات رایانشی بزرگ مقیاس از نوع اول به شدت نیازمند به استفاده از ابزارها و فناوری‌های مجازی‌سازی است. از این‌رو کلدسیم در دو سطح به تدارک ماشین‌های مجازی می-پردازد: در سطح میزبان و در سطح ماشین مجازی. در سطح اول این امکان وجود دارد که مشخص شود چه مقدار از توان پردازشی هر هسته‌ی یک میزبان به هر ماشین مجازی تخصیص می‌یابد. در سطح دوم مشخص می‌شود که هر ماشین مجازی چه مقدار از توان پردازشی در دسترس خود را در اختیار هر یک از کاربردهایی که بر روی موتور اجرایی آن میزبانی می‌شوند قرار می‌دهد.در هر دو سطح می‌توان از سیاست‌های تخصیص منابع متفاوتی مانند «اشتراک زمانی»[۳۷] و یا «اشتراک مکانی»[۳۸] استفاده نمود. برای وضوح بیشتر این مساله در شکل ۸ ترکیب‌های مختلفی از دو سیاست اشاره شده نشان داده شده است.



شکل ۸ - ترکیب‌های مختلفی از دو سیاست اشتراک زمانی و اشتراک مکانی [۲۱]

[37] *Time-shared*
[38] *Space-shared*

در این شکل تاثیر سیاست‌های مختلف تدارک ماشین مجازی، در چگونگی اجرای عملیات رایانشی نشان داده شده است. در حالت اول (به ترتیب از بالا) سیاست اشتراک مکانی برای ماشین‌های مجازی (سطح اول) و وظایف عملیاتی (سطح دوم)، در حالت دوم سیاست اشتراک مکانی برای ماشین‌های مجازی و سیاست اشتراک زمانی برای وظایف عملیاتی، در حالت سوم سیاست اشتراک زمانی برای ماشین‌های مجازی و سیاست اشتراک مکانی برای وظایف عملیاتی، و در نهایت در حالت چهارم سیاست اشتراک زمانی برای ماشین‌های مجازی و وظایف عملیاتی در پیش گرفته شده است.

در کلدسیم امکان مدل‌سازی جنبه‌های اقتصادی اجرای عملیات رایانشی در مراکز داده نیز فراهم شده است. برای مثال می‌توان هزینه‌های متناظر با استفاده از هر واحد حافظه، هر واحد از فضای ذخیره-سازی و هر واحد از پهنای باند را پیکربندی نمود. همچنین می‌توان هزینه‌ی حاصل از اجرای هر یک از وظایف عملیاتی را نیز لحاظ نمود. در این شبیه‌ساز المان‌های شبکه مانند لینک‌های ارتباطی و مسیریاب‌ها به طور مشخص مدل‌سازی نمی‌شوند، بلکه در قالب یک مدل انتزاعی، تاخیر حاصل از عمل‌کرد این المان‌ها در انتقال داده‌ها و انجام عملیات رایانشی در محیط شبیه‌سازی لحاظ می‌شود.

یکی از ویژگی‌های مهم این شبیه‌ساز، پشتیبانی آن از الگوهای گوناگون و پویای بار کاری است. برای مدل‌سازی مشخصات کاربردهای مختلف از مولفه‌ای به نام «کُلدِلِت»[۳۹] استفاده می‌شود. همچنین برای ایجاد پویایی در رفتار کاربردها، مولفه‌ی مذکور توسعه یافته و «مدل بهره‌وری»[۴۰] به آن اضافه شده است. این مدل دارای پارامترهایی است که با تعریف این پارامترها، می‌توان هنگام پیاده‌سازی کاربردهای مختلف، نیازمندی‌هایی که آن‌ها در سطح ماشین مجازی و منابع رایانشی دارند را مشخص نمود.

همان‌طور که پیش‌تر اشاره شد، یکی از لایه‌های معماری کلدسیم از گریدسیم تشکیل یافته است. طبق بررسی‌های انجام گرفته در بخش‌های قبل، گریدسیم برای مدیریت رویدادها از چارچوب شبیه‌سازی سیم‌جاوا استفاده می‌کند. به منظور رفع محدودیت‌هایی که به‌کارگیری سیم‌جاوا اعمال می‌کند و برای شبیه‌سازی سناریوهای پیچیده‌تر، در معماری کلدسیم چارچوب شبیه‌سازی گسسته‌ی جدیدی توسعه داده شده است. این چارچوب از چندین مولفه تشکیل شده است که مولفه‌های اصلی آن عبارتند از:

- «کلدسیم» اصلی‌ترین مولفه‌ی است که صفوف رویدادها را مدیریت می‌کند و اجرای شبیه-سازی را به صورت گام به گام جلو می‌برد.

---

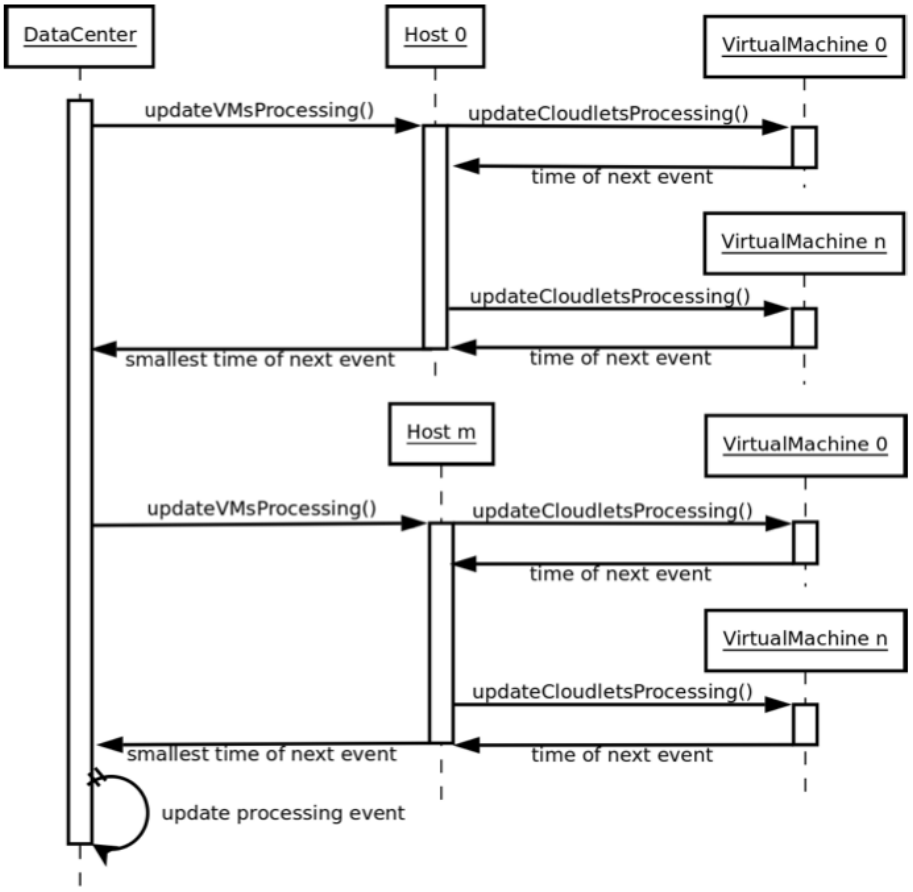
[39] *Cloudlet*
[40] *Utilization model*

- «صف تاخیر»[۴۱] مولفه‌ای است که رویدادهای به تاخیر افتاده را پیاده‌سازی می‌کند.
- «صف آینده»[۴۲] مولفه‌ای است که رویدادهایی که در آینده در دسترس مولفه‌ی کلدسیم فرا می‌گیرد را پیاده‌سازی می‌کند.
- «سرویس اطلاعات ابر» مولفه‌ای است که سرویس‌های مربوط به ثبت[۴۳]، اندیس‌گذاری و یافتن[۴۴] منابع را فراهم می‌کند. سایر مولفه‌ها مانند کاربرها و دلال‌های مراکز داده می‌توانند از این مولفه، برای اطلاعات درباره‌ی منابع در دسترس، درخواست سرویس نمایند.
- «برچسب»[۴۵]ها، دستورات استاتیکی هستند که نوع عملی که مولفه‌های گریدسیم هنگام ارسال یا دریافت رویدادها باید انجام دهند را نشان می‌دهند.
- مولفه یا کلاس «سیم‌ایونِت»[۴۶] نشان‌دهنده‌ی رویدادی است که در شبیه‌سازی بین موجودیت‌ها تبادل می‌شود.
- «پیش‌بینی کننده»ها مولفه‌هایی هستند که برای انتخاب رویدادها از صف تاخیر مورد استفاده قرار می‌گیرند.

پردازش وظایف عملیاتی بر عهده‌ی ماشین‌های مجازی متناظر با آن‌ها است و در هر مرحله از شبیه-سازی باید پیشرفت آن‌ها به‌روز رسانی و پایش شود. به این منظور، یک رویداد داخلی برای اطلاع موجودیت مرکز داده از زمان مورد انتظار برای تکمیل عملیات ایجاد می‌شود.

---

[41] *Deferred Queue*
[42] *Future Queue*
[43] *registration*
[44] *discovery*
[45] *tag*
[46] *SimEvent*



شکل ۹ – فرآیند به‌روز رسانی [۲۲]

بنابراین، در هر مرحله از شبیه‌سازی، هر یک از مراکز داده، به ازای هر یک از میزبان‌های خود متدی را فراخوانی می‌کنند، که در نتیجه‌ی آن وضعیت پردازش وظایف عملیاتی در حال اجرا بر روی ماشین‌های مجازی را به‌روز رسانی می‌شود. آرگومان ورودی این متد، زمان کنونی شبیه‌سازی است و مقداری که به عنوان خروجی باز می‌گرداند، زمان مورد انتظار برای تکمیل اجرای یکی از وظایف عملیاتی در حال اجرا بر روی هر یک از ماشین‌های مجازی میزبان‌های مرکز داده است. کم‌ترین زمانی که توسط میزبان‌ها اعلام (برگردانده) می‌شود به عنوان زمان ایجاد روتداد داخلی بعدی در نظر گرفته می‌شود.

به دنبال فراخوانی تابع به‌روز رسانی پردازش ماشین‌های مجازی توسط مرکز داده، تابع به‌روز رسانی کاربردها در سطح میزبان‌ها فراخوانی می‌شود. این فراخوانی منجر می‌شود به این که ماشین‌های مجازی وضعیت پردازش وظایف عملیاتی خود را (اتمام یافته، متوقف شده و یا در حال اجراست) به-

روز رسانی کنند. این متد مشابه متد قبل پیاده‌سازی می‌شود با این تفاوت که پیاده‌سازی در سطح ماشین‌های مجازی صورت می‌گیرد. نمودار زمانی کل این فرآیند در
شکل ۹ نمایش داده شده است.

به این ترتیب ماشین‌های مجازی زمان مورد انتظار برای تکمیل اجرای وظایف عملیاتی که در حال حاضر بر عهده دارند را اعلام می‌کنند (باز می‌گردانند). از میان مقادیر محاسبه شده برای زمان‌های مورد انتظار، کم‌ترین مدت زمان اجرایی که اعلام شده به موجودیت مرکز داده فرستاده می‌شود. در نتیجه مدت زمان‌های اجرا در صفی نگهداری می‌شوند و مرکز داده بر مبنای این مقادیر، گام‌های شبیه‌سازی را پیش می‌برد. اگر وظایف عملیاتی تکمیل شده‌ای در این صف در حال انتظار باشند، در این صورت از صف خارج می‌شوند و به کاربر ارسال می‌شوند. بنابراین کل فرآیند شبیه‌سازی بر این مبنای ساده پیاده‌سازی می‌شود.

به این ترتیب با استفاده از کلدسیم، پژوهش‌گران می‌توانند با ترکیب انواع پیکربندی‌های کاربردها و توزیع تقاضاهای کاربران در محیط شبیه‌سازی، و یا با توصیف انواع سناریوهای تست مختلف بر اساس پیکربندی‌هایی که به طور پیش‌فرض توسط شبیه‌ساز پشتیبانی می‌شود، به شبیه‌سازی عملیات رایانشی بزرگ مقیاس و توسعه‌ی کاربردهای متناظر با آن‌ها بپردازند.

۲-۷- گرین کلُد

گرین‌کلد[۴۷] [۲۳] یکی دیگر از شبیه‌سازهایی است که در حوزه‌ی عملیات رایانشی بزرگ مقیاس به‌کار گرفته می‌شود و تاکید آن بر روی ارزیابی کارایی انرژی در مراکز داده است. این شبیه‌ساز توسعه‌یافته‌ی شبیه‌ساز شبکه‌ای به نام اِن‌اِس۲ [۲۴] است که امکان مدل‌سازی و بررسی دقیقی از میزان انرژی مصرفی المان‌های مراکز داده، از جمله سرورها، سوییچ‌ها و لینک‌ها را تحت انواع توزیع بارکاری، برای کاربران فراهم می‌نماید. علاوه بر این، در مقایسه با سایر شبیه‌سازهای مشابه، مانند کلدسیم، گرین‌کلد تنها شبیه‌سازی است که در آن تاکید خاصی بر روی ارتباطاتی که در زیرساخت مراکز داده صورت می‌گیرد شده است.

به بیان دقیق‌تر، گرین‌کلد انرژی مصرف شده توسط هر دو نوع المان مراکز داده (هم المان‌های پردازشی و هم لینک‌های ارتباطی) را در شبیه‌سازی در نظر می‌گیرد که این قابلیت، امکان کنترل

---

[47] *GreenCloud*

سطح بالایی را ایجاد می‌کند. در این شبیه‌ساز، مولفه‌های سرور با گره‌های «تک هسته»ای[۴۸] پیاده‌سازی می‌شوند که توان پردازشی آن‌ها بر حسب میپس[۴۹] (میلیون دستور بر ثانیه) یا بر حسب فلاپس، و ظرفیت منابع ذخیره‌سازی و حافظه‌ی آن‌ها از پیش تعیین شده است و دارای مکانیزم‌های مختلف زمان‌بندی کار هستند. سرورها در رَک[۵۰]ها جای می‌گیرند و از طریق سوییچی به لایه‌ی دسترسی شبکه متصل می‌شوند.

مدل‌سازی توان سرورها وابسته به بهره‌وری واحد پردازنده‌ی آن‌هاست. یک سرور در حالت بیکاری حدود دو سوم مصرف پیک بار خود انرژی مصرف می‌نماید [۲۵و۲۶]. این مقدار انرژی مصرفی صرف انجام وظایف دایمی و ثابت سرورها مانند مدیریت ماژول‌های حافظه، دیسک‌ها، منابع ورودی و خروجی، و دیگر ابزارهای جانبی می‌شود. از سویی دیگر، مصرف توان به طور خطی با افزایش بار پردازنده‌ی سرور افزایش می‌یابد. از این‌رو گرین‌کلد، در مدل‌سازی توان سرورها اجازه می‌دهد تا مکانیزم‌های «حفظ توان»[۵۱] متمرکزی پیاده‌سازی شود تا امکان تجمیع بارهای کاری در حداقل تعداد سرورهای پردازشی فراهم گردد.

روش‌های دیگری نیز برای مدیریت توان قابل انتخاب است که از آن جمله می‌توان به «مقیاس‌دهی پویای ولتاژ/ فرکانس»[۵۲] [۲۷] اشاره نمود. این روش‌های مدیریت توان برای المان‌های ارتباطات در شبکه نیز قابل به‌کارگیری است. از جمله‌ی این المان‌ها سوییچ‌ها و لینک‌های شبکه هستند که بار کاری را برای اجرا به سرورهای پردازشی هدایت می‌کنند. از آن‌جایی که توان مصرفی این المان‌ها به طور خطی با نرخ انتقال بار کاری توسط آن‌ها افزایش می‌یابد، با به‌کارگیری روش مقیاس‌دهی پویای ولتاژ، می‌توان با توجه به الگوی بار ترافیکی و سطح بهره‌وری آن‌ها، نرخ انتقال را به منظور مدیریت توان، تنظیم نمود. در این شبیه‌ساز می‌توان از راه‌کارهای متفاوتی برای اتصال سوییچ‌ها و سرورها استفاده کرد که راه‌کار انتخابی به پهنای باند در دسترس و کیفیت و خصوصیات لینک فیزیکی مدل‌سازی شده وابسته است.

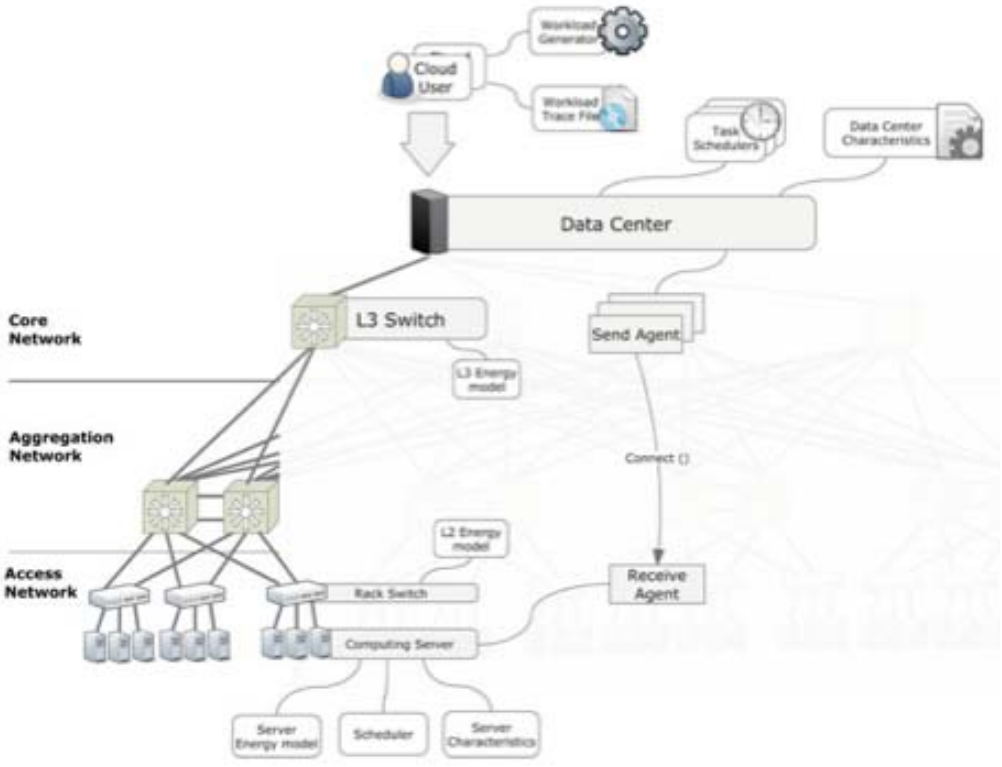
مدل معماری گرین‌کلد در

شکل ۱۰ نشان داده شده است. همان‌طور که در شکل مشاهده می‌شود، این مدل در انطباق با معماری متداول مراکز داده، از سه لایه‌ی هسته، توزیع و دسترسی تشکیل یافته است. در این شبیه‌ساز از انواع متفاوتی از الگوهای بار کاری پشتیبانی می‌شود. از جمله‌ی این الگوها می‌توان به بار کاری «با حجم

---

[48] *Single core*
[49] *Mips (Million Instructions per Second)*
[50] *rack*
[51] *Power saving*
[52] *DVFS (Dynamic Voltage/Frequency Scaling)*

پردازشی بالا»[۵۳] اشاره کرد. این الگو بار کاری کاربردهای پربازده که به حل مسایل محاسباتی می‌پردازند را مدل‌سازی می‌کند. باید توجه داشت که سرورهایی که این نوع بار کاری را پردازش می‌کنند، نیاز چندانی به انتقال داده از طریق شبکه‌ی مرکز داده را ندارند.



شکل ۱۰ – مدل معماری لایه‌ای گرین‌کلد [۲۳]

الگوی دیگر بار کاری «با حجم داده‌ی بالا»[۵۴] است که بر خلاف نوع قبل، بار محاسباتی چندانی را در سرورهای پردازشی ایجاد نمی‌کند، اما در مقابل نیاز شدیدی به انتقال داده دارد. این نوع بار کاری، کاربردهایی مانند به اشتراک‌گذاری فایل‌های ویدیویی را مدل‌سازی می‌کند که تقاضاهای کاربران منجر به انتقال جریان‌های ویدیویی می‌شود. در این حالت، نقش المان‌های ارتباطی در شبکه‌ی مراکز داده از اهمیت بالایی برخوردار می‌شود و نمی‌توان در فرآیند شبیه‌سازی از آن‌ها صرف‌نظر نمود.

---

[53] *Computationally intensive*
[54] *Data intensive*

۲-۸- فنون و ابزارهای شبیه‌سازی بزرگ مقیاس

علاوه بر فنون و ابزارهایی که تا کنون مورد بررسی قرار گرفتند و برای شبیه‌سازی عملیات رایانشی در محیط‌های رایانشی بزرگ مقیاس به‌کار گرفته می‌شوند (دسته‌ی اول)، دسته‌ای دیگر از فنون و ابزارها نیز مطرح می‌شوند که برای استفاده در شبیه‌سازی‌های بزرگ مقیاس مناسب هستند. این دسته که تحت عنوان فنون و ابزارهای شبیه‌سازی بزرگ‌مقیاس مطرح می‌شوند، به منظور انجام عملیات رایانشی بزرگ‌مقیاس، قابلیت به‌کارگیری و توسعه بر روی چنین بسترهایی را دارند. به طور کلی این دسته از شبیه‌سازها به ظرفیت بالایی از منابع محاسباتی نیاز دارند و بسیاری از آن‌ها با استفاده از فنون متداول در «محاسبات پربازده»[۵۵] پیاده‌سازی می‌شوند. برای روشن‌تر شدن تفاوتی که بین این دسته از شبیه-سازها، با فنون و ابزارهایی که تا اینجا مورد بررسی قرار گرفتند، وجود دارد، در این بخش به برخی از آن‌ها اشاره می‌کنیم.

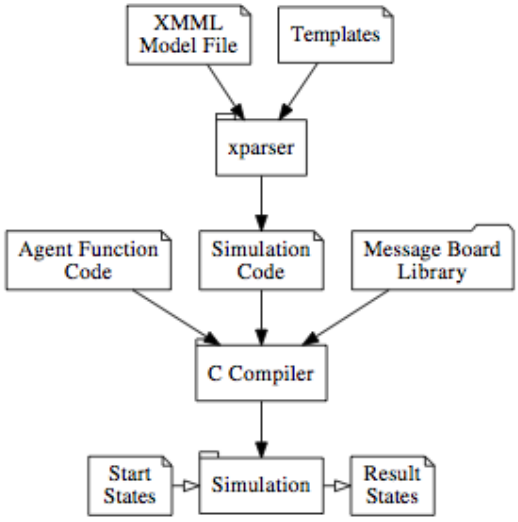
۲-۸-۱- فلِیم

یکی از شبیه‌سازهای بزرگ مقیاسی که در حوزه‌ی عملیات رایانشی و به‌خصوص در پردازش موازی به‌کار می‌رود، فلیم[۵۶] [۱۳] است. چارچوب این شبیه‌ساز «مبتنی بر عامل»[۵۷] است و اساسا به صورت موازی طراحی شده است. از این‌رو محیط شبیه‌سازی با توزیع عوامل بر روی گره‌های پردازشی موازی‌سازی می‌شود، بدون این‌که برای استفاده از آن نیازی به تخصص در پردازش موازی باشد.

هر عامل، با یک «ماشین حالت»[۵۸] بدون چرخه نمایش داده می‌شود، که این ماشین حالت، رفتار عامل در هر تکرار[۵۹] را مشخص می‌کند. «توابع گذر از حالت»[۶۰] متناظر با هر عامل، هم به حافظه‌ی داخلی عامل و هم به اطلاعات جریان‌های ورودی و خروجی آن دسترسی دارند. در فلِیم، توصیف مدل شبیه‌سازی از سند ایکس‌اِم‌اِل استخراج و با تجزیه‌ی آن کد شبیه‌سازی با استفاده از سی نوشته می‌شود. این کد به همراه کد توابع گذر عامل‌ها (گذر از حالتی به حالت دیگر) که توسط مدل‌ساز و با استفاده از سی نوشته شده است، کامپایل می‌شوند. شمای کلی مدل معماری این شبیه‌ساز در شکل ۱۱ نمایش داده شده است.

---

[55] *High Performance Computing*
[56] *Flame*
[57] *Agent-based*
[58] *state machine*
[59] *iteration*
[60] *state transition function*



شکل ۱۱ – شمای کلی مدل معماری فلیم [۱۳]

بنابراین در این شبیه‌سازی، مدل‌ساز با تعریف مجموعه‌ای از توابع گذر، که چگونگی تغییر حالت عامل از حالتی به حالت دیگر را بیان می‌کنند، به توصیف عامل می‌پردازد. به این ترتیب این توابع گذر به همراه حالت‌های متناظر با آن‌ها در ماشین حالت بدون چرخه، رفتار عامل را مشخص می‌کنند. ساختار هر عامل شامل یک حافظه‌ی داخلی است که حالت عامل را نگه‌داری می‌کند و برای اعمال توابع گذر، مقادیر آن دریافت و به‌روز رسانی می‌شوند. تنها ارتباط میان عوامل از طریق پیغام‌هایی است که بین جریان‌های ورودی و خروجی آن‌ها رد و بدل می‌شود. به بیان دیگر، ماهیت عامل‌ها موازی است و هر عامل به صورت مستقل قابل مدیریت است. بنابراین شبیه‌ساز می‌تواند با توزیع عوامل بر روی گره‌های پردازشی مجزا، در مقیاسی بزرگ به انجام عملیات رایانشی بپردازد.

۲-۸-۲- سویجِس

یکی دیگر از شبیه‌سازهای بزرگ مقیاس که در عملیات رایانشی به‌کار می‌رود سویجِس[۶۱] [۱۴] نام دارد که هدف اولیه‌ی طراحی آن، ساده کردن فرآیند مدل‌سازی است. به این ترتیب که به مدل‌ساز اجازه می‌دهد تا پارامترهای فضاهای جستجوی موردنظر خود را تعریف و برای اجرا به این شبیه‌ساز ارسال نماید. سپس سویجس تمام عملیات شبیه‌سازی را زمان‌بندی و اجرا می‌کند و نتایج حاصل را نیز

[61] *SWAGES*

جمع‌آوری و بر روی فایل‌سیستم ذخیره می‌نماید. مولفه‌های معماری این شبیه‌ساز به دو بخش سرور و کلاینت تقسیم می‌شوند.

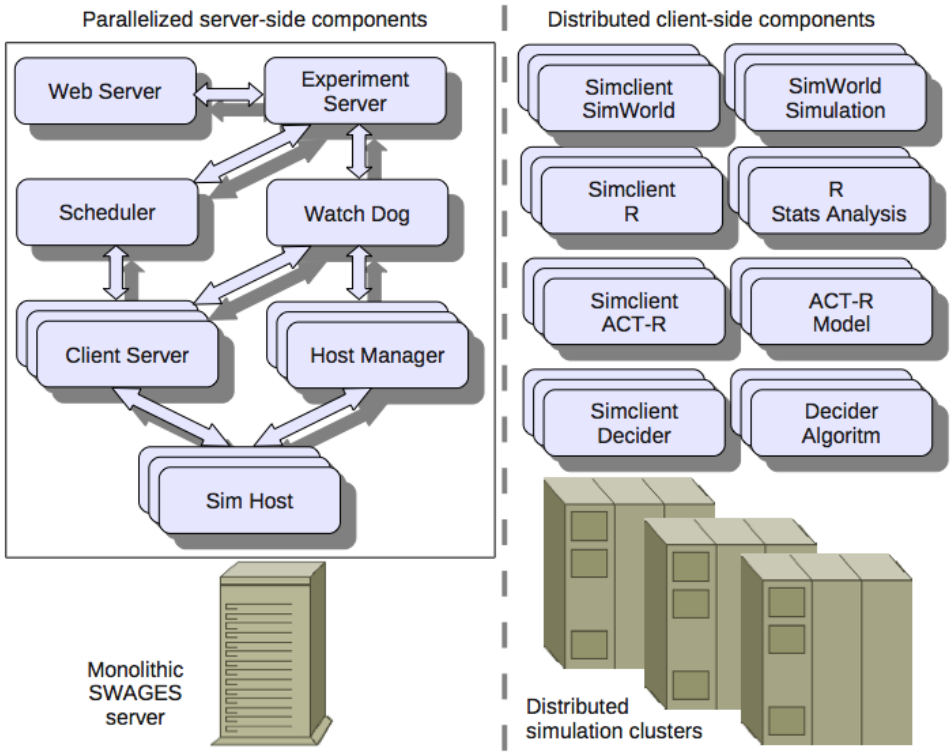
مولفه‌های سمت سرور وظیفه‌ی زمان‌بندی، توزیع، شروع و پایش چگونگی اجرای کلاینت‌های توزیع‌یافته‌ی شبیه‌سازی را بر عهده دارند. کاربران می‌توانند با استفاده از مولفه‌های این بخش، آزمایش‌های علمی خود را ارسال کنند، وضعیت آن‌ها را چک کنند و در نهایت نتایج این آزمایش‌ها را مشاهده نمایند. پارامترهای که توسط کاربران تعریف می‌شود شامل پارامترهای مدل که مجموعه شرایط اولیه‌ی آزمایش‌ها را در برمی‌گیرد و پارامترهای سیستم است. پارامترهای سیستم می‌تواند در برگیرنده‌ی اولویت آزمایش‌ها و پارامترهای زمان‌بندی آن‌ها برای اجرا روی میزبان‌ها، سطوح نظارت و پارامترهای بازیابی، فرمت داده‌های خروجی و مکان ذخیره‌ی آن‌ها، عملیات تحلیلی و آماری که بر روی نتایج خروجی انجام می‌گیرد و چگونگی اطلاع‌رسانی روند پیش‌رفت شبیه‌سازی به کاربر باشد.

در سمت کلاینت، مولفه‌ها از یک سو عملیات شبیه‌سازی را نمایش می‌دهند و از سوی دیگر برای به‌روز رسانی وضعیت فرآیند شبیه‌سازی به طور متناوب با سمت سرور ارتباط برقرار می‌کنند. وظیفه‌ی اصلی این مولفه‌ها، ذخیره و حفظ وضعیت شبیه‌سازی، و بررسی وضعیت اجرای آن با توجه به معیارهای تعریف‌شده توسط کاربران است (به بیان دقیق‌تر، بررسی این‌که با توجه به معیارهای تعریف‌شده توسط کاربران، امکان ادامه‌ی اجرای شبیه‌سازی بر روی میزبان کنونی وجود داشته باشد). همچنین بخش کلاینت دارای واسطی است که از طریق آن می‌تواند با محیط‌های شبیه‌سازی دیگری که از زبان‌های مانند جاوا، «آر»[۶۲]، «لیسپ»[۶۳] و «اِسکیم»[۶۴] استفاده می‌کند، به برقراری تعامل بپردازد.

برای افزایش قابلیت مقیاس‌پذیری سویجس، در نسخه‌ی بعدی آن تمام مولفه‌ها به صورت توزیع‌یافته پیاده‌سازی شدند تا از قابلیت‌هایی مانند تعادل بار خودکار و مدیریت میزبان‌ها، خطایابی و مکانیزم‌های دیگری از این دست پشتیبانی نماید. در نگاهی سطح بالاتر، این شبیه‌ساز از مولفه‌های مدیر، موتور، تحلیل‌گر[۶۵] و نمایان‌گر[۶۶] تشکیل یافته است (علاقمندان می‌توانند برای جزییات بیشتر به [۱۴] مراجعه نمایند). در شکل ۱۲شمای کلی از معماری این شبیه‌ساز و مولفه‌های آن در دو سمت سرور و کلاینت نمایش داده شده است.

---

[62] *R*
[63] *Lisp*
[64] *Scheme*
[65] *analyzer*
[66] *visualizer*



شکل ۱۲ – مدل معماری سویجس و مولفه‌های آن [۱۴]

مسولیت مدیر دریافت مشخصات آزمایش‌ها و پایان دادن به شبیه‌سازی بر اساس معیارهای تعریف شده است. موتور مشخصات شبیه‌سازی شامل جزییات اجرا و پارامترهای فضای ارزیابی را از موتور دریافت می‌کند و وظیفه دارد تا محیط شبیه‌سازی را بر روی مجموعه‌ای از گره‌های شبکه‌شده توزیع نماید و نتایج حاصل را تجمیع و یکی نماید. نتایجی که توسط موتور جمع‌آوری شده است به تحلیل-گر ارسال می‌شود تا بر روی داده‌ها تحلیل‌های آماری انجام گیرد. نمایان‌گر نیز در تکمیل عملیات تحلیل‌گر، با نمایش داده‌های پردازش شده امکان شناسایی پارامترها و روابط مهم، و همچنین بررسی بازده شبیه‌سازی را فراهم می‌کند.

۲-۸-۳- ری‌پَستاِچ‌پی‌سی

«ری‌پست‌اِچ‌پی‌سی»[۶۷] [۱۵و۱۶] یکی از مطرح‌ترین شبیه‌سازهای بزرگ مقیاسی است که در عملیات رایانشی به‌کار گرفته می‌شود. از آن‌جایی که محاسبات پربازده روی‌کرد مناسبی برای ارضای نیازمندی‌های شبیه‌سازهای بزرگ مقیاس به‌شمار می‌رود [۱۷]، در طراحی ری‌پست‌اِچ‌پی‌سی (همان-طور که از نام آن مشخص است) همین روی‌کرد در پیش گرفته شده است. چارچوب این شبیه‌ساز نیز مشابه فِلیم، مبتنی بر عامل است و ساختار ساده‌ی آن انعطاف‌پذیری بالایی را از خود نشان می‌دهد. علاوه بر این انعطاف‌پذیری، تمام ارتباطات بین فرآیندی و اشتراک اطلاعات مورد نیاز برای سنکرون‌سازی وضعیت عامل‌های شرکت کننده در فرآیندهای مختلف، در این شبیه‌ساز به صورت «شفاف»[۶۸] انجام می‌پذیرد.

ری‌پست‌اِچ‌پی‌سی محیط شبیه‌سازی چندسکویی است که مبتنی بر «سی++»[۶۹] طراحی شده است. این شبیه‌ساز برای مدل‌سازی‌های بزرگ مقیاسی که اندازه یا پیچیدگی آن‌ها بیش از توان یک پردازنده‌ی منفرد است توسعه داده شده است. از این‌رو تاکید آن بر روی اجراهای توزیع‌یافته‌ای است که به منظور آن، چندین فرآیند با استفاده از واسط‌های گذردهی پیغام (اِم‌پی‌آی) به تبادل و اشتراک عامل-ها با یک‌دیگر می‌پردازند. هر یک از فرآیندها مسوول اجرای رفتار زیر مجموعه‌ای از عامل‌هایی است که در کل شبیه‌سازی شرکت دارند. بنابراین هر فرآیند دارای یک زمان‌بند، یک زمینه و نقشه-های متناظر با آن زمینه است.

به طور کلی مولفه‌های اصلی و هسته‌ی ری‌پست‌اِچ‌پی‌سی عبارتند از:

- «شناسه‌ی عامل» که عددی است که به طور یکتا یکی از عامل‌های شبیه‌سازی را شناسایی می‌کند. این شناسه علاوه بر شماره‌ی شناسایی عامل، نوع و رتبه‌ی فرآیند آن را نیز مشخص می‌کند. به بیان دقیق‌تر، شماره‌ی شناسایی عامل در یک رتبه‌ی مشخص از فرآیندها منحصر به‌فرد است. رتبه‌ی فرآیند، شماره‌ی شناسایی فرآیندی است که عامل با آن متناظر است و برای هر فرآیند یکتاست. نوع عامل نیز عددی است که کلاسی از عامل‌ها با رفتار یکسان را مشخص می‌کند.
- یکی دیگر از مولفه‌های ری‌پست‌اِچ‌پی‌سی، «زمینه»[۷۰] نام دارد و در واقع کانتینر ساده‌ای است که عامل‌های با رفتارهای یکسان را در گروه‌های مختلف قرار می‌دهد. همچنین توسعه-دهنده‌ی شبیه‌ساز نیز، باید از طریق زمینه به ویژگی‌های مختلف عامل‌ها دسترسی یابد.

---

[67] *RepastHPC*
[68] *transparent*
[69] *C++*
[70] *context*

- مولفه‌ی دیگر «نقشه»[۷۱] است که به اعمال ساختاری می‌پردازد که در آن ساختار تمام عامل-ها سازمان‌دهی شده‌اند. این ساختار با استفاده از سمنتیک[۷۲] نقشه، روابط بین عامل‌ها را تعریف می‌کند. در ری‌پست‌اچ‌پی‌سی از سه نقشه‌ی گوناگون پشتیبانی می‌شود: «شبکه» که شامل مجموعه‌ای از گره‌ها و لینک‌های بین آن‌هاست و گرافی را مشخص می‌کند. «توری» که اطلاعات توپولوژی را در برمی‌گیرد و هر عامل ممکن است یک ماتریس یک، دو و یا سه بعدی را اشغال نماید. «فضای پیوسته» که اساساً توریی است که مختصات آن نقاط شناورند.
- «زمان‌بند» مولفه‌ای است که به عامل‌ها امکان زمان‌بندی روی‌دادها را می‌دهد. این مولفه مانع از تداخل کارها (به این معنا که پیش از پایان یافتن اجرای کار فعلی، کار جدید آغاز شود) در فرآیندها می‌شود و به این ترتیب اجرای هماهنگ و یکنواخت شبیه‌ساز را تضمین می‌نماید.

ری‌پست‌اچ‌پی‌سی به صورت محیطی کاملاً موازی طراحی شده است که در آن فرآیندهای زیادی به طور موازی با هم‌دیگر در حال اجرا هستند و حافظه بین آن‌ها به اشتراک گذاشته نشده است. عامل‌ها نیز به نوبه‌ی خود بین فرآیندها توزیع شده‌اند. هر فرآیندی مسوول عامل‌های محلی خود است. عامل-های محلی هر فرآیند، عامل‌هایی هستند که کد آن‌ها توسط آن فرآیند اجرا می‌شود. به منظور ایجاد تعامل میان عامل‌ها، ممکن است کپی عامل‌های غیر محلی نیز در فرآیند وجود داشته باشد (تا عامل-های محلی با آن‌ها تعامل کنند).

از آن‌جایی که این محیط شبیه‌سازی بر روی چندین فرآیند توزیع شده است، اغلب لازم است تا میان فرآیندها ارتباط ایجاد شده و اطلاعات وضعیت شبیه‌سازی فرآیندهای مختلف سنکرون شود. به بیان دقیق‌تر:

- هنگامی که فرآیندی به کپی‌هایی از عامل‌های فرآیند دیگری نیاز دارد
- و یا وقتی فرآیندی شامل عامل‌های غیر محلی کپی‌شده از فرآیندهای دیگری باشد که نیاز به به‌روز رسانی آن‌ها با آخرین وضعیت نسخه‌های اصلی باشد
- هم‌چنین هنگامی که توری و یا فضای بافرها نیاز به به‌روز رسانی داشته باشد
- و در نهایت وقتی که لازم باشد تا عاملی کاملاً از یک فرآیند به فرآیند دیگری انتقال داده شود

---

[71] *projection*
[72] *semantic*

لازم است تا ارتباطات و عملیات سنکرون‌سازی انجام گیرد. بیشتر عملیات مورد نیاز به صورت خودکار توسط خود شبیه‌ساز انجام می‌گیرد و تنها کافی است تا مدل‌ساز، کد برنامه‌های لازم برای استخراج وضعیت عامل‌ها، بسته‌بندی اطلاعات آن برای انتقال، باز نمودن بسته‌ی انتقال داده شده، و ایجاد یا به‌روز رسانی عامل‌های متناظر با اطلاعات دریافتی را در اختیار شبیه‌ساز قرار دهد. چگونگی باز و بسته کردن این بسته‌بندی در ری‌پست‌اچ‌پی‌سی، الگوی بسته‌بندی نامیده می‌شود و برای تمام حالت‌های سنکرون‌سازی اتفاق می‌افتد (ممکن است جزییات آن بسته به انواع سنکرون‌سازی متفاوت باشد).

ری‌پست‌اچ‌پی‌سی از زمان‌بندی زمان‌گسسته برای اجرای شبیه‌سازی استفاده می‌کند. هر فرآیند زمان-بندی گسسته‌ی خود را دارد و زمان‌بندی فرآیندها با هم سنکرون می‌شود (تمام فرآیندها تیک زمانی یکسانی را اجرا می‌کنند). همچنین هر فرآیند باید یک نمونه[۷۳] از زمینه‌ی مشترک را داشته باشد. زمینه‌ی مشترک شامل تمام عامل‌های محلی و غیر محلی فرآیند است و متدهای لازم برای ایجاد و بازیابی آن‌ها را داراست. هر عاملی که به زمینه‌ی مشترک اضافه می‌شود به طور خودکار به نقشه‌های متناظر با آن اضافه می‌شود. یادآوری می‌شود که نقشه‌ها، ساختار روابط اعضای زمینه‌ها را مشخص و اعمال می‌کنند.

در نهایت این شبیه‌ساز به دو روش به جمع‌آوری داده‌های تولید شده می‌پردازد. در روش اول مولفه‌ی جمع‌آوری داده، داده‌های فرآیندها را با استفاده از توابعی مانند مجموع، کمینه و بیشینه، تجمیع کرده و نتایج حاصل از منابع مختلف داده‌ها را به تفکیک، بر روی فایلی می‌نویسد. عمل‌کرد مولفه‌ی جمع-آوری داده در روش دوم مشابه روش اول است با این تفاوت که داده‌های فرآیندها تجمیع نمی‌شوند. بنابراین در فایل حاوی نتایج خروجی، رتبه‌ی فرآیندی که داده‌ها در آن تولید شده‌اند مشخص می-شود.

---

[73] *instance*

۳- ویژگی‌های کلی فنون و ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ مقیاس

در این بخش به بررسی و استخراج ویژگی‌های کلی فنون و ابزارهای شبیه‌سازی که در عملیات رایانشی بزرگ‌مقیاس کاربرد دارند می‌پردازیم. در طبقه‌بندی ارایه شده در بخش پیش، این فنون و ابزارها را به دو دسته‌ی کلی تقسیم کردیم. در راستای همین طبقه‌بندی، در این بخش ابتدا تاکید خود را بر روی ویژگی‌های مشترک بین این دو دسته قرار خواهیم داد و سپس در ادامه به ویژگی‌های خاص فنون و ابزارهای مطرح در هر یک از این دسته‌ها می‌پردازیم.

یکی از ویژگی‌های مهمی که از بررسی شبیه‌سازهای عملیات رایانشی بزرگ مقیاس قابل استخراج است و می‌توان آن را از ویژگی‌های مشترک بین هر دو دسته در نظر گرفت، بار کاری پردازشی و چگونگی مدل‌سازی آن در محیط شبیه‌سازی است. همان‌طور که در بخش پیش بررسی کردیم، رایانش توری و رایانش ابری به عنوان جدیدترین و اصلی‌ترین فناوری‌های مطرح برای انجام عملیات رایانشی بزرگ مقیاس به‌شمار می‌روند، و از این‌رو، بسیاری از فنون و ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ مقیاس، بر مبنای این دو فناوری توسعه داده شده‌اند. ویژگی هر دو دسته از شبیه‌سازها که مبتنی بر این دو فناوری هستند (چه دسته‌ی اول که به شبیه‌سازی محیط‌های رایانش توری و ابری، و انجام عملیات رایانشی در آن‌ها می‌پردازند و چه دسته‌ی دوم که بر روی بسترهای رایانش توری و ابری به شبیه‌سازی عملیات رایانشی بزرگ‌مقیاس می‌پردازند) از نظر بار کاری پردازشی تا حدی متفاوت است.

در عملیات رایانش توری، بار کاری به صورت دنباله‌ای از کارها مدل‌سازی می‌شوند که هر یک از این کارها به نوبه‌ی خود به تعدادی وظیفه‌ی عملیاتی شکسته می‌شوند. وظایف ممکن است از یک-دیگر مستقل باشند و یا به یک‌دیگر وابستگی داشته باشند به این ترتیب که خروجی یک وظیفه برای شروع اجرای وظیفه‌ی دیگر مورد نیاز باشد. علاوه بر این، در عملیات رایانش توری بزرگ مقیاس، بنا بر ماهیتی که دارند، تعداد کارها بیشتر از میزان منابع محاسباتیِ در دسترس است. هدف اصلی در چنین عملیاتی، کمینه کردن زمان مورد نیاز برای پردازش تمام کارهاست (که ممکن است هفته‌ها و یا

ماه‌ها به طول انجامد)، و این در حالی است که محدودیت اکیدی برای پایان یافتن هیچ‌یک از کارها (به طور مجزا) وجود ندارد.

در مقابل در عملیات رایانش ابری بزرگ مقیاس، تقاضاهای دریافتی اغلب برای کاربردهایی مانند مرور وب، «ارسال بلادرنگ پیغام»[۷۴] و کاربردهای مختلف «تحویل محتوا» تولید شده‌اند. کارهای لازم برای سرویس‌دهی به این تقاضاها، معمولا استقلال بیشتری از هم دارند و حجم بار کاری کمتری ایجاد می‌کنند. اما در مقابل معمولا محدودیت اکیدی برای پایان یافتن این کارها وجود دارد که باید «توافق‌نامه‌ی کیفیت سرویس»[۷۵] مشخص‌شده برای آن‌ها را ارضا نماید. بنابراین تفاوت در این الگوهای بار کاری و نیازمندی‌های مطرح در مدل‌سازی هر یک از آن‌ها، باید در فنون و ابزارهای شبیه‌سازی عملیات رایانشی بزرگ مقیاس (در هر دو دسته) لحاظ شده و توسط قابلیت‌های این فنون و ابزارها، مورد پشتیبانی قرار گیرد.

از جمله دیگر ویژگی‌هایی که هر دو دسته از فنون و ابزارهای شبیه‌سازی عملیات رایانشی بزرگ مقیاس در آن‌ها با یک‌دیگر اشتراک دارند، می‌توان به دو مورد زیر اشاره نمود:

- رفتار شبیه‌سازی: این ویژگی کلی، فنون و ابزارهای شبیه‌سازی را بر حسب چگونگی انجام و پیش‌روی فرآیند شبیه‌سازی، از یک‌دیگر متمایز می‌سازد. به این ترتیب که برخی از شبیه‌سازها در دسته شبیه‌سازهای «قطعی» قرار می‌گیرند که در این دسته، هیچ رویداد تصادفی در طول فرآیند شبیه‌سازی اتفاق نمی‌افتد. از این‌رو، تکرار فرآیند شبیه‌سازی همواره منجر به نتایج یکسانی خواهد شد. در مقابل، دسته‌ی دیگری از شبیه‌سازها «تصادفی» هستند که در نتیجه‌ی لحاظ رویدادهای تصادفی در طول فرآیند شبیه‌سازی، منجر به نتایج متفاوت و غیرقابل پیش‌بینی می‌شوند (از این‌رو فرآیند شبیه‌سازی باید بر حسب نیاز به تعداد دفعات مناسب تکرار شود).
- شیوه مقداردهی: این ویژگی کلی، چگونگی مقداردهی به مولفه‌های شرکت‌کننده‌ی موجود در فرآیند شبیه‌سازی را مشخص می‌کند. به بیان دقیق‌تر، بر اساس این ویژگی، دسته‌ای از شبیه‌سازها «گسسته» و دسته‌ای دیگر «پیوسته» هستند. در شبیه‌سازهای گسسته، «متغیرهای حالت» تنها در تعداد محدود و قابل شمارشی از نقاط زمانی تغییر مقدار می‌یابند. هر یک از این تغییرات (و یا در واقع تعیین این نقاط زمانی)، در نتیجه‌ی رخ دادن رویدادی

---

[74] *Instant messaging*
[75] *SLA (service level agreement)*

در طول فرآیند شبیه‌سازی است. در مقابل، در شبیه‌سازهای پیوسته، مقادیر متغیرهای حالت به صورت پیوسته و بدون جهش تغییر می‌کند (به تعداد نامتناهی حالت وجود دارد).

علاوه بر ویژگی‌های ذکر شده، که از ویژگی‌های مشترک فنون و ابزارهای شبیه‌سازی در عملیات رایانشی بزرگ مقیاس به شمار می‌روند، دسته‌ای دیگر از ویژگی‌ها به طور خاص از بررسی شبیه-سازهای عملیات رایانشی بزرگ مقیاس، در هر یک از دو دسته‌ی طبقه‌بندی، قابل استخراج هستند که در ادامه به تفکیک هر دسته به این ویژگی‌ها خواهیم پرداخت.

دسته‌ی اول از فنون و ابزارهای شبیه‌سازی که در عملیات رایانشی بزرگ مقیاس مورد استفاده قرار می‌گیرند، در برخی از ویژگی‌ها با دسته‌ی دوم تفاوت دارند. این ویژگی‌ها عبارتند از:

- از جمله ویژگی‌های اصلی که فنون و ابزارهای شبیه‌سازی دسته‌ی اول را از یک‌دیگر متمایز می‌سازد، می‌توان به نحوه‌ی تعریف موجودیت‌های شبیه‌سازی شده اشاره کرد. ابزارهای شبیه‌سازهای مطرح در عملیات رایانشی بزرگ مقیاس، اگرچه در دارا بودن این ویژگی با سایر شبیه‌سازها اشتراک دارند، اما در اغلب آن‌ها باید مجموعه‌ی مشخصی از عناصر و موجودیت‌ها تعریف شوند. عناصری که منابع رایانشی را میزبانی می‌کنند، شبکه‌ای که این عناصر از طریق آن با یک‌دیگر در ارتباطند و همچنین کاربردهای که در عملیات رایانشی مورد استفاده قرار می‌گیرند، از اعضای اصلی این مجموعه به‌شمار می‌روند که باید مشخصات آن‌ها در فرآیند شبیه‌سازی تعریف شود.
- این دسته از فنون و ابزارها قابلیت کنترل شرایط آزمایش‌شده در فرآیند شبیه‌سازی را دارا هستند. اهمیت و ضرورت ویژگی کنترل‌پذیری این است که امکان ارزیابی عمل‌کرد هر بخش از فرآیند شبیه‌سازی را به طور مستقل فراهم می‌سازد و بنابراین می‌توان با آزمودن سناریوهای مختلف و شرایط متفاوت در محیط شبیه‌سازی، به محدودیت‌های مدل‌ها و راه-کارهای پیشنهادی مورد آزمایش پرداخت. بدیهی است که پشتیبانی کردن از این ویژگی به نوبه‌ی خود نیازمند آن است که محیط شبیه‌سازی از قابلیت پیکربندی مناسبی برخوردار باشد.
- یکی دیگر از ویژگی‌هایی که فنون و ابزارهای این دسته را از هم متمایز می‌سازد، نحوه‌ی گزارش نتایج خروجی فرآیند شبیه‌سازی است که می‌تواند در سطوح انتزاعی متفاوتی صورت گیرد. به بیان دقیق‌تر، برخی از ابزارها ممکن است در خروجی خود تنها امکان مقایسه در سطح کیفی را فراهم آوردند که در نتیجه‌ی آن یک راه‌کار نسبت به راه‌کارهای دیگر مورد آزمایش بهتر شناخته شود. همچنین ممکن است خروجی برخی دیگر از

ابزارهای شبیه‌سازی کمی باشد و بنابراین با گزارش ارزیابی‌ها به صورت کمی، نتایج قابل اندازه‌گیری و مقایسه باشند.

دسته‌ی دوم از شبیه‌سازهایی که در بخش پیش مورد بررسی قرار گرفتند، تحت عنوان فنون و ابزارهای شبیه‌سازی بزرگ مقیاس مطرح شدند. علاوه بر ویژگی‌های مشترک با دسته‌ی اول که پیش‌تر به آن‌ها اشاره شد، بیشتر محیط‌های شبیه‌سازی که در این دسته قرار می‌گیرند از ویژگی‌های مشابه دیگری نیز برخوردار هستند که برخی از این ویژگی‌ها عبارتند از:

- این شبیه‌سازها معمولاً به طور خودکار بر روی چندین میزبان توزیع می‌شوند. بنابراین یکی از ویژگی مهم آن‌ها مقیاس‌پذیری[۷۶] است. این ویژگی مهم باعث می‌شود تا این شبیه‌سازها بازده بالایی داشته باشند، و به سادگی بتوانند تعداد زیادی از شبیه‌سازی‌های موازی را اجرا نمایند.
- این فنون و ابزارها به استفاده از محیط‌های رایانشی ناهم‌گونی می‌پردازند. با دارا بودن این ویژگی، این شبیه‌سازها می‌توانند برای مثال بر روی ترکیبی از خوشه‌های محاسباتی تخصیص یافته و مجموعه‌ای از میزبان‌های مستقل[۷۷] اجرا شوند. این در حالی است که نیازی به نصب هیچ‌گونه نرم‌افزاری بر روی این میزبان‌ها وجود ندارد و تنها کافی است امکان دسترسی به آن‌ها فراهم باشد.
- این فنون و ابزارها می‌توانند شبیه‌سازی‌ها را بر اساس میزان منابع محاسباتی در دسترس، موازی‌سازی کنند. بنابراین یکی از ویژگی‌های آن‌ها هم‌زمان‌سازی[۷۸] اجرای فرآیندهای شبیه‌سازی است. اهمیت این ویژگی زمانی است که بستر محیط‌های شبیه‌سازی از موازی‌سازی (شامل الگوریتم‌های زمان‌بندی سنکرون و آسنکرون) پشتیبانی کند (استفاده از منابع پویا به منظور بیشینه شدن بازده).
- این شبیه‌سازها معمولا در نهایت منجر به تولید حجم بالایی از داده‌های خروجی می‌شوند و از این‌رو اغلب امکان تحلیل‌های آماری مختلفی را بر روی این داده‌ها فراهم می‌نمایند. از

---

[76] *Scalability*
[77] *Stand-alone*
[78] *Concurrency*

جمله‌ی این تحلیل‌ها می‌توان به برازش مدل (بر اساس معیارهای برازش) و «اکتشاف مدل»[۷۹] (بر اساس محدودیت‌های مدل) اشاره کرد.

- اغلب این شبیه‌سازها دارای مکانیزم‌هایی هستند که از پایان‌پذیر بودن فرآیند شبیه‌سازی اطمینان حاصل می‌کنند و هم‌چنین در پایان این فرآیند، امکان «تجسم»[۸۰] و نمایش مجموعه دادگانی که در نتیجه‌ی شبیه‌سازی‌های بزرگ مقیاس حاصل می‌شوند را فراهم می‌نمایند.

۳-۱- حوزه‌ی محوری در فاز دوم

ما در فاز دوم این پژوهش قصد داریم تاکید اصلی خود را بر روی «آموزش و یادگیری الکترونیکی» به عنوان یکی از حوزه‌های خدمات رایانشی پژوهشگاه قرار دهیم. علت اصلی انتخاب این حوزه در فاز دوم پژوهش این است که با توجه به فنون و ابزارهای شبیه‌سازی بررسی شده در این بخش، این حوزه برای ادامه‌ی پژوهش مناسب‌تر شناخته شده و از انعطاف‌پذیری بالاتری برای تعریف سناریوهای مختلف در شبیه‌سازی عملیات رایانشی بزرگ‌مقیاس برخوردار است. البته باید توجه داشت که در حالت کلی دامنه‌ی تنوع کاربردهایی که منجر به عملیات رایانشی بزرگ مقیاس می‌شوند بسیار گسترده است. همان‌طور که پیش‌تر اشاره شد، از جمله‌ی این کاربردها می‌توان به «انجام محاسبات علمی و پربازده»، «شبکه‌های اجتماعی»، «آموزش و یادگیری الکترونیکی»، «ارایه‌ی طرح‌ها و پایان-نامه‌های الکترونیکی»، و «کاربردهای وب» اشاره کرد که همگی از حوزه‌های خدمات رایانشی پژوهشگاه هستند که پتانسیل بالایی برای گسترش بر روی بسترهای رایانشی بزرگ مقیاس دارند.

از این رو می‌توان در راستای این پژوهش، در طرح‌های پژوهشی جداگانه‌ی دیگری با انتخاب هر یک از این حوزه‌ها به عنوان حوزه‌ی محوری، رویه‌ای مشابه روند در پیش گرفته شده در فاز دوم این پژوهش را در حوزه‌های دیگر نیز انجام داد. به بیان دقیق‌تر، صرف‌نظر از تنوع و گوناگونی کاربردهای اشاره شده، می‌توان نیاز به مقیاس‌پذیری بالا و دسترسی بالا به منابع رایانشی را به عنوان نیازمندی مشترک برای تمام آن‌ها در نظر گرفت. به بیان دیگر، رفتار تمام این کاربردها با مشخص کردن میزان نیازمندی آن‌ها به منابع رایانشی قابل توصیف و مدل‌سازی هستند. از این‌رو صرف‌نظر از این که در ادامه‌ی این طرح پژوهشی و یا در طرح‌های پژوهشی دیگر، کدام یک از این حوزه‌های کاربردی عملیات رایانشی بزرگ مقیاس مورد تاکید قرار خواهد گرفت، چگونگی به‌کارگیری

---

[79] *Model discovery*
[80] *visualization*

ابزارها و فنون شبیه‌سازی تفاوت چندانی نخواهد داشت. تفاوت اصلی میان حوزه‌های مختلف، در سناریوهای شبیه‌سازی قابل تعریف در آن‌هاست. بنابراین در صورت تعریف سناریوهای مشابه، نحوه-ی به‌کارگیری این فنون و ابزارها در شبیه‌سازی هر یک از این حوزه‌های کاربردی، به سادگی و با اندکی تغییرات، قابل اعمال به بسیاری از حوزه‌های دیگر نیز خواهد بود.

٤- شناسایی مجموعه نیازمندی‌ها

در بخش دوم این طرح پژوهشی، فنون و ابزارهای شبیه‌سازی که به طور خاص در عملیات رایانشی بزرگ‌مقیاس به‌کار گرفته می‌شوند را بر اساس «نحوه‌ی به‌کارگیری» به دو دسته‌ی کلی تقسیم کردیم، ۱) فنون و ابزارهایی که بر مبنای شبیه‌سازی محیط‌های رایانشی بزرگ‌مقیاس توسعه داده شده‌اند و ۲) فنون و ابزارهای شبیه‌سازی که بر روی بسترهای رایانشی بزرگ‌مقیاس قابل به‌کارگیری هستند. هر دو دسته از این فنون و ابزارها در شبیه‌سازی سناریوهای مختلف در حوزه‌ی «آموزش و یادگیری الکترونیکی»، که به عنوان یکی از حوزه‌های خدمات رایانشی پژوهشگاه مورد تاکید فاز دوم پژوهش ما قرار دارد، قابل به‌کارگیری هستند. از این رو پیش از آن که در بخش بعدی به چگونگی به‌کارگیری هر یک از این دو دسته فنون و ابزارها در در حوزه‌ی «آموزش و یادگیری الکترونیکی» بپردازیم، در این بخش ابتدا نیازمندی‌های فنی لازم برای به‌کارگیری آن‌ها را به تفکیک مورد بررسی قرار خواهیم داد.

دسته‌ی نخست فنون و ابزارها که به شبیه‌سازی بسترهای رایانشی مناسب برای انجام عملیات رایانشی بزرگ‌مقیاس، مانند محیط‌های رایانش توری و رایانش ابری می‌پردازند، از جنبه‌ی سخت‌افزاری نیازمندی خاص و فراتر از ماشین‌های پردازشی معمولی ندارند. به بیان دقیق‌تر، با توجه به پیشرفت‌های اخیری که در منابع پردازشی و ذخیره‌سازی کامپیوترهای شخصی ایجاد شده است، این دسته از فنون و ابزارهای شبیه‌سازی بر روی ماشین‌های پردازشی با مشخصات فنی متداول قابل اجرا هستند.
ابزارهای شبیه‌سازی که در دسته‌ی نخست در این پژوهش مطرح شدند، مانند گریدسیم، اُپُتَرسیم، کلُدسیم و گرین‌کلد از جنبه‌ی حقوقی از جمله نرم‌افزارهای «آزاد»[۸۱] و «متن‌باز»[۸۲] هستند. به بیان دقیق‌تر، طبق تعریف ارایه شده توسط بنیاد نرم‌افزارهای آزاد [۳۸]، این ابزارها از لحاظ نرم‌افزاری، آزادی‌های زیر را برای کاربران خود فراهم می‌کنند:

- کاربران باید اجازه داشته باشند که نرم‌افزار مورد نظر را برای هر قصد و منظوری اجرا کنند.

---

[81] *free*
[82] *open-source*

- کاربران باید اجازه داشته باشند کدهای منبع نرم‌افزار را مطالعه کرده و آن را مطابق با نیازهای خود تغییر دهند. برای رسیدن به این هدف، کدهای منبع نرم‌افزار باید در اختیار کاربران قرار گیرد.
- کاربران باید اجازه داشته باشند نرم‌افزار را مجددا منشتر کرده و در اختیار دیگران قرار دهند. این کار می‌تواند به صورت رایگان صورت گیرد.

همچنین طبق تعریف موسسه پیشگامان متن‌باز، نرم‌افزار متن‌باز تنها به معنی در دسترس ساختن کد منبع نیست. بلکه علاوه بر آن مجوز، باید ویژگی‌های زیر را هم داشته باشد [۳۹]:

- نرم‌افزار باید قابل توزیع مجدد باشد.
- نرم‌افزار باید شامل کد منبع باشد و این کد منبع را باید بتوان تغییر داد و مجددا منتشر کرد.
- مجوز نباید در برابر افراد یا گروه خاصی تبعیض قائل شود.
- مجوز نباید کاربر را برای رسیدن به یک هدف خاص محدود کند.
- مجوز نباید مختص به یک محصول خاص باشد.
- مجوز نباید نرم‌افزارهای دیگری که به همراه نرم‌افزار مورد نظر عرضه شده‌اند را محدود کرده و تحت تاثیر قرار دهد.
- مجوز نباید تکنولوژی خاصی را محدود کند.

بنابراین با توجه به ویژگی‌های حقوقی نرم‌افزارهای آزاد و متن‌باز، ابزارهای شبیه‌سازی عنوان شده، بدون آن که نیازمندی خاصی از جنبه‌ی نرم‌افزاری داشته باشند قابل به‌کارگیری هستند. به طور کلی سیاست مهمی که در این پژوهش در پیش گرفته شده به این ترتیب بوده است که فنون و ابزارهای شبیه‌سازی که در هر یک از دو دسته انتخاب و مطرح شده‌اند از لحاظ نرم‌افزاری آزاد و متن‌باز باشند تا به این ترتیب بتوان با اجتناب از بسیاری از محدودیت‌های عملی که برای ابزارهای انحصاری وجود دارد آن‌ها را به‌کار گرفت.

در مقابل دسته‌ی اول که تا اینجا به بررسی نیازمندی‌های فنی آن‌ها از جنبه‌های مختلف نرم‌افزاری و سخت‌افزاری پرداختیم، دسته‌ی دوم که ما در این پژوهش آن‌ها را فنون وابزارهای شبیه‌سازی بزرگ-مقیاس نامیدیم از نیازمندی‌های به نسبت پیچیده‌تری برخوردارند (یادآوری می‌شود که این دسته،

ابزارهایی را در بر می‌گیرد که به منظور انجام عملیات رایانشی بزرگ‌مقیاس، قابلیت به‌کارگیری و توسعه بر روی چنین بسترهایی را دارند).

از جنبه‌ی سخت‌افزاری، این دسته از فنون و ابزارهای شبیه‌سازی نیازمند محیط‌های رایانشی توزیع‌شده هستند. به بیان دقیق‌تر معماری آن‌ها به گونه‌ای طراحی شده است که بر روی چندین رایانه‌ی خودکار که توسط یک شبکه‌ی رایانه‌ای با یک‌دیگر در ارتباط هستند، اجرا شوند. با پشتیبانی از این قابلیت مهم و اساسی است، که این دسته از شبیه‌سازها امکان انجام عملیات رایانشی بزرگ‌مقیاس را فراهم می‌سازند. به منظور تحقق یافتن این قابلیت و اجرا شدن این ابزارها بر روی خوشه‌های بزرگ سخت-افزاری، مساله‌ی جدیدی مطرح می‌شود و آن چگونگی توزیع وظایف و عملیات رایانشی در شبکه‌ی سخت‌افزاری است. بنابراین برای به‌کار گیری این دسته از شبیه‌سازها لازم است تا به گونه‌ای، به رفع این نیازمندی پرداخته شود.

نیاز به توزیع عملیات رایانشی بر روی مجموعه‌ای از رایانه‌هایی که روی هم رفته به عنوان یک خوشه‌ی سخت‌افزاری شناخته می‌شوند را می‌توان به عنوان نیازمندی میان افزاری دسته دوم از شبیه سازها در نظر گرفت. یکی از متداول‌ترین راه‌کارها برای رفع این نیازمندی، استفاده از چارچوب و مدل برنامه‌نویسی «نگاشت‌کاهش» است که توسط شرکت گوگل به طور خاص به منظور پشتیبانی از رایانش توزیع‌شده ارایه شده است [۴۰]. این مدل با الهام‌گیری از نگاشت و کاهش که از مفاهیم به-کار رفته در زبان‌های برنامه‌نویسی تابعی هستند، ایجاد شده است [۴۱] و ساختار منطقی دو گام آن به این ترتیب است که:

- در گام نگاشت، گره اصلی[۸۳] ورودی را به قطعاتی کوچک‌تر تقسیم می‌نماید (تقسیم مساله‌ی بزرگ به مسایل کوچک) و سپس این مسایل کوچک (زیر مسایل) بین گره‌های کارگر[۸۴] تقسیم می‌شود. یک گره کارگر نیز ممکن است این عملیات را به نوبه‌ی خود تکرار نماید، و به این ترتیب ساختاری درختی و چند مرحله‌ای ایجاد شود. هر گره کارگر زیرمساله‌ی خود را حل نموده و نتیجه را به گره اصلی خود برمی‌گرداند.
- در گام کاهش، گره اصلی جواب زیرمسایل را از گره‌های کارگرش گرفته و خروجی را می‌سازد تا خروجی نهایی، که حل مساله‌ی ورودی است، را ایجاد نماید.

---

[83] *Master*
[84] *Slave*

با استفاده از چارچوب پردازشی نگاشت‌کاهش می‌توان نیاز دسته دوم از شبیه‌سازها به توزیع بر روی خوشه‌های سخت‌افزاری را برطرف ساخت. در بخش بعد به نحوه‌ی دقیق طراحی این مدل در به‌کارگیری فنون و ابزارهای شبیه‌سازی خواهیم پرداخت.

در نهایت از جنبه‌ی نرم‌افزاری، همان‌طور که پیش‌تر (به عنوان سیاست کلی در پیش گرفته شده در این پژوهش) بیان شد، فنون و ابزارهای شبیه‌سازی مطرح شده در دسته‌ی دوم، مشابه دسته‌ی اول، پشتیبان نرم‌افزارهای آزاد و متن‌باز هستند. از این رو بدون داشتن نیازمندی‌های حقوقی محدود کننده در نرم‌افزارهای انحصاری، به آسانی قابل به‌کارگیری و توسعه خواهند بود.

بنابراین با برآورده نمودن انواع نیازمندی‌های مطرح شده در این بخش – در جنبه‌های مختلف سخت‌افزاری، نرم‌افزاری و یا میان‌افزاری – می‌توان فنون و ابزارهای موردنظر در هر یک از دو دسته‌ی مذکور را در شبیه‌سازی عملیات رایانشی بزرگ‌مقیاس در حوزه‌های مختلف از جمله حوزه‌ی آموزش و یادگیری الکترونیکی به‌کار گرفت. در بخش بعد، با جزییات بیشتر به تبیین چگونگی به‌کارگیری این فنون و ابزارها در تحلیل و ارزیابی سناریوهای متداول در انجام عملیات رایانشی بزرگ‌مقیاس در حوزه‌ی آموزش و یادگیری الکترونیکی خواهیم پرداخت.

۵- تبیین چگونگی به‌کارگیری فنون و ابزارها

همان‌طور که در فاز نخست (بخش سوم این گزارش) اشاره شده، ما در این پژوهش تاکید اصلی خود را بر روی «آموزش و یادگیری الکترونیکی» به عنوان یکی از حوزه‌های خدمات رایانشی پژوهشگاه قرار داده‌ایم. از سوی دیگر، در نتیجه‌ی بررسی فنون و ابزارهای شبیه‌سازی که در عملیات رایانشی بزرگ‌مقیاس به‌کار گرفته می‌شوند، آن‌ها را به دو دسته‌ی کلی تقسیم‌بندی کردیم. از این رو در این بخش قصد داریم تا به چگونگی به‌کارگیری فنون و ابزارهای هر یک از این دو دسته در تحلیل و ارزیابی سناریوهای مختلف در حوزه‌ی خدماتی «آموزش و یادگیری الکترونیکی» بپردازیم.

به این منظور، در راستای تقسیم‌بندی ارایه شده، این بخش در دو زیربخش جداگانه تنظیم شده است. در زیربخش نخست (۵-۱)، سناریوهایی را تحلیل و ارزیابی خواهیم کرد که به توسعه‌ی سیستم یادگیری و آموزش الکترونیکی در محیط‌های رایانشی بزرگ‌مقیاس می‌پردازند (پوشش دسته‌ی اول از فنون و ابزارها و تبیین چگونگی به‌کارگیری آن‌ها). در زیربخش دوم (۵-۲) نیز به طور خاص به چگونگی به‌کارگیری فنون و ابزارهای بزرگ‌مقیاس در ارایه‌ی سرویس‌های هوشمند یادگیری و آموزش الکترونیکی – که از جمله سرویس‌های مهمی هستند که ارایه‌ی آن‌ها تنها با انجام عملیات رایانشی بزرگ‌مقیاس امکان‌پذیر است- خواهیم پرداخت (پوشش دسته‌ی دوم از فنون و ابزارها و تبیین چگونگی به‌کارگیری آن‌ها).

۵-۱- توسعه‌ی سیستم یادگیری و آموزش الکترونیکی در محیط‌های رایانشی بزرگ‌مقیاس

در این زیربخش، به تبیین چگونگی به‌کارگیری فنون و ابزارهای دسته‌ی اول در تحلیل و ارزیابی سناریوهای مختلفی که در توسعه‌ی سیستم‌های یادگیری الکترونیکی در محیط‌های رایانشی بزرگ-مقیاس مطرح می‌شود خواهیم پرداخت. به این منظور ابتدا نحوه‌ی طراحی سیستم‌های یادگیری الکترونیکی در بسترهای رایانش ابری که یکی از نمونه‌های بارز محیط‌های رایانشی بزرگ‌مقیاس به-

شمار می‌روند را مورد بحث قرار خواهیم داد[۸۵]. سپس با استفاده از ابزار کلُدسیم[۸۶] (که یکی از ابزارهای عنوان شده در دسته‌ی اول است) به تحلیل و ارزیابی سناریوهای مختلف برای توسعه‌ی سیستم‌های یادگیری الکترونیکی در بسترهای رایانش ابری خواهیم پرداخت[۸۷].

از میان فناوری‌های مختلفی که برای آموزش و یادگیری وجود دارند، «یادگیری مبتنی بر وب» مزایای بسیار زیادی را نسبت به «یادگیری مبتنی بر کلاس درس» دربردارد. یکی از این مزایای مهم، کاهش هزینه‌های یادگیری است. عدم نیاز به محیط‌های فیزیکی برای آموزش و یادگیری، هزینه‌ها را کاهش داده و امکان آن را در هر زمان و مکانی که یادگیرنده تمایل داشته باشد فراهم می‌سازد. علاوه بر این، فرد آموزش‌دهنده می‌تواند به راحتی مواد آموزشی را بروز رسانی کند و با مشارکت در ارایه‌ی محتوای چندرسانه‌ای، ضمن ایجاد محیطی دوستانه، درک مفاهیم را برای افراد یادگیرنده ساده‌تر سازد.

این روش یادگیری را می‌توان یک روی‌کرد «یادگیرنده محور»[۸۸] دانست که با وجود مزایای اشاره شده، چالش‌هایی جدی نیز به همراه دارد [۴۲،۴۳]. در حال حاضر، سیستم‌های آموزش الکترونیکی، در سطح زیرساخت، از مقیاس‌پذیری بسیار کمی برخوردارند. بسیاری از منابع، اختصاص‌یافته هستند و تنها به منظورهای خاصی می‌توانند بکار گرفته شوند. بنابراین هنگام افزایش بار سیستم، باید منابع جدیدی از همان نوع، اضافه و پیکربندی شوند که این مساله هزینه‌های مدیریت منابع را به شدت افزایش می‌دهد.

یکی دیگر از چالش‌های کلیدی، مساله‌ی بهره‌وری کارا از منابع است. برای مثال در کاربردهای متداول دانشگاهی، اغلب سیستم‌ها و سرورها در طول شب و یا در تعطیلات بین ترم، میزان بهره‌وری پایینی دارند. این در حالی است که همین منابع در زمان‌های دیگری مانند اواخر ترم، به شدت مورد نیازند. بنابراین چاره‌ای جز نگهداری ماشین‌هایی که بسیاری مواقع بیکارند و هدر دادن پتانسیل آن‌ها وجود نخواهد داشت.

---

[۸۵] مطالبی که در این مرحله به آن‌ها اشاره می‌شود برگرفته از یکی از مقاله‌های پژوهشگر این طرح است که با عنوان «طراحی سیستم یادگیری الکترونیکی مبتنی بر مدل‌های رایانش ابری» در مجموعه مقالات هفتمین کنفرانس ملی و چهارمین کنفرانس بین‌المللی یادگیری و آموزش الکترونیکی ایران به چاپ رسیده است.

[86] *CloudSim*

[۸۷] مطالبی که در این مرحله به آن‌ها اشاره می‌شود برگرفته از یکی از مقاله‌های «علمی-پژوهشی» ارسال شده توسط پژوهشگر این طرح است که با عنوان «ارزیابی کارایی مدل‌های رایانش ابری در ارایه سرویس‌های یادگیری الکترونیکی» در حال حاضر در فاز نهایی داوری است.

[88] *Learner-centered*

در نهایت باید علاوه بر این چالش‌ها، هزینه‌های دیگری که باید برای ارایه‌ی سیستم یادگیری الکترونیکی پرداخته شوند، مانند هزینه‌های نگهداری سایت و سیستم‌های کامپیوتری، و هزینه‌های نصب و پشتیبانی فنی از بسته‌های نرم‌افزاری، را نیز در نظر گرفت [۴۴]. همان‌طور که در ادامه بحث خواهد شد، آموزش و یادگیری الکترونیکی در بسترهای رایانش ابری - که نمونه‌ی بارزی از محیط-های رایانشی بزرگ‌مقیاس به‌شما می‌روند - راه‌کار مناسبی برای پرداختن به مسایل و چالش‌های مطرح شده است.

رایانش ابری مبتنی بر معماری سرویس‌گرا است [۴۵،۴۶]. سرویس‌هایی که در اختیار کاربران رایانش ابری قرار می‌گیرد معمولا در سه رده‌ی اصلی قابل دسته‌بندی هستند که این دسته‌ها تحت عنوان مدل-های سرویس رایانش ابری نیز شناخته می‌شوند . این سه مدل در واقع سطوح انتزاعی سرویس‌های رایانش ابری را توصیف می‌کنند و عبارتند از:

- «نرم‌افزار به عنوان سرویس»[۸۹] (یا به طور مخفف سَس): در این مدل، فراهم‌کننده‌ی سرویس ابر مسوولیت اجرا و نگهداری کاربردهای نرم‌افزاری و زیرساخت و بستر رایانشی را بر عهده دارد. از دید کاربر رایانش ابری، چنین سرویسی یک «واسط کاربرد»[۹۰] مبتنی بر وب دارد که از طریق آن سرویس‌ها و کاربردهای نرم‌افزاری به طور کامل بر روی شبکه‌ی اینترنت ارایه شده و از طریق مرورگر وب کاربر قابل دسترسی هستند (مانند جی‌میل[۹۱] و گوگل-داکس[۹۲]). کاربران می‌توانند با روش‌ها و وسایل مختلفی مانند کامپیوتر ثابت یا سیار و یا گوشی همراه به کاربردهای میزبانی شده دسترسی یابند. این مدل سرویس نسبت به مدل‌های رایج ارایه‌ی نرم‌افزار دارای این مزیت است که کاربران نیازی به خرید مجوز، نصب، بروزرسانی، نگهداری و اجرای نرم‌افزار بر روی پایانه‌ی خود ندارند و مهم‌تر این که از مزایای مهمی مانند قابلیت پیکربندی و مقیاس‌پذیری بالا نیز برخوردارند.
- «بستر به عنوان سرویس»[۹۳] (یا به طور مخفف پَس): در این مدل، فراهم‌کننده‌ی سرویس ابر مسوولیت اجرا و نگهداری سیستم نرم‌افزاری (سیستم عامل) و زیرساخت منابع رایانشی را بر عهده دارد. کاربر چنین مدل سرویسی، به مدیریت و اجرای کاربردهای نرم‌افزاری بر روی سیستم عامل و منابع مجازی تامین شده توسط فراهم‌کننده‌ی سرویس ابر می‌پردازد. بنابراین کاربر تقریبا کنترلی بر روی سیستم عامل و منابع سخت‌افزاری ندارد. بر خلاف سرویس مدل

---

[89] *Software as a Service (SaaS)*
[90] *Application Programming Interface (API)*
[91] *Gmail*
[92] *Google Docs*
[93] *Platform as a Service (PaaS)*

سَس که کاربردهای تکمیل شده و آماده‌ی استفاده را در اختیار کاربر قرار می‌دهد، در این مدل سرویس به کاربر فرصت داده می‌شود تا مستقیما بر روی ابر به طراحی، توسعه و آزمایش کاربرد مورد نظر خود بپردازد. همچنین این مدل سرویس امکان هم‌کاری و کار گروهی اعضای یک پروژه را فراهم می‌کند. به این ترتیب که کاربران با استفاده از این مدل سرویس رایانش ابری می‌توانند در حالی‌که در کشورهای گوناگونی هستند با هم‌کاری هم، به طور مثال به توسعه‌ی یک وب‌سایت بپردازند. به بیان کامل‌تر در این مدل سرویس، تمام تسهیلات مورد نیاز برای پیشتیبانی از چرخه‌ی ساخت و ارایه‌ی کاربردها و سرویس‌های وب از طریق اینترنت در اختیار کاربر قرار می‌گیرد.

- «زیرساخت به عنوان سرویس»[۹۴] (به طور مخفف یَس): در این مدل، فراهم‌کننده‌ی سرویس ابر، مجموعه‌ای از منابع مجازی رایانشی (مانند پهنای باند شبکه، ظرفیت ذخیره‌سازی، حافظه و توان پردازشی) را در ابر فراهم می‌نماید. بنابراین مسوولیت اجرا و نگهداری سیستم عامل و سایر کاربردهای نرم‌افزاری بر روی این منابع مجازی بر عهده‌ی کاربر سرویس است. در ارایه‌ی این مدل سرویس، برای تبدیل منابع فیزیکی به منابع منطقی (که دارای این قابلیت هستند که به طور پویایی توسط کاربران بکارگرفته شده و رهاسازی شوند) از فناوری‌های مجازی‌سازی استفاده می‌شود. به این ترتیب کاربران به جای خریداری سرورها، فضای مرکز داده و تجهیزات شبکه، این منابع را به صورت مجازی و به عنوان سرویس زیرساخت از فراهم‌کننده‌ی سرویس ابر دریافت می‌نمایند.

با توجه به اهمیت روزافزون رایانش ابری در سال‌های اخیر، به‌کارگیری این فناوری در حوزه‌ی یادگیری و آموزش الکترونیکی نیز مورد توجه پژوهش‌گران قرار گرفته و در مقاله‌ها و کارهای پژوهشی مختلفی به این موضوع پرداخته شده است که در ادامه به تعدادی از آن‌ها اشاره می‌کنیم.

در مقاله‌ی [۵۱] ویژگی‌های کلی سیستم یادگیری الکترونیکی بیان شده است و بر این اساس به توصیف طرح کلی معماری سیستمی پرداخته شده است که تلفیقی میان قابلیت‌های یادگیری الکترونیکی و سرویس‌های رایانش ابری ایجاد نماید. اما در این مقاله کارایی سیستم مبتنی بر معماری پیشنهاد شده، مورد ارزیابی قرار نگرفته است. در مقاله‌ی [۵۲] برای طراحی سیستم یادگیری الکترونیکی مبتنی بر رایانش ابری، معماری سیستم را بر پایه‌ی منابع توزیع‌شده‌ای قرار داده است که توسط کامپیوترهای شخصی کاربران تامین می‌شود. این معماری پیشنهادی، با وجود آن‌که ویژگی

---

[94] *Infrastructure as a Service (IaaS)*

کشسانی منابع، که از ویژگی‌های مشخص معماری رایانش ابری است را تامین می‌نماید، اما از قابلیت مقیاس‌پذیری بالا، که از دیگر شاخصه‌های این نوع معماری است، برخوردار نیست. به بیان دیگر، هماهنگی منابع توزیع‌یافته برای عمل‌کرد سیستم بر عهده‌ی یک نود (کامپیوتر) مرکزی است که نقطه‌ی گلوگاه سیستم محسوب می‌شود.

در مقاله‌ی [۵۳] طرح مشخصی برای پیاده‌سازی معماری یک سیستم یادگیری الکترونیکی بر اساس مدل‌های رایانش ابری ارایه نشده است و تنها به بیان مزایا و تاثیرات مثبت بکارگیری فناوری رایانش ابری در حوزه‌ی یادگیری الکترونیکی پرداخته شده است. در مقاله‌های [۵۵، ۵۴] پس از بررسی مزایای استفاده از سرویس‌های رایانش ابری در سیستم‌های یادگیری و آموزش الکترونیکی، به ارایه و طراحی معماری مفهومی این سیستم‌ها بر پایه‌ی مدل‌های رایانش ابری پرداخته شده است، اما کارایی سیستم‌های پیاده‌سازی شده بر اساس طرح معماری پیشنهادی مورد ارزیابی قرار نگرفته است.

به طور کلی مروری بر کارهای انجام گرفته نشان می‌دهد که در بیشتر آن‌ها، بحث و بررسی چالش‌ها و فرصت‌ها، و مزایا و معایب به‌کارگیری رایانش ابری در حوزه‌ی یادگیری الکترونیکی مورد تاکید قرار داده شده است. در حالی که مساله‌ی ارزیابی کارایی سیستم‌های یادگیری و آموزش الکترونیکی مبتنی بر مدل‌های رایانش ابری نیز، در کنار بررسی‌های انجام گرفته در کارهای پیشین، از اهمیت بالایی برخوردار است. ما در ادامه، پس از اشاره‌ی کوتاهی به مزایای استفاده از سرویس‌های رایانش ابری در سیستم‌های یادگیری و آموزش الکترونیکی، به ارایه و طراحی معماری این سیستم‌ها بر پایه‌ی مدل‌های رایانش ابری می‌پردازیم. سپس با پیاده‌سازی معماری طراحی شده با استفاده از ابزار شبیه-سازی کِلُدسیم، سناریوهای مختلف توسعه‌ی سیستم پیشنهادی برای یادگیری و آموزش الکترونیکی را مورد تحلیل و ارزیابی قرار می‌دهیم.

برخی از قابلیت‌ها و مزایایی که ارایه‌ی سرویس‌های یادگیری الکترونیکی در محیط‌های رایانش ابری، به همراه خواهند داشت عبارتند از [۴۷،۴۲]:

- دارا بودن قابلیت دسترسی از طریق وب نشان‌دهنده‌ی سادگی دسترسی به سرویس‌های یادگیری الکترونیکی در محیط‌های رایانش ابری است. زیرا هر کسی در هر زمان و از هر مکانی امکان دسترسی به کاربرد را خواهد داشت.
- به علت عدم نیاز به نرم‌افزار خاص در سمت کاربر، هزینه‌های درخواست‌کنندگان و مشترکین کاهش می‌یابد. زیرا نیازی به صرف هزینه‌های نصب و نگهداری نرم‌افزار و

مدیریت و بکارگیری سرور نخواهد بود. همچنین هزینه‌های پایین‌تری برای حقوق مالکیت کاربرد پرداخته می‌شود.

- قابلیت پرداخت اشتراک بر حسب استفاده، برای بازار آموزش نرم‌افزاری بسیار مناسب است و امکان دسترسی به کاربردهای پیچیده‌تر و غنی‌تر را فراهم می‌سازد.
- از آن‌جایی که کاربرد بر روی سرورهای ابر اجرا می‌شود، سیستم از مقیاس‌پذیری بالایی برخوردار خواهد بود. بنابراین با افزایش تعداد افراد یادگیرنده، کارایی نرم‌افزار کاهش نیافته و امکان پشیبانی از چندین موسسه‌ی آموزشی فراهم خواهد بود.
- در نتیجه‌ی ذخیره‌ی اطلاعات بر روی ابر، به منظور انتقال از یک قطعه و سیستم به قطعه و سیستم دیگر، نیازی به پشتیبان‌گیری موقت وجود ندارد. همچنین افراد یادگیرنده، امکان ایجاد و نگهداری پایگاهی از اطلاعات را دارند که حجم آن در صورت نیاز قابل افزایش است.
- در صورت خرابی کامپیوتر کاربر، نیازی به جبران و بازیابی وضعیت وجود ندارد، زیرا تقریبا تمام داده‌ها در ابر ذخیره شده‌اند و بنابراین از دست نمی‌روند.
- افراد یادگیرنده می‌توانند با استفاده از قطعات مختلف مانند گوشی همراه، کامپیوتر سیار و یا میزکار (در صورت دسترسی به اینترنت) از طریق ابر و کاربردهای مبتنی بر مرورگر، فایل-های خود را بیابند و آن‌ها را ویرایش کنند.
- یکی دیگر از قابلیت‌ها، انعطاف‌پذیری منابع است. مقیاس زیرساخت رایانشی را می‌توان بر حسب نیاز افزایش و یا کاهش داد و به این ترتیب سود سرمایه‌گذاری را بیشینه کرد.

از لحاظ امنیتی نیز بکارگیری مدل‌های رایانش ابری، برای افراد و موسسات آموزشی که به استفاده و توسعه‌ی سرویس‌های یادگیری الکترونیکی می‌پردازند، مزایایی به همراه خواهد داشت که برخی از آن‌ها عبارتند از:

- برای افرادی که قصد سواستفاده از سیستم را دارند، تعیین موقعیت ماشینی که داده‌ی موردنظر آن‌ها (مانند سوالات امتحانی، نتایج) را ذخیره کرده است، تقریبا غیرممکن است. چنین افرادی نمی‌توانند مولفه‌ها و اجزای فیزیکی متناظر با دارایی‌های دیجیتالی را شناسایی کنند.
- قابلیت مجازی‌سازی، امکان جایگزینی سریع سرورهای واقع در ابر را فراهم می‌سازد، بدون این‌که در نتیجه‌ی این جایگزینی هزینه و خسارت زیادی وارد شود. با ایجاد نمونه‌های

همسانی از ماشین‌های مجازی در ابر، به سادگی می‌توان مدت زمان از کار افتادگی سرویس را کاهش داد.

- به دلیل ذخیره‌ی متمرکز داده‌ها، از دست دادن یک کاربر تاثیر قابل توجهی بر روی سیستم نخواهد گذاشت و تا زمانی که بخش عمده‌ی داده و کاربردها بر روی ابر ذخیره شده باشد، امکان اتصال سریع کاربران جدید وجود خواهد داشت.
- پایش نمودن دسترسی به داده‌ها آسان‌تر خواهد بود، زیرا تنها پایش یک مکان مورد نیاز است (نه برای مثال، هزاران سیستم کامپیوتری متعلق به یک دانشگاه). همچنین، از آنجایی که برای تمام کاربران ابر، تنها یک مدخل ورودی در نظر گرفته می‌شود. تغییر سیاست‌ها و تکنیک‌های امنیتی به سادگی قابل آزمایش و پیاده‌سازی هستند.

به این ترتیب رایانش ابری با دارا بودن این قابلیت‌ها و مزایا، راه‌کار مناسب و کم‌هزینه‌ای را برای آموزش و یادگیری الکترونیکی ارایه می‌نماید که می‌تواند توسط مجموعه‌های آموزشی و تحقیقاتی که قصد ارایه و توسعه‌ی چنین سرویس‌هایی دارند (مانند پژوهشگاه علوم و فن‌آوری اطلاعات ایران) بکار گرفته شود.

سرویس‌های رایانش ابری در سطوح مختلف می‌توانند در ارایه و توسعه‌ی محیط‌های یادگیری الکترونیکی مورد استفاده قرار گیرند:

- سرویس زیرساخت: به منظور بکارگیری رایانش ابری در این سطح، می‌توان محیط یادگیری الکترونیکی را مبتنی بر سرویس‌های زیرساخت فراهم‌کنندگان ابر ارایه کرد.
- سرویس بستر: به منظور بکارگیری رایانش ابری در این سطح، می‌توان محیط یادگیری الکترونیکی را مبتنی بر «واسط‌های توسعه»[95]ی ارایه شده توسط فراهم‌کنندگان سرویس‌های بستر توسعه داد.
- سرویس نرم‌افزار: به منظور بکارگیری رایانش ابری در این سطح، می‌توان از محیط‌های یادگیری الکترونیکی ارایه شده توسط فراهم‌کنندگان سرویس‌های نرم‌افزار استفاده نمود.

در ادامه بررسی خواهیم کرد که چگونه یادگیری الکترونیکی را می‌توان به صورت یک سرویس آموزشی در ابر ارایه نمود. برای پیاده‌سازی این سرویس مبتنی بر مدل‌های رایانش ابری، کاربران نیازمندی‌های سخت‌افزاری زیادی ندارند و از این رو پیاده‌سازی از پیچیدگی بالایی برخوردار نخواهد بود.

---

[95] *Development interface*

۵-۱-۱- طراحی سیستم یادگیری الکترونیکی

همان‌طور که پیش‌تر اشاره کردیم، مدل‌های رایانش ابری پتانسیل بالایی در ارایه‌ی محیط‌های یادگیری الکترونیکی دارند، و با میزبانی کاربردهای یادگیری الکترونیکی بر روی ابر و بهره‌گیری از قابلیت‌های مجازی‌سازی سخت‌افزار، می‌توان هزینه‌های ایجاد و نگهداری منابع مورد نیاز در روند یادگیری الکترونیکی را کاهش داد. به بیان دقیق‌تر، با بکارگیری رایانش ابری می‌توان زیرساختی قابل اطمینان، انعطاف‌پذیر و مقرون به صرفه برای ارایه‌ی سرویس‌های یادگیری الکترونیکی فراهم ساخت که دارای قابلیت تنظیم خودکار و تضمین کیفیت سرویس است.

در حال حاضر ترکیب فن‌آوری‌های رایانش ابری و یادگیری الکترونیکی گسترش زیادی یافته است. برخی از کارهای انجام گرفته در این حوزه بر روی بکارگیری سرویس‌های زیرساخت ابر تاکید کرده‌اند و به این منظور، برای دانشجویان، در بازه‌های زمانی خاص به رزرو ماشین‌های مجازی پرداخته‌اند [۴۸]. به عنوان مثال دیگری در این حوزه می‌توان به کاربردی به نام بلواِسکای[۹۶] [۴۹] اشاره کرد که معماری آن به گونه‌ای طراحی شده است که توانایی مدیریت کارای سرویس‌های یادگیری الکترونیکی را داشته باشد و همچنین با پیش‌بینی و تامین منابع برای محتواهای پرطرفدار، کارایی سیستم را در دسترسی‌های هم‌زمان تضمین نماید. البته در این کاربرد و دیگر کاربردهای تجاری مشابه آن، به جزییات مربوط به چگونگی دست‌یابی به این اهداف پرداخته نشده است. همچنین می‌توان به چارچوبی به نام کلودآی‌اِی[۹۷] [۵۰] اشاره کرد که با ایجاد و پیکربندی تصاویر ماشین‌های مجازی بر حسب تقاضا، به دانش‌آموزان این امکان را می‌دهد که محیط‌های «جاوا سِرولِت» خود را (شامل مای-اِس‌کیوال، تام‌کَت و ...) برای انجام آزمایش‌هایشان در اختیار داشته باشند. با در پیش گرفتن این روی‌کرد، دانش‌آموزان می‌توانند بر روی توسعه و آزمایش کاربردهای خود تمرکز نمایند.

با در نظر گرفتن مولفه‌های سازنده یک سیستم یادگیری الکترونیکی، می‌توان قابلیت‌ها و سرویس-هایی که چنین محیطی به کاربران خود (که اغلب دانشجویان، اساتید و پژوهشگران هستند) ارایه می-کند را شامل موارد زیر دانست:

- مدیریت مواد آموزشی، که از جمله‌ی این مواد آموزشی می‌توان به ارایه‌ها[۹۸]، سندها[۹۹]، عکس‌ها، نقشه‌ها و پروژه‌های آموزشی اشاره کرد.

---

[96] *BlueSky*
[97] *CloudIA*
[98] *presentations*
[99] *documents*

- ارایه‌ی ابزارهایی برای انجام آزمایش‌های علمی و شبیه‌سازی آن‌ها
- فراهم کردن امکان تعاملات جمعی، که این تعاملات ممکن است از طریق گپ[100]، پیامک، رایانامه، انجمن[101]، کنفرانس‌های صوتی و تصویری، نظرگیری و جستجوی پروفایل کاربران انجام گیرد.
- ارایه‌ی ابزارهای ارزیابی دانشجویان، مانند برگزاری امتحان‌ها و دریافت و تصحیح تکالیف آموزشی
- دارا بودن مخزنی[102] برای نگهداری مواد آموزشی

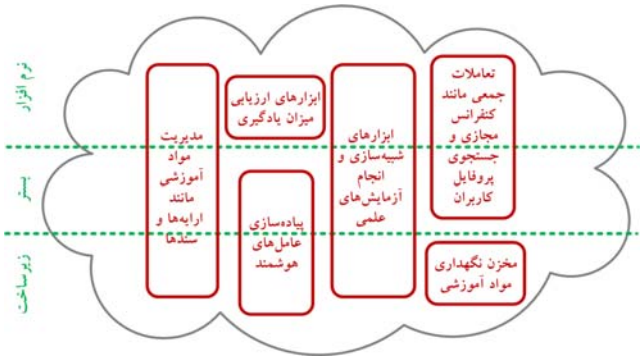
علاوه بر این قابلیت‌ها معمولا در محیط‌های یادگیری الکترونیکی «عامل‌های هوشمند»[103] نیز پیاده-سازی می‌شوند که با دنبال کردن روند یادگیری دانشجویان، بازخوردهایی را ایجاد می‌کنند که هم می‌تواند در اختیار اساتید قرار گیرد و هم می‌تواند به عنوان مبنایی برای کمک به دانشجویان برای یادگیری موثرتر آن‌ها قرار گیرد.

قابلیت‌هایی که برای محیط یادگیری الکترونیکی به آن‌ها اشاره شد را می‌توان به شیوه‌های مختلفی با بکارگیری سرویس‌های رایانش ابری (سرویس زیرساخت، بستر و نرم‌افزار) پیاده‌سازی نمود که نمونه‌ای از آن در شکل ۱۳ نشان داده شده است. در شیوه‌ی طراحی شده، برای پیاده‌سازی هر یک از قابلیت‌های اشاره شده، با توجه به نیازمندی‌های آن‌ها، بکارگیری سرویس مشخصی از سرویس‌های رایانش ابری پیشنهاد گردیده است. به عنوان یک نمونه، به منظور ارایه‌ی ابزارهای شبیه‌سازی در محیط یادگیری الکترونیکی، ممکن است بنابر نیازمندی‌های آزمایش علمی موردنظر، نیاز به بکارگیری هر سه نوع سرویس زیرساخت، بستر و نرم‌افزار باشد. برای مثال دانشجویان مهندسی کامپیوتر برای انجام آزمایش‌هایی بر روی مسیریابی در شبکه‌های کامپیوتری و یا نحوه‌ی پیکربندی فایروال در شبکه نیاز به چندین دستگاه کامپیوتری دارند. بنابراین برای انجام چنین آزمایش‌هایی ایجاد ماشین‌های مجازی (به ازای دستگاه‌های کامپیوتری مورد نیاز) با بکارگیری سرویس زیرساخت‌های رایانش ابری می‌تواند چاره‌ساز باشد. در انجام آزمایش‌های مربوط به بسیاری دیگر از دروس، دانشجویان نیاز به محیط برنامه‌نویسی خاص با پیکربندی‌های مشخصی دارند. در چنین مواردی می-توان با بکارگیری سرویس بسترهای رایانش ابری، بستر مناسب را به صورت آماده در اختیار دانشجویان قرار داد تا به سادگی کاربردهای موردنظر خود را بر روی آن توسعه و مورد آزمایش قرار

---

[100] *chat*
[101] *forum*
[102] *repository*
[103] *Smart agents*

دهند. در نهایت در برخی دیگر از دروس مانند آنالیز گراف، ممکن است تنها اجرای یک مدل شبیه-سازی شده در قالب یک اَپلت برای پیشبرد روند آموزش و یادگیری کافی باشد، که در این صورت با بکارگیری سرویس نرم‌افزارهای رایانش ابری می‌توان به این منظور دست یافت.



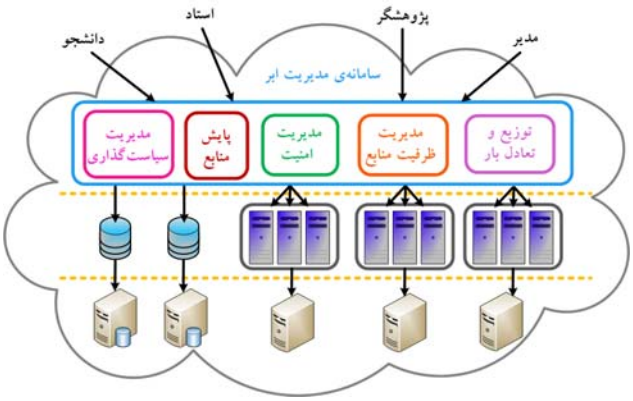
شکل ۱۳ . چگونگی سرویس‌های رایانش ابری (سرویس زیرساخت، بستر و نرم‌افزار) به منظور پیاده‌سازی قابلیت‌های متداول یک سیستم یادگیری و آموزش الکترونیکی

نحوه‌ی طراحی یک سیستم متداول یادگیری الکترونیکی در محیط‌های رایانش ابری در شکل ۱۴ نشان داده شده است. در این معماری، نخستین لایه مربوط به سامانه‌ی مدیریت ابر است که از یک سو با کاربران انتهایی و از سوی دیگر با منابع مجازی موجود در ابر در ارتباط است. این سامانه وظیفه دارد، با بررسی تقاضاهای دریافتی از سوی کاربران، نیازمندی‌های محاسباتی آن‌ها را استخراج کرده و متناسب با این نیازمندی‌ها، سرویس رایانش ابری مناسبی که در پاسخ به این تقاضاها باید ارایه گردد را تعیین نماید. برای دست‌یابی به این منظور، این سامانه خود از زیرسامانه‌های گوناگونی تشکیل یافته است که عبارتند از:

- بخش پایش و نظارت بر نحوه‌ی اجرای تقاضاهای دریافتی، نحوه‌ی پیکربندی و بکارگیری منابع و سلامت یا از کارافتادگی آن‌ها (این بخش بازخوردهای مورد نیاز برای عملکرد مناسب سایر بخش‌ها مانند بخش توزیع بار و یا مدیریت ظرفیت منابع را تامین می‌نماید.)
- بخش توزیع بار محاسباتی بر روی منابع فیزیکی به منظور ایجاد تعادل بار حاصل از ماشین-های مجازی مورد نیاز
- بخش مدیریت ظرفیت منابع محاسباتی و ذخیره‌سازی و مقیاس‌پذیری آن‌ها (افزایش و یا کاهش میزان منابع) در صورت نیاز

- بخش مدیریت امنیت به منظور نظارت بر ورود کاربران به محیط، نحوه‌ی دسترسی و سایر فعالیت‌های آن‌ها، و اطمینان از حفظ محرمانگی و یکپارچگی اطلاعات و داده‌ها و امنیت تراکنش‌های کاربران
- بخش مدیریت سیاست‌گذاری به منظور پایه‌گذاری و حفظ سیاست‌های آموزش و یادگیری موردنظر، و هم‌چنین سیاست‌های زمان‌بندی و نحوه‌ی تخصیص منابع با توجه به اولویت نیازمندی‌های محیط یادگیری الکترونیکی

لایه‌ی دوم در معماری طراحی شده، دربرگیرنده‌ی فضای ذخیره‌سازی و ماشین‌های مجازی پیاده-سازی شده، به منظور ایجاد محیط یادگیری الکترونیکی بر مبنای رایانش ابری است. در نهایت پایین-ترین لایه‌ی معماری، از منابع فیزیکی و سخت‌افزارهایی تشکیل یافته است که منابع مجازی مورد نیاز در لایه‌ی بالاتر برای راه‌اندازی محیط یادگیری الکترونیکی، در عمل بر روی آن‌ها اجرا و پیاده‌سازی می‌شوند.



شکل ۱۴. نحوه‌ی طراحی یک سیستم یادگیری الکترونیکی در محیط رایانش ابری

سامانه‌ی مدیریت ابر نقش مهمی بر عهده دارد و سطوح مختلف «زیرساختی»، «محتوایی» و «کاربردی» محیط یادگیری الکترونیکی را مدیریت می‌کند. در سطح زیرساخت، این سامانه با بکارگیری فناوری‌های مجازی‌سازی سخت‌افزار و نرم‌افزار، از پایداری و در دسترس بودن منابع زیرساختی محیط یادگیری الکترونیکی اطمینان حاصل می‌کند و ظرفیت محاسباتی و ذخیره‌سازی مورد نیاز در سطوح دیگر (محتوایی و کاربردی) را تامین می‌نماید. مدیریت سطح محتوایی محیط یادگیری الکترونیکی، به طور عمده دربرگیرنده‌ی مدیریت پایگاه‌های داده و فایل‌ها و سرویس‌های وب مورد نیاز برای ارایه‌ی محتوا است. در این سطح، علاوه بر حفظ و نگهداری محتوا، باید به مسایل

مربوط به تامین واسط‌های استاندارد برنامه‌نویسی کاربردها برای استفاده از محتوا نیز پرداخته شود. در نهایت در سطح کاربرد، این سامانه به مدیریت ابزارها و سرویس‌های کاربردی محیط یادگیری الکترونیکی می‌پردازد و واسط‌های تعامل با کاربران و دیگر برنامه‌های کاربردی را تامین می‌نماید. به این ترتیب بر اساس طرح معماری پیشنهادی، متناسب با نیازمندی‌های هر یک از کاربران در محیط یادگیری الکترونیکی، سرویس‌های نرم‌افزار، بستر و یا زیرساخت رایانش ابری در اختیار آن‌ها قرار خواهد گرفت.

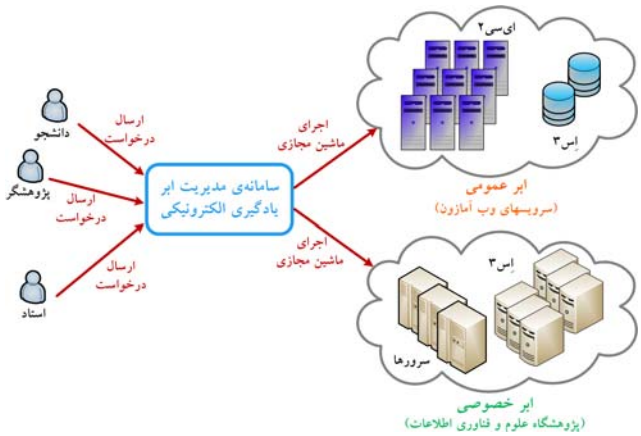
معماری طراحی شده برای یک سیستم یادگیری و آموزش الکترونیکی (**شکل ۱۴**) محدود به سرویس‌های قابل ارایه‌ی یک «ابر خصوصی»[۱۰۴] نیست. به بیان دیگر، در این معماری الزامی نیست که تمام منابع رایانشی و سرویس‌های مختلف مبتنی بر آن‌ها (در سطوح زیرساخت، بستر و نرم‌افزار) تحت مالکیت و مدیریت سازمان ارایه‌دهنده‌ی سرویس یادگیری و آموزش الکترونیکی قرار داشته باشند. هر موسسه‌ی آموزشی و یا پژوهشی که بر اساس این معماری به ارایه‌ی سرویس بپردازد، می‌تواند علاوه بر منابع اختصاصی خود، از سرویس‌های رایانش ابری که فراهم‌کنندگان «ابر عمومی»[۱۰۵] در اختیار کاربران عمومی قرار می‌دهند نیز استفاده نماید. برای نمونه می‌توان به آمازون به عنوان یکی از فراهم‌کنندگان مطرح سرویس‌های ابر عمومی اشاره کرد. «ابر محاسباتی کشسان»[۱۰۶] و «سرویس ذخیره‌سازی ساده»[۱۰۷] از جمله «سرویس‌های وب آمازون»[۱۰۸] هستند که قابلیت بکارگیری در ایجاد محیط یادگیری الکترونیکی را دارند. در شکل ۱۵ چگونگی پیاده‌سازی معماری پیشنهادی برای سیستم یادگیری و آموزش الکترونیکی بر اساس مدل ترکیبی سرویس‌های وب آمازون و منابع اختصاصی موسسه‌ی آموزشی و یا پژوهشی ارایه‌دهنده‌ی سرویس یادگیری و آموزش الکترونیکی (مانند پژوهشگاه علوم و فناوری اطلاعات ایران) نمایش داده شده است. سرویس ذخیره‌سازی ساده و ابر محاسباتی کشسان (که به طور مخفف اِس۳ و ای‌سی۲ نامیده می‌شوند) به کاربران این امکان را می‌دهند که ظرفیت ذخیره‌سازی و محاسباتی سیستم یادگیری الکترونیکی را مطابق نیاز خود، به سادگی افزایش یا کاهش دهند و تنها به میزان منابع مصرفی خود هزینه بپردازند.

به این ترتیب در هر زمان که منابع خصوصی موسسه‌ی آموزشی یا پژوهشی موردنظر برای راه‌اندازی و نگهداری سیستم یادگیری و آموزش الکترونیکی کافی نباشد، می‌توان ماشین‌های مجازی مورد نیاز

---

[104] *private cloud*
[105] *public cloud*
[106] *Elastic Compute Cloud (EC2)*
[107] *Simple Storage Service (S3)*
[108] *Amazon Web Services (AWS)*

سیستم را بر روی ابر عمومی به اجرا درآورد. بنابراین سرویس یادگیری و آموزش الکترونیکی در یک «ابر هایبرید»[۱۰۹] ارایه می‌شود که با ترکیب سرویس‌های ابر خصوصی و عمومی در سطوح مختلف زیرساخت، بستر و نرم‌افزار، از مزایای هر دو مدل برخوردار خواهد بود. به این ترتیب که ضمن ساده‌تر شدن امکان مقیاس‌پذیری سیستم، امکان بکارگیری کارا و بر حسب نیاز منابع فراهم می‌شود و هزینه‌های مدیریت منابع، راه‌اندازی و نگهداری سیستم ارایه‌دهنده‌ی سرویس یادگیری و آموزش الکترونیکی کاهش می‌یابد. در ادامه به تحلیل و ارزیابی کارایی سیستم پیاده‌سازی شده بر اساس معماری پیشنهادی خواهیم پرداخت.



شکل ۱۵. چگونگی پیاده‌سازی معماری پیشنهادی برای سیستم یادگیری و آموزش الکترونیکی بر اساس مدل ترکیبی

۵-۱-۲- ارزیابی راه‌کار پیشنهادی

همان‌طور که پیش‌تر، در طراحی راه‌کار پیشنهادی بحث کردیم، سیستم موردنظر از یک‌سو با کاربران و از سوی دیگر با منابع رایانش ابری در ارتباط است. از این رو در ادامه‌ی این بخش، در بیان هر سناریوی آزمایش، علاوه بر سرویس‌های مورد درخواست کاربران، ظرفیت منابع رایانشی سیستم را نیز مشخص می‌کنیم. لازم به ذکر است که منابع رایانشی از منابع محاسباتی (یا به طور مخفف سی‌پی-

[109] *hybrid cloud*

یو[۱۱۰])، منابع شبکه‌ای (یا پهنای باند[۱۱۱])، حافظه[۱۱۲] و منابع ذخیره‌سازی (یا فضای ذخیره‌سازی[۱۱۳]) تشکیل می‌شوند [۵۱].

در تعریف معیاری که برای ارزیابی کارایی سیستم در هر یک از سناریوها در نظر گرفته شده است، لازم است تا هم کیفیت سرویسی که کاربران از سیستم دریافت می‌کنند و هم میزان بهره‌وری منابع رایانشی سیستم لحاظ شود. به این منظور، معیار کارایی سیستم به صورت زیر تعریف می‌شود:

$$\text{کارایی سیستم} = \alpha \times \frac{\text{میزان تقاضاهای سرویس داده شده}}{\text{میزان تقاضاهای دریافت شده}} \qquad (۱)$$

که در آن نسبت میزان تقاضاهای سرویس داده شده به میزان تقاضاهای دریافت شده، بیانگر کیفیت سرویسی است که کاربران از سیستم دریافت می‌کنند. آلفا ضریب بهره‌وری منابع رایانشی سیستم است که تاثیر این فاکتور را در کارایی سیستم لحاظ می‌کند و مطابق (۲) تعریف می‌شود:

$$\alpha = \text{(ضریب بهره‌وری منابع)} \frac{\text{میزان منابع مصرف شده}}{\text{میزان منابع اختصاص یافته}} \qquad (۲)$$

که در آن، نسبت میزان منابعی که برای سرویس‌دهی سیستم یادگیری الکترونیکی به طور متوسط مصرف می‌شود به میزان منابعی که برای پیاده‌سازی سیستم اختصاص یافته است (برای سرویس‌دهی سیستم در حالت پیک بار کاری)، به عنوان ضریب بهره‌وری منابع تعریف شده است. در حالت ایده‌آل، و به بیان دقیق‌تر در صورت بهره‌گیری کامل از قابلیت کشسانی منابع ابر، $\alpha = ۱$ خواهد شد و در غیر این صورت ($\alpha < ۱$)، متناسب با کاهش ضریب بهره‌وری منابع از کارایی سیستم کاسته خواهد شد.

با استناد بر معیار صورت‌بندی شده در (۱)، تعریف جامعی از کارایی سیستم ارایه می‌شود که در آن کارایی سیستم هم در ارتباط با کاربران و هم در ارتباط با منابع رایانشی قابل ارزیابی خواهد بود. از

---

[110] *CPU (Central processing unit)*
[111] *BW (Bandwidth)*
[112] *RAM*
[113] *Storage*

این‌رو در سناریوهایی که در ادامه‌ی این بخش بررسی می‌شوند، این معیار به عنوان یکی از شاخص-های مهم، اندازه‌گیری شده و مورد تحلیل قرار خواهد گرفت. سناریوهایی که در ادامه‌ی این بخش مورد ارزیابی قرار گرفته‌اند به گونه‌ای انتخاب شده‌اند که انواع صورت‌های متداول ارایه‌ی یک سیستم آموزش و یادگیری الکترونیکی بر روی بسترهای رایانشی بزرگ‌مقیاس را پوشش دهند. به بیان دقیق‌تر در این سناریوها چگونگی پیاده‌سازی سیستم پیشنهادی را به ترتیب بر فراز بسترهای رایانشی خصوصی، عمومی و هایبرید (که صورت‌های متداول بسترهای رایانش ابری به شمار می-روند) مورد ارزیابی قرار می‌دهیم.

ارزیابی سیستم پیاده‌سازی شده مبتنی بر مدل ابر خصوصی

در سناریوی اول حالتی را در نظر می‌گیریم که راه‌کار پیشنهادی برای ارایه‌ی سیستم آموزش و یادگیری الکترونیکی با استفاده از منابع موجود در پایگاه داده محلی و یا به بیان دیگر در یک ابر خصوصی پیاده‌سازی شده است. شرایطی که برای آزمایش سیستم در این سناریو در نظر گرفته شده به این ترتیب است که، کاربران سیستم یادگیری و آموزش الکترونیکی از تعداد N دانشجو و پژوهشگر تشکیل یافته است و فرض می‌کنیم که این تعداد از ۱۰ کاربر آغاز شده و تا ۱۰۰۰ کاربر افزایش می‌یابد. از این تعداد کاربر ۶۰ در صد، در حال استفاده از سرویس‌های مربوط به انجام شبیه-سازی و آزمایش‌های علمی در سیستم و ۴۰ درصد در حال استفاده از سرویس‌های مربوط به تعاملات جمعی در سیستم هستند[۱۱۴]. باید توجه داشت که هر یک از این نوع سرویس‌ها نیازمندی‌های پردازشی متفاوتی دارند و متناسب با آن، بار کاری متفاوتی را بر سیستم اعمال می‌کنند. این در حالی است که هر دو نوع سرویس به طور مشترک از نیازمندی «تاخیر قابل پیش‌بینی» برخوردارند. از این‌رو میزان تاخیری که کاربران در دریافت سرویس موردنظر خود از سیستم تجربه می‌کنند از اهمیت بالایی برخوردار است و به عنوان یک شاخص ارزیابی اندازه‌گیری خواهد شد.

مشخصات منابع رایانشی که برای این سناریو در نظر گرفته شده به این ترتیب است که سرورهای واقع در ابر خصوصی دارای سی‌پی‌یو با توان ۱ میلیون دستور بر ثانیه، پهنای باند ۱ گیگابایت بر ثانیه، ۱

---

[۱۱۴] لازم به ذکر است که این مقادیر به عنوان مثال انتخاب شده‌اند تا به این ترتیب شرایط ارزیابی انجام گرفته مشخص شود. از این رو این مقادیر می‌توانند بسته به شرایط ارزیابی بنا بر نیاز تغییر داده شوند.

گیگابایت حافظه و ۱ ترابایت فضای ذخیره‌سازی هستند. در مجموع ۲ سرور در پایگاه داده‌ی محلی موجود است که سیستم بار کاری را به صورت متعادل بین آن‌ها توزیع می‌نماید.
حال سیستم یادگیری الکترونیکی شبیه‌سازی شده[۱۱۵] با مشخصات فوق را راه‌اندازی نموده و به تدریج تعداد کاربران آن را افزایش می‌دهیم و در شرایط مختلف بار کاری، به ارزیابی راه‌کار پیشنهادی می‌پردازیم. در جدول ۱، نتایج حاصل از اندازه‌گیری معیار کارایی سیستم در شرایط مختلف گزارش شده است.

جدول ۱. ارزیابی کارایی سیستم پیاده‌سازی شده مبتنی بر ابر خصوصی.

تعداد کاربران	۱۰ کاربر	۲۰۰ کاربر	۵۰۰ کاربر	۱۰۰۰ کاربر
کیفیت سرویس	۱۰۰٪	۱۰۰٪	۹۱/۶۶٪	۵۶٪
بهره‌وری منابع	۲۱/۴۳٪	۶۲/۵٪	۱۰۰٪	۱۰۰٪
کارایی سیستم	۲۱/۴۳٪	۶۲/۵٪	۹۱/۶۶٪	۵۶٪

همان‌طور که مشاهده می‌شود به تدریج با افزایش بار کاری، میزان بهره‌وری منابعی که به سیستم اختصاص داده شده است افزایش می‌یابد و در نتیجه کارایی سیستم نیز بهبود می‌یابد. این روند تا زمانی ادامه می‌یابد که تمام منابع سیستم به طور کامل زیر بار رود، و یا به بیان دیگر بهره‌وری منابع ۱۰۰٪ شود. پس از آن در صورت افزایش بیشتر تعداد کاربران، دیگر سیستم توانایی ارایه‌ی سرویس به تمام تقاضاهای دریافت‌شده از سوی کاربران را نخواهد داشت و از این‌رو به تدریج با کاهش کیفیت سرویس از کارایی سیستم کاسته خواهد شد.
با توجه به اهمیت میزان تاخیر در سرویس‌دهی سیستم، نتایج حاصل از اندازه‌گیری این شاخص نیز در شرایط مختلف بار کاری در جدول ۲ آورده شده است. در این جدول، منظور از سرویس‌های نوع ۱ و ۲ به ترتیب، سرویس‌های مربوط به انجام شبیه‌سازی و آزمایش‌های علمی و سرویس‌های مربوط به تعاملات جمعی در سیستم است.

جدول ۲. میزان تاخیر در سرویس‌های مختلف سیستم پیاده‌سازی شده مبتنی بر ابر خصوصی.

تعداد کاربران		۱۰ کاربر	۲۰۰ کاربر	۵۰۰ کاربر	۱۰۰۰ کاربر
میانگین تاخیر	سرویس نوع ۱	۲/۵ ثانیه	۳۱/۹ ثانیه	۶۶/۶۵ ثانیه	۱۳۳/۴۳ ثانیه
	سرویس نوع ۲	۱/۷۵ ثانیه	۲۰/۱ ثانیه	۴۱/۷۴ ثانیه	۸۳/۴۲ ثانیه

[۱۱۵] یادآوری می‌شود که در تمام سناریوهای این بخش، محیط شبیه‌سازی کِلُدسیم مورد استفاده قرار گرفته است.

میانگین تاخیری که کاربران هر نوع سرویس دریافت می‌کنند، متناسب با میانگین بار پردازشی که آن نوع سرویس بر سیستم اعمال می‌کند متفاوت است. با توجه به این‌که منابع موجود در پایگاه داده‌ی محلی ظرفیت ثابتی دارند که باید برای ارایه‌ی سرویس به تقاضاهای کاربران به اشتراک گذاشته شوند، به تدریج با افزایش تعداد کاربران سیستم، میانگین مدت زمان تاخیر در سرویس‌دهی به آن‌ها افزایش می‌یابد. همان‌طور که مشاهده می‌شود، در شرایطی که بارکاری سیستم از ظرفیت منابعی که برای سرویس‌دهی به سیستم اختصاص داده شده است بیشتر می‌شود، میانگین تاخیر در سرویس‌دهی به کاربران به حد زیادی افزایش می‌یابد. لازم به ذکر است که در این سناریو، با توجه به این‌که دسترسی به منابع ابر خصوصی از طریق شبکه‌ی داخلی پایگاه داده‌ی محلی صورت می‌گیرد، تاخیر شبکه تقریبا برابر صفر در نظر گرفته شده است.

تحلیل کلی نتایج به‌دست آمده در این بخش به این شرح است که: در این سناریو برای راه‌اندازی سیستم یادگیری الکترونیکی تنها به استفاده از منابع موجود در پایگاه داده‌ی محلی پرداخته شده است که دارای ظرفیت ثابت و مشخصی هستند. از این‌رو با وجود این‌که با به‌کارگیری فنون مجازی‌سازی به پایش و توزیع متعادل بار در سیستم پرداخته شده است، اما به دلیل ظرفیت ثابت منابع سیستم و عدم مقیاس‌پذیری بالای آن، کارایی سیستم به شدت به بار کاری حاصل از سرویس‌های درخواستی کاربران وابسته خواهد بود. به بیان دقیق‌تر به دلیل عدم پشتیبانی کامل از کشسانی منابع در این سناریو، نمی‌توان مقیاس‌پذیری بالا و کارایی قابل پیش‌بینی از سیستم انتظار داشت.

ارزیابی سیستم پیاده‌سازی شده مبتنی بر مدل ابر عمومی

در سناریوی دوم حالتی را در نظر می‌گیریم که سیستم یادگیری الکترونیکی با استفاده از منابع رایانش ابری که توسط فراهم‌کنندگان عمومی ارایه می‌شود پیاده‌سازی شده است. این سناریو به خصوص در شرایطی رخ می‌دهد که سازمان یا موسسه‌ی ارایه‌دهنده‌ی سیستم یادگیری الکترونیکی، پایگاه داده‌ای را به طور محلی در مالکیت خود ندارد. در چنین شرایطی این سازمان یا موسسه می‌تواند با به‌کارگیری راه‌کار پیشنهادی به پیاده‌سازی سیستم مبتنی بر مدل ابر عمومی بپردازد و تنها به میزان منابعی که از ابر عمومی مصرف می‌شود هزینه پرداخت نماید. باید توجه داشت که در این سناریو منابع مورد نیاز برای پیاده‌سازی سیستم از طریق شبکه‌ی اینترنت قابل دسترسی خواهد بود (بر خلاف سناریوی قبل که این منابع در پایگاه داده‌ی محلی واقع بود). از این‌رو در این سناریو، تاخیر شبکه قابل

صرف‌نظر نبوده و باید در میزان تاخیر در سرویس‌دهی سیستم لحاظ شود. از سوی دیگر در این سناریو، محدودیتی بر روی ظرفیت منابع سیستم وجود ندارد و به دلیل استفاده از مدل ابر عمومی می‌توان میزان منابع سیستم را نامحدود در نظر گرفت (بر خلاف سناریوی قبل که ظرفیت منابع موجود در پایگاه داده‌ی محلی ثابت بود).

برای فراهم کردن امکان مقایسه‌ی دقیق‌تر، تعداد کاربران سیستم، وضعیت افزایش آن‌ها و همچنین مشخصات سرویس‌هایی که مورد تقاضای آن‌هاست را مشابه سناریوی قبل در نظر می‌گیریم. نتایج بدست آمده از اندازه‌گیری هزینه‌ی سرویس‌دهی و کارایی راه‌کار پیشنهادی مبتنی بر مدل ابر عمومی در شرایطی که تعداد کاربران سیستم از ۱۰ کاربر به ۱۰۰۰ کاربر افزایش می‌یابد در جدول ۳ گزارش شده است.

جدول ۳. ارزیابی کارایی و هزینه‌ی سیستم پیاده‌سازی شده مبتنی بر ابر عمومی.

تعداد کاربران	۱۰ کاربر	۲۰۰ کاربر	۵۰۰ کاربر	۱۰۰۰ کاربر
کارایی سیستم	٪۱۰۰	٪۱۰۰	٪۱۰۰	٪۱۰۰
هزینه سرویس‌دهی	۱۸۹ دلار	۲۷۱ دلار	۳۲۵ دلار	۵۴۲ دلار

از آن‌جایی که در این سناریو، پیاده‌سازی سیستم مبتنی بر مدل ابر عمومی انجام گرفته است، میزان منابع سیستم را می‌توان نامحدود در نظر گرفت. از این‌رو صرف‌نظر از تعداد کاربران، سیستم همواره می‌تواند به تمام تقاضاهای دریافت‌شده، سرویس ارایه نماید (به بیان دقیق‌تر کیفیت سرویس ٪۱۰۰ است). همچنین به دلیل پشتیبانی کامل از کشسانی منابع، میزان منابعی که در ابر عمومی به سیستم اختصاص داده می‌شود، همواره متناسب با میزان منابع مصرفی سیستم است (به بیان دقیق‌تر بهره‌وری منابع ٪۱۰۰ است). در نتیجه، سیستم در شرایط مختلف بار کاری همواره به کارایی ٪۱۰۰ دست می‌یابد. البته باید توجه داشت که در این سناریو، متناسب با میزان مصرف منابع ابر عمومی باید هزینه پرداخت شود. هزینه‌هایی که در جدول ۳ آورده شده است، بر اساس فرمول (۳) محاسبه شده است:

$$\begin{aligned}\text{هزینه‌ی کل} = {} & (\text{هزینه مصرف هر واحد}) \times (\text{میزان منابع ذخیره‌سازی}) + (\text{هزینه مصرف هر واحد}) \times (\text{میزان منابع پردازشی}) \\ & (\text{هزینه مصرف هر واحد}) \times (\text{میزان منابع شبکه}) + (\text{هزینه مصرف هر واحد}) \times (\text{مقدار حافظه}) + \end{aligned} \quad (۳)$$

که در این محاسبات هزینه مصرف هر واحد منابع محاسباتی (سی‌پی‌یو)، منابع شبکه‌ای (پهنای باند)، حافظه و منابع ذخیره‌سازی (فضای ذخیره‌سازی) به ترتیب برابر ۳ دلار، ۰/۱ دلار، ۰/۰۵ دلار و ۰/۱

دلار در نظر گرفته شده است[۱۱۶]. در نهایت نتایج به‌دست آمده برای میانگین تاخیر اندازه‌گیری شده در سرویس‌دهی به کاربران در جدول ۴ آورده شده است.

جدول ٤. میزان تاخیر در سرویس‌های مختلف سیستم پیاده‌سازی شده مبتنی بر ابر عمومی.

تعداد کاربران		۱۰ کاربر	۲۰۰ کاربر	۵۰۰ کاربر	۱۰۰۰ کاربر
میانگین تاخیر	سرویس نوع ۱	۷/۵ ثانیه	۷/۵ ثانیه	۷/۵ ثانیه	۷/۵ ثانیه
	سرویس نوع ۲	۶/۷۵ ثانیه	۶/۷۵ ثانیه	۶/۷۵ ثانیه	۶/۷۵ ثانیه

همان‌طور که مشاهده می‌شود، مشابه با سناریوی قبل، هر یک از انواع سرویس‌ها متناسب با حجم بار کاری که بر سیستم اعمال می‌کنند، منجر به میانگین تاخیر متفاوتی در سرویس‌دهی سیستم می‌شوند. با این تفاوت که در این سناریو، همان‌طور که پیشتر بحث شد، با افزایش تعداد کاربران میانگین تاخیر در سرویس‌دهی سیستم تغییری نمی‌کند. لازم به ذکر است که در تمام نتایج گزارش شده در این جدول تاخیر حاصل از به‌کارگیری منابع سیستم از طریق شبکه‌ی اینترنت به طور متوسط برابر ۵ ثانیه در نظر گرفته شده است. با توجه به متغیر بودن کیفیت دسترسی از طریق اینترنت در نقاط مختلف، می‌توان مقدار تاخیر لحاظ شده به این منظور را متناسب با وضعیت تغییر داد.

تحلیل کلی نتایج به‌دست آمده در این بخش به این شرح است که: در این سناریو برای راه‌اندازی سیستم یادگیری الکترونیکی به طور کامل به استفاده از منابع تامین شده توسط فراهم‌کنندگان عمومی رایانش ابری پرداخته شده است. از این‌رو راه‌کار پیشنهادی با پشتیبانی کامل از قابلیت کشسانی منابع و همچنین با در اختیار داشتن منابع متناسب با میزان تقاضا، کارایی قابل پیش‌بینی و بالایی را از خود نشان می‌دهد (با فرض دسترسی به شبکه‌ی اینترنت). همچنین سیستم از مقیاس‌پذیری بسیار بالایی برخوردار است، و تنها با پرداخت هزینه‌ی نسبی پایینی متناسب با بار کاری، می‌تواند با افزایش تعداد کاربران هم‌چنان با کیفیت مطلوب به سرویس‌دهی ادامه دهد.

ارزیابی سیستم پیاده‌سازی شده مبتنی بر مدل ابر هایبرید

[۱۱۶] طبق قیمت‌گذاری‌های متداول در مدل‌های کسب و کار فراهم‌کنندگان مطرح ابر عمومی

در سناریوی سوم حالتی را در نظر می‌گیریم که سیستم یادگیری الکترونیکی بر اساس مدل ابر هایبرید و با استفاده از ترکیبی از منابع موجود در پایگاه داده‌ی محلی و منابع تامین شده توسط فراهم‌کنندگان عمومی رایانش ابری پیاده‌سازی شده است. رخ دادن چنین سناریویی در عمل بسیار متداول است. برای مثال در صورتی که سازمان یا موسسه‌ی ارایه‌دهنده‌ی سیستم یادگیری الکترونیکی، منابع بسیار محدودی به طور محلی در اختیار داشته باشد (در این شرایط در پیش گرفتن این سناریو بر سناریوی اول ارجحیت دارد) و یا در صورتی‌که بنا بر موقعیت محلی، دسترسی آن به شبکه‌ی اینترنت غیرقابل پیش‌بینی باشد (در این شرایط در پیش گرفتن این سناریو بر سناریوی دوم ارجحیت دارد)، می‌تواند با به‌کارگیری راه‌کار پیشنهادی به پیاده‌سازی سیستم مبتنی بر مدل ابر هایبرید بپردازد.

در این سناریو نیز مشابه دو سناریوی قبل، فرض می‌کنیم که کاربران سیستم یادگیری الکترونیکی در حال استفاده از سرویس‌های مربوط به انجام شبیه‌سازی و آزمایش‌های علمی (٪۶۰ درصد کاربران) و تعاملات جمعی (٪۴۰ درصد کاربران) در سیستم هستند. مشخصات منابع رایانشی که برای این سناریو در نظر گرفته شده به این ترتیب است که منابع موجود در پایگاه داده‌ی محلی به نسبت محدود است و تنها از یک سرور تشکیل یافته است که این سرور دارای سی‌پی‌یو با توان ۱ میلیون دستور بر ثانیه، پهنای باند ۱ گیگابایت بر ثانیه، ۱ گیگابایت حافظه و ۱ ترابایت فضای ذخیره‌سازی است. مشابه سناریوهای قبل، تعداد کاربرانی که در حال استفاده از سرویس‌های سیستم هستند را از ۱۰ کاربر به ۱۰۰۰ کاربر افزایش می‌دهیم تا کارایی راه‌کار پیشنهادی را در شرایط مختلف بار کاری مورد ارزیابی قرار دهیم. نتایج این ارزیابی در جدول ۵ گزارش شده است.

جدول ۵. ارزیابی کارایی و هزینه‌ی سیستم پیاده‌سازی شده مبتنی بر ابر هایبرید.

تعداد کاربران	۱۰ کاربر	۲۰۰ کاربر	۵۰۰ کاربر	۱۰۰۰ کاربر
کیفیت سرویس	٪۱۰۰	٪۱۰۰	٪۱۰۰	٪۱۰۰
بهره‌وری منابع	٪۴۲/۸۶	٪۱۰۰	٪۱۰۰	٪۱۰۰
کارایی سیستم	٪۴۲/۸۶	٪۱۰۰	٪۱۰۰	٪۱۰۰
بارگذاری روی ابر	٪۰	٪۲۰	٪۳۳	٪۶۰
هزینه سرویس‌دهی	۰۰/۰	۵۴ دلار	۱۰۸ دلار	۳۲۵ دلار

در این سناریو، متناسب با حجم پردازشی تقاضاهای کاربران سیستم و همچنین سیاست‌های مدیریت منابع تعریف شده در سیستم، بخشی از بار کاری با استفاده از منابع ابر عمومی سرویس‌دهی می‌شود. نتایج آورده شده در این جدول، علاوه بر این‌که سهم این بخش از منابع سیستم (واقع در ابر عمومی)

را در سرویس‌دهی به کل بار کاری سیستم مشخص می‌کند، هزینه‌ی محاسبه شده برای استفاده از این منابع را نیز نشان می‌دهد.

همان‌طور که در جدول ۵ نشان داده شده است، سیستم یادگیری الکترونیکی مبتنی بر مدل ابر هایبرید، همواره می‌تواند به تمام تقاضاهای دریافت شده از سوی کاربران سرویس‌دهی نماید (کیفیت سرویس ٪۱۰۰). در شرایط بار کاری پایین (نتایج گزارش شده به ازای ۱۰ کاربر در این جدول)، سیستم تنها با استفاده از منابع موجود در پایگاه داده‌ی محلی به سرویس‌دهی کاربران می‌پردازد. اما به تدریج با افزایش تعداد کاربران، بار کاری سیستم از ظرفیت منابع محلی آن بیشتر می‌شود و از این‌رو، بخشی از بار سیستم بر روی منابعی که بر حسب نیاز در ابر عمومی به آن اختصاص داده می شود، بارگذاری می‌شود. در این سناریو سیاست مدیریت منابع در سیستم به این ترتیب تعریف شده است که تنها پس از زیر بار رفتن کامل منابع محلی از منابع واقع در ابر سیستم استفاده شود. بدیهی است که در این شرایط بهره‌وری منابع سیستم ٪۱۰۰ است. بنابراین با شروع به استفاده از منابع ابر، سیستم به کارایی ٪۱۰۰ دست می‌یابد.

محاسبه‌ی هزینه‌هایی که در جدول ۵ آورده شده، بر طبق فرمول (۳) در بخش قبل انجام گرفته است. در نتایج گزارش شده به ازای ۱۰ کاربر، هزینه‌ی سرویس‌دهی برابر صفر است. زیرا در این شرایط، حجم بار کاری سیستم پایین است، و در نتیجه سیستم بدون استفاده از منابع واقع در ابر عمومی به سرویس‌دهی می‌پردازد. به طور کلی، همان‌طور که انتظار می‌رود در مقایسه با سناریوی قبل، هزینه‌ی سرویس‌دهی به کاربران سیستم کاهش یافته است که علت آن استفاده از منابع محلی در ترکیب با منابع ابر عمومی در پیاده‌سازی سیستم است.

در نهایت نتایج اندازه‌گیری شده برای میانگین تاخیری که کاربران سرویس‌های مختلف در شرایط مختلف بار کاری سیستم تجربه می‌کنند، در جدول ۶ آورده شده است. مشابه سناریوهای قبل، در این جدول نیز، منظور از سرویس‌های نوع ۱ و ۲ به ترتیب، سرویس‌های مربوط به انجام شبیه‌سازی و آزمایش‌های علمی و سرویس‌های مربوط به تعاملات جمعی است.

جدول ٦. میزان تاخیر در سرویس‌های مختلف سیستم پیاده‌سازی شده مبتنی بر ابر هایبرید.

تعداد کاربران		۱۰ کاربر	۲۰۰ کاربر	۵۰۰ کاربر	۱۰۰۰ کاربر
میانگین تاخیر	سرویس نوع ۱	۲/۵ ثانیه	۱۳/۵۱ ثانیه	۲۳/۷۶ ثانیه	۲۸/۹۳ ثانیه
	سرویس نوع ۲	۱/۷۵ ثانیه	۸/۷۱ ثانیه	۱۶/۷۷ ثانیه	۱۸/۷ ثانیه

همان‌طور که مشاهده می‌شود در شرایطی که ۱۰ کاربر در حال استفاده از سیستم هستند، در نتیجه‌ی سرویس‌دهی کامل از منابع محلی، میانگین تاخیر به نسبت پایین‌تر است. این در حالی است که با شروع به بارگذاری سیستم بر روی منابع ابر عمومی، تاخیر حاصل از شبکه‌ی خارجی نیز به تاخیر سرویس‌دهی افزوده می‌شود. باید دقت شود که این تاخیر تنها در سرویس‌دهی به بخشی از بار کاری که از طریق منابع سیستم در ابر عمومی سرویس داده می‌شوند تاثیرگذار است. بنابراین متناسب با بارگذاری این بخش از منابع سیستم، در نتیجه‌ی لحاظ تاخیر شبکه، میانگین تاخیر در سرویس‌دهی افزایش می‌یابد.

در این سناریو نیز مشابه سناریوی قبل تاخیر دسترسی از طریق شبکه‌ی خارجی به طور میانگین ۵ ثانیه در نظر گرفته شده است که این مقدار متناسب با وضعیت دسترسی از طریق شبکه‌ی اینترنت در نقاط مختلف، قابل تغییر است. به طور کلی نتایج گزارش‌شده در جدول ۶ نشان می‌دهد که در این سناریو، با افزایش تعداد کاربران سیستم، میانگین تاخیر در سرویس‌دهی به آن‌ها به کندی افزایش می‌یابد[۱۱۷] و همواره می‌توان آن را در حد قابل قبولی تنظیم نمود.

تحلیل کلی نتایج به‌دست آمده در این بخش به این شرح است که: در این سناریو برای پیاده‌سازی سیستم یادگیری الکترونیکی هم از منابع موجود در پایگاه داده‌ی محلی و هم از منابع تامین شده توسط فراهم‌کنندگان عمومی ابر استفاده شده است. با توجه به این‌که راه‌اندازی سیستم بر اساس راه-کار پیشنهادی، وابستگی کامل به دسترسی شبکه‌ی اینترنت ندارد و با استفاده از منابع محلی نیز می-توان به کاربران (به تعداد محدود) سرویس‌دهی کرد، عمل‌کرد سیستم در مقایسه با سناریوی دوم از اطمینان‌پذیری بالاتری برخوردار است. از سوی دیگر بر طبق سیاست‌های تعریف‌شده در بخش مدیریت منابع سیستم یادگیری الکترونیکی، می‌توان با به‌کارگیری منابع ابر عمومی و ترکیب آن با منابع محلی سیستم، تعداد کاربرانی که از سرویس‌های ارایه‌شده توسط سیستم استفاده می‌کنند را بر حسب نیاز افزایش داد و در نتیجه سیستم با پشتیبانی از قابلیت کشسانی منابع از مقیاس‌پذیری بسیار بالایی (به خصوص در مقایسه با سناریوی اول) برخوردار است.

پس از تحلیل و ارزیابی سناریوهایی که به توسعه‌ی سیستم یادگیری و آموزش الکترونیکی در محیط-های رایانشی بزرگ‌مقیاس می‌پردازند با استفاده از فنون و ابزارهای دسته‌ی اول (به عنوان نمونه

---

[۱۱۷] لازم به ذکر است که نرخ افزایش میانگین تاخیر تحت تاثیر سیاست‌های مدیریت منابع سیستم قرار دارد. همان‌طور که پیشتر اشاره شد، در این سناریو سیاست مدیریت منابع در سیستم به این ترتیب تعریف شده است که تنها پس از زیر بار رفتن کامل منابع محلی از منابع واقع در ابر سیستم استفاده شود. با بررسی بیشتر این موضوع در کارهای آتی، می‌توان با ایجاد تغییر در این سیاست، میانگین تاخیر در سرویس‌دهی را تنظیم کرد و برای مثال سیاست را به گونه‌ای تعریف کرد که نرخ افزایش میانگین تاخیر کاهش یابد.

کلُدسیم) در این زیربخش، در زیربخش ۵-۲ به طور خاص به چگونگی به‌کارگیری فنون و ابزارهای بزرگ‌مقیاس (دسته‌ی دوم) در ارایه‌ی سرویس‌های هوشمند یادگیری و آموزش الکترونیکی خواهیم پرداخت.

۵-۲- ارایه‌ی سرویس‌های هوشمند یادگیری و آموزش الکترونیکی با استفاده از فنون و ابزارهای شبیه‌سازی بزرگ‌مقیاس

دسته‌ی دوم از فنون و ابزارهایی که در این پژوهش مطرح شدند به نوبه‌ی خود نقش مهمی در ارایه و ارزیابی سرویس‌های یادگیری و آموزش الکترونیکی ایفا می‌کنند. با این وجود به‌کارگیری آن‌ها در خصوص سرویس‌های مقیاس‌پذیر[۱۱۸] مانند سرویس‌های «توصیه‌گر»[۱۱۹] که در سیستم‌های هوشمند یادگیری و آموزش الکترونیکی مورد نیازند اهمیت ویژه‌ی می‌یابد. یکی از اهداف اصلی در طراحی سیستم‌های هوشمند در یادگیری الکترونیکی که در حال حاضر نقش مهمی در بهبود فرآیند یادگیری الکترونیکی ایفا می‌کنند، ایجاد قابلیت تطبیق محتوای درسی ارایه شده، متناسب با نیازمندی‌های هر یک از افراد یادگیرنده است. این مساله به خصوص در سال‌های اخیر، با مطرح شدن و گسترش روزافزون بحث « درس‌های برخط باز و فراگیر»[۱۲۰] و در نتیجه افزایش حجم و تنوع مولفه‌های درسی ارایه شده در سیستم‌های یادگیری الکترونیکی بزرگ‌مقیاس، از اهمیت و پیچیدگی بیشتری برخوردار شده است. در همین راستا، در این زیربخش ابتدا بحث کوتاهی در خصوص سیستم‌های هوشمند آموزش و یادگیری الکترونیکی و به طور خاص انواع سرویس‌های توصیه‌گر در این حوزه خواهیم داشت. سپس به تبیین چگونگی به‌کارگیری فنون و ابزارهای دسته‌ی دوم - که به طور خاص برای اجرا در محیط‌های رایانشی بزرگ‌مقیاس طراحی شده‌اند - برای ارایه‌ی چنین سرویس‌هایی در سیستم یادگیری الکترونیکی می‌پردازیم.

تمام سیستم‌هایی که به گونه‌ای توانایی شبیه‌سازی رفتارهای آموزگار در هر یک از جنبه‌های آموزش را دارند و با تقلید این رفتارها، بدون نیاز به حضور آموزگار، از فرآیند یادگیری دانش‌آموزان پشتیبانی می‌کنند، تحت عنوان «سیستم‌های هوشمند آموزش»[۱۲۱] تعریف می‌شوند [۵۹، ۶۰]. در حال

---

[118] *scalable*
[119] *Recommender*
[120] *MOOC (Massive Online Open Course)*
[121] *Intelligent Tutoring System*

حاضر به‌کارگیری روی‌کرد سیستم‌های هوشمند در یادگیری الکترونیکی گسترش قابل ملاحظه‌ای یافته است و در آینده‌ای نزدیک این سیستم‌ها نقش مهمی در بهبود فرآیند یادگیری الکترونیکی ایفا می‌کنند [۵۹]. یکی از اهداف اصلی در طراحی این سیستم‌های هوشمند، ایجاد قابلیت تطبیق محتوای درسی ارایه شده، متناسب با نیازمندی‌های هر یک از افراد یادگیرنده در سیستم است [۷۱].

در راستای ایجاد قابلیت تطبیق‌پذیری در نحوه‌ی ارایه‌ی محتوای درسی، سرویس‌های توصیه‌گر، با هدف پیشنهاددهی مواد و یا به طور کلی مولفه‌های درسی مناسب به افراد یادگیرنده، به عنوان یکی از مهم‌ترین سرویس‌های قابل ارایه در سیستم‌های هوشمند یادگیری الکترونیکی شناخته شده‌اند و فنون متناظر با پیاده‌سازی آن‌ها در سال‌های اخیر توسعه‌ی زیادی یافته است [۷۴،۷۳،۷۲].

با گسترش روزافزون سیستم‌های یادگیری الکترونیکی بزرگ‌مقیاس و به ویژه مطرح شدن بحث «موک» در این حوزه، حجم و تنوع دروس ارایه شده از طریق این سیستم‌ها افزایش قابل توجهی یافته است. افزایش حجم و تنوع دروس قابل دسترس در سیستم‌های یادگیری الکترونیکی از یک سو و پیشرفت فنآوری‌های قابل بکارگیری در این حوزه از سوی دیگر، سطح انتظار کاربران و پیچیدگی نیازمندی‌های آن‌ها را به تدریج افزایش داده است. به گونه‌ای که امروزه سرویس‌های متداول در سیستم‌های یادگیری الکترونیکی دیگر در جواب‌گویی به نیازمندی‌های کاربران کافی نیست و به منظور پیشنهاد مولفه‌های درسی مناسب به آنها، ارایه‌ی سرویس‌های پیشرفته‌تری مانند سرویس‌های توصیه‌گر محتوای درسی از جمله قابلیت‌های ضروری این سیستم‌ها محسوب می‌شود. در واقع این سرویس‌ها را می‌توان به عنوان سرویس‌های ارزش افزوده‌ای[۱۲۲] در نظر گرفت که با استفاده از اطلاعات غنی موجود و یا قابل جمع‌آوری در سیستم‌های یادگیری الکترونیکی بزرگ‌مقیاس (مانند سیستم‌هایی که در حوزه‌ی موک در حال فعالیت هستند)، امکان ارایه‌ی خدماتی فراتر از سرویس‌های ساده در سیستم‌های یادگیری الکترونیکی متداول را فراهم ساخته‌اند. با ارایه‌ی چنین سرویس‌های ارزش افزوده‌ای، این سیستم‌ها از حالت کنش‌پذیر[۱۲۳] گذر کرده‌اند و به سیستم‌های فعال و کنش-گری[۱۲۴] تبدیل شده‌اند که دارای قابلیت انعطاف‌پذیری بالایی در تطبیق با نیازمندی‌های کاربران خود است و می‌تواند محتوای درسی مناسب افراد یادگیرنده و یا گروه‌های مختلف را، به ترتیب متناسب با نیاز شخصی و یا گروهی آن‌ها تامین نماید.

---

[122] *Value-added services*
[123] *passive*
[124] *Proactive*

فنون توصیه را می‌توان بر اساس نحوه‌ای که به ارایه‌ی پیشنهاد می‌پردازند، به چند نوع کلی تقسیم نمود. از جمله انواع اصلی، می‌توان به فنون «مبتنی بر آمار»[۱۲۵]، «مبتنی بر قانون»[۱۲۶]، «مبتنی بر محتوا»[۱۲۷] و مبتنی بر «پالایش مشترک»[۱۲۸] اشاره کرد. از میان این انواع، فنون مبتنی بر محتوا و مبتنی بر پالایش مشترک به صورت گسترده‌تری مورد استفاده قرار گرفته‌اند. در روی‌کردهای مبتنی بر محتوا، سرویس‌های توصیه‌گر بر اساس میزان مشابهت[۱۲۹] میان نمایه[۱۳۰](پروفایل) کاربران و مولفه‌ها، به ارایه‌ی پیشنهاد به کاربران می‌پردازد. در مقابل در روی‌کردهای مبتنی بر پالایش مشترک، مبنای پیشنهادات ارایه شده توسط سرویس‌های توصیه‌گر، وابستگی[۱۳۱] بین کاربران و یا بین مولفه‌هاست. به این ترتیب که به ازای هر کاربر/ مولفه‌ی درسی مورد نظر، پروفایل آن کاربر/ مولفه با سایر کاربران/ مولفه‌ها مقایسه می‌شود، کاربران/ مولفه‌های مشابه با آن شناسایی می‌شوند، و در نهایت بر اساس مشترکات میان آن‌ها، گزینه‌های پیشنهادی مناسب به کاربر توصیه می‌گردد.

با گسترش سیستم‌های یادگیری الکترونیکی بزرگ‌مقیاس و افزایش تعداد کاربران و حجم مولفه‌های درسی ارایه شده در آن‌ها، ارایه‌ی سرویس‌های پیشرفته‌ای مانند سرویس‌های توصیه‌گر در سطوح بالا، مستلزم انجام عملیات رایانشی حجیمی در سطوح پایین است. به بیان دقیق‌تر، به منظور ارایه‌ی پیشنهادهای باکیفیت و مبتنی بر فنون و الگوریتم‌های توصیه‌گر در سیستم‌های یادگیری الکترونیکی، به خصوص در صورت پیاده‌سازی این سرویس‌ها در مقیاس‌های بزرگ، صرف‌نظر از این که کدام یک از انواع فنون ذکر شده به‌کار گرفته شوند، اغلب لازم است تا میلیون‌ها عملیات محاسباتی در سطح بستر رایانشی با موفقیت انجام گیرد. از این‌رو در حالی‌که طراحی سرویس‌های توصیه‌گر به منظور شناسایی و دست‌یابی به مولفه‌های درسی مورد نظر کاربران از چالش‌های مطرح در حوزه یادگیری الکترونیکی به‌شمار می‌رود، در نظر گرفتن قابلیت مقیاس‌پذیری این سرویس بر اهمیت این مساله می‌افزاید. از این‌رو در این پژوهش، فنون و ابزارهای دسته‌ی دوم را - که به طور خاص برای اجرا در محیط‌های رایانشی بزرگ‌مقیاس طراحی شده‌اند - برای ارایه‌ی چنین سرویس‌هایی در سیستم یادگیری الکترونیکی انتخاب نموده‌ایم. همان‌طور که در بخش چهارم مطرح شد، یکی از نیازمندی‌های اصلی فنون و ابزارهای شبیه‌سازی بزرگ‌مقیاس (دسته‌ی دوم)، نیاز به توزیع عملیات رایانشی بر روی مجموعه‌ای از رایانه‌هایی است که روی هم رفته به عنوان یک خوشه‌ی سخت‌افزاری

---

[125] *Statistics-based*
[126] *Rule-based*
[127] *Content-based*
[128] *Collaborative filtering*
[129] *Similarity*
[130] *Profile*
[131] *Correlation*

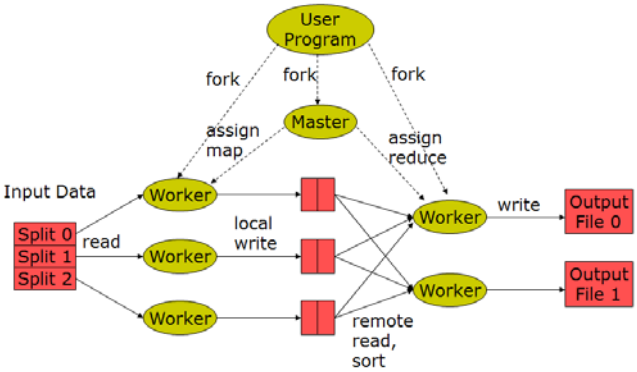
شناخته می‌شوند. بنابراین، در ادامه‌ی این زیربخش در قالب دو سناریوی متفاوت، به طور دقیق به چگونگی به‌کارگیری این فنون و ابزارها در چارچوب «نگاشت کاهش»[۱۳۲]، به منظور ایجاد قابلیت مقیاس‌پذیری در طراحی سرویس‌های توصیه‌گر در سیستم‌های یادگیری الکترونیکی خواهیم پرداخت. به بیان دقیق‌تر، در هر یک از این سناریوها یکی از روی‌کردهای متداولی که در طراحی سرویس‌های توصیه‌گر در پیش‌گرفته می‌شود را مورد ارزیابی قرار خواهیم داد. همان‌طور که پیش‌تر اشاره شد، فنون «مبتنی بر آمار»، «مبتنی بر قانون»، «مبتنی بر محتوا» و مبتنی بر «پالایش مشترک» از انواع صورت‌های مطرح در ارایه‌ی سرویس‌های توصیه‌گر به شمار می‌روند، که از میان این انواع، فنون مبتنی بر محتوا و مبتنی بر پالایش مشترک به صورت گسترده‌تری مورد استفاده قرار گرفته‌اند. از این رو سناریوهایی که در ادامه برای بررسی انتخاب نموده‌ایم، به طور خاص به چگونگی ارایه‌ی سرویس‌های توصیه‌گر مبتنی بر محتوا و مبتنی بر پالایش مشترک در سیستم آموزش و یادگیری الکترونیکی می‌پردازند. پیش از پرداختن به این سناریوها، در ادامه ابتدا معرفی کوتاهی بر چارچوب پردازشی نگاشت کاهش خواهیم داشت.

نگاشت کاهش چارچوبی پردازشی است که از جانب شرکت گوگل برای پشتیبانی از رایانش توزیع‌شده و بزرگ‌مقیاس ارایه شده است. برتری چارچوب نگاشت‌کاهش، در این است که اجازه می‌دهد تا انجام عملیات‌های پردازش و کاهش توزیع شود. فراهم آوردن این امکان مستلزم آن است که هر کدام از این نگاشت‌ها مستقل از دیگران باشد، و این استقلال خود متضمن اجرای موازی نگاشت‌هاست.

ساختار منطقی این چارچوب در شکل ۱۶ نمایش داده شده است. توابع نگاشت و کاهش از نگاشت‌کاهش با استفاده از ساختار داده‌ای زوج مرتب (کلید، مقدار) نمایش داده می‌شود. نگاشت یک مجموعه از زوج مرتب‌های (کلید، مقدار) از داده‌ای با نوع دامنه ورودی را گرفته دریافت نموده و مجموعه‌ای از زوج مرتب‌های (کلید، مقدار) از دامنه‌ای دیگر را به عنوان خروجی تولید می نماید. تابع کاهش مجموعه ای از زوج مرتب‌های (کلید، (لیستی از مقادیر)) را مصرف نموده و مجموعه‌ای از زوج مرتب‌های (کلید، مقدار) را بازمی‌گرداند.

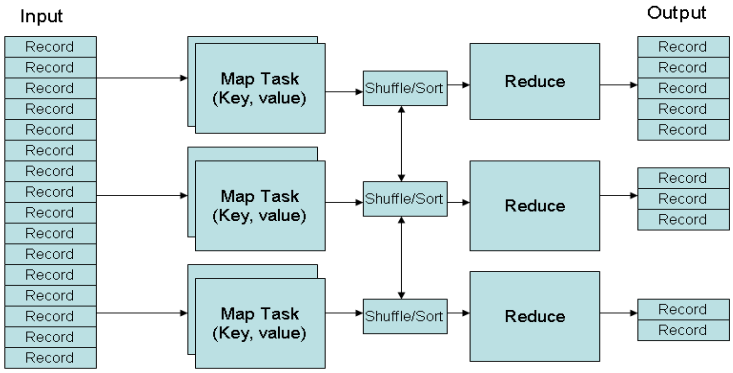
---

[132] *MapReduce*



شکل ۱۶- ساختار منطقی چارچوب پردازشی نگاشت کاهش

البته در بین نگاشت‌ها و کاهش‌ها یک فاز نهفته جابه‌جایی[۱۳۳] و مرتب‌سازی[۱۳۴] نیز وجود دارد که در آنها کلیدهای مشابه از کلیه نگاشت‌ها تجمیع شده و پس از مرتب سازی آنها را در قالب (کلید، (لیستی از مقادیر)) برای یک کاهش مشخص ارسال می شود. طرح کلی معماری این چارچوب پردازشی در شکل ۱۷ نمایش داده شده است.



شکل۱۷- طرح معماری نگاشت کاهش

از میان راه‌کارهای آزاد و متن‌بازی که از این چارجوب پردازشی الگوگیری می‌کنند، به‌کارگیری «هدوپ»[۱۳۵] [۶۲] که با استفاده از جاوا پیاده‌سازی شده است، گسترش قابل ملاحظه‌ای یافته است. برنامه‌هایی که در این قالب نوشته می‌شوند، به طور خودکار موازی‌سازی شده و بر روی خوشه‌ای از

---

[133] *Shuffling*
[134] *Sorting*
[135] *Hadoop, available: http://hadoop.apache.org/core/*

ماشین‌های محاسباتی پیاده‌سازی می‌شوند. به منظور اجرای الگوی نگاشت‌کاهش در هدوپ، یک «ردگیر کار»[۱۳۶] نقش گره‌ی اصلی را ایفا می‌کند و عملیات پردازشی را قسمت‌بندی کرده و هر قسمت را به عنوان ورودی در اختیار یکی از نگاشت‌ها قرار می‌دهد. سپس نتایج حاصل از اعمال این توابع در فایل‌سیستم محلی که «فایل‌سیستم توزیع‌شده‌ی هدوپ»[۱۳۷] (یا به طور مخفف اِچ‌دی‌اِف‌اِس) نام دارد ذخیره می‌شود. هدوپ زمان‌بندی عملیات پردازشی مورد نظر را به‌گونه‌ای تنظیم می‌کند که گذردهی ورودی/خروجی خوشه افزایش، هزینه‌ی محاسباتی عملیات کاهش و در نهایت کارایی کلی فرآیند پردازشی بهبود یابد.

۵-۲-۱- ارایه‌ی سرویس‌های توصیه‌گر مبتنی بر محتوا

یکی از روی‌کردهای متداولی که در طراحی سرویس‌های توصیه‌گر در پیش‌گرفته می‌شود، ارایه‌ی پیشنهاد مبتنی بر محتوا به کاربران است. روش‌های توصیه‌ی مبتنی بر محتوا اغلب بر اساس اطلاعات و خصوصیات آیتم‌هایی که سیستم به ارایه‌ی پیشنهاد درباره‌ی آن‌ها می‌پردازد، شکل می‌گیرند. به بیان کلی، در این روش‌ها سعی می‌شود تا بر اساس تجربه‌های موجود کاربر در سیستم، به پیشنهاد آیتم-های جدیدی که به نسبت بیشترین نزدیکی را با اولویت‌های کاربر دارند پرداخته شود. بنابراین در این روش از یک‌سو به مدل‌سازی آیتم‌ها بر اساس ویژگی‌های مرتبط آن‌ها پرداخته می‌شود. از سوی دیگر نیز اولویت‌های کابران بر اساس این ویژگی‌ها مدل‌سازی شده و مشخص می‌شوند. در نهایت با مقایسه‌ی این مدل‌ها، آیتم‌های مناسب به کاربر پیشنهاد می‌گردد.

با این مقدمه می‌توان به نقش کلیدی مفهوم «بردار ویژگی‌ها» و اهمیت ارتباط آن با اولویت‌های کاربر، در طراحی سیستم‌های توصیه‌گر مبتنی بر محتوا پی برد. برای گردآوری تجربه‌های کاربران در سیستم (به منظور استخراج اولویت‌های آن‌ها)، راه‌های گوناگونی وجود دارد. برخی از این راه‌ها عبارتند از:

- ساخت پروفایل کاربران توسط خود آن‌ها: واگذاری کامل ساخت پروفایل به کـابران اغلـب مطلوبیت چندانی ندارد، و بهتر است تنها امکان ویرایش آن به کاربران داده شود.

---

[136] *Job tracker*
[137] *Hadoop Distributed File System (HDFS)*

- استخراج پروفایل کاربران از فعالیت‌های آن‌ها: می‌توان پروفایل کاربران را به صورت ضمنی بر اساس فعالیت‌هایی مانند ثبت‌نام در دروس، مرور محتوا و یا بارگیری[۱۳۸] منابع درسی توسط آن‌ها ایجاد کرد.
- استخراج پروفایل کاربران از طریق نظرسنجی: می‌توان پروفایل کاربران را به صورت آشکارا بر اساس نتایج نظرسنجی آن‌ها در مورد آیتم‌ها (مولفه‌های درسی) ایجاد کرد.

البته باید توجه داشت که در صورت دنبال کردن هر یک از این راه‌ها (یا به تنهایی و یا به صورت ترکیبی)، همواره مساله‌ی نگاشت از فضای اولویت کاربران به فضای ویژگی‌های آیتم‌ها را باید در نظر داشت. برای مثال فرض می‌کنیم که می‌خواهیم بر اساس نظرسنجی‌های کاربران به ایجاد پروفایل آن‌ها بپردازیم. در این صورت برای نگاشت بین فضای اولویت کاربران به فضای ویژگی‌های آیتم‌ها، می‌توان تعدادی کلیدواژه در اختیار کاربران قرار داد تا درباره‌ی هریک از آن‌ها، نظرات خود را به شکل یکی از سه حالت «علاقه‌مند»، «بی‌علاقه» و یا «بدون نظر» در سیستم اعلام نمایند. به این ترتیب می‌توان واژگان کلیدی اولویت کاربران را به‌دست آورد. حال در این مرحله، با فرض این‌که واژگان کلیدی اولویت کاربران را در اختیار داشته باشیم، چگونه به پیشنهاد آیتم‌های مناسب به آن‌ها بپردازیم؟ در ادامه این بخش به متداول‌ترین الگویی که به این منظور استفاده می‌شود و قابلیت طراحی با مقیاس‌پذیری بالایی نیز دارد می‌پردازیم.

به طور کلی، به منظور پیشنهاد آیتم‌های مناسب به کاربران باید دو عامل اصلی را در نظر گرفت: ۱) ممکن است فرکانس برخی واژه‌ها نسبت به واژه‌های دیگر در حد قابل ملاحظه‌ای بالاتر باشد، ۲) میزان مرتبط بودن واژگان مختلف در پیشنهاد آیتم‌ها یکسان نیست. جهت لحاظ این دو عامل در ارایه‌ی پیشنهاد به کاربران از روش وزن‌دهی «تی‌اِف‌آی‌دی‌اِف»[۱۳۹] استفاده می‌شود که در آن وزن-دهی به واژگان به ترتیب زیر محاسبه می‌شود:

وزن واژه بر حسب تی‌اِف‌آی‌دی‌اِف = فرکانس واژه × عکس فرکانس آیتم

که در آن فرکانس واژه تعداد دفعاتی است که واژه در یک آیتم تکرار شده است. عکس فرکانس آیتم، میزان کم‌یاب بودن آیتم‌هایی که شامل واژه‌ی مورد نظر هستند را نشان می‌دهد و معمولا با

---

[۱۳۸] *download*

[۱۳۹] *Term Frequency Inverse Document Frequency (TFIDF)*

اندازه‌گیری لگاریتم نسبت تعداد آیتم‌ها به تعداد آیتم‌های شامل واژه محاسبه می‌شود. بیان رسمی تی-اِف‌آی‌دی‌اِف به صورت زیر است:

که $|D|$ تعداد کل آیتم‌ها (مولفه‌های درسی) موجود در سیستم یادگیری الکترونیکی است، و $|\{d: t_i \in d\}|$ تعداد آیتم‌هایی است که واژه‌ی موردنظر در آن‌ها به‌کار رفته است. مفهوم تی‌اِف-آی‌دی‌اِف را می‌توان به خوبی در سرویس‌های توصیه‌گر مبتنی بر محتوا در سیستم‌های یادگیری الکترونیکی به‌کار گرفت. به این ترتیب که هر یک از مولفه‌های درسی موجود در سیستم را به صورت بردار وزن‌داری از برچسب[۱۴۰]های آن در نظر می‌گیریم . با محاسبه‌ی این وزن‌ها بر حسب تی‌اِف‌آی‌دی‌اِف به ایجاد پروفایلی برای هر یک از این مولفه‌های درسی می‌پردازیم. با استفاده از این پروفایل‌های تی‌اِف‌آی‌دی‌اِف و همچنین نتایج نظرسنجی‌های کاربران، پروفایل‌های کاربران را با توجه به اولویت‌های آن‌ها ایجاد می‌نماییم. در نهایت با تطبیق و مقایسه‌ی این پروفایل‌ها با سایر مولفه-های درسی که در حال حاضر در سیستم یادگیری الکترونیکی ارایه می‌شود، به پیشنهاد مولفه‌های درسی مناسب به افراد یادگیرنده می‌پردازیم.

به بیان دقیق‌تر روش فوق بر پایه‌ی مدل فضای برداری [۶۱] ارایه شده است. در این مدل، مجموعه‌ی تمام حالت‌های ممکن برای کلیدواژه‌ها، فضای محتوا را شکل می‌دهند که هر کلیدواژه یک بعد از این فضا محسوب می‌شود. به این ترتیب، متناظر با هر آیتم و هر کاربر بردار جداگانه‌ای در این فضا تعریف می‌شود که بر حسب نزدیکی این دو بردار به هم‌دیگر، میزان مطابقت ویژگی‌های آیتم با اولویت‌های کاربر سنجیده می‌شود. نمایش ویژگی‌های آیتم‌ها بر حسب بردار ویژگی‌ها می‌تواند به یکی از صورت‌های زیر باشد:

- صفر و یک ساده، بنا بر این‌که کدام یک از کلیدواژه‌ها در برچسب‌های آیتم استفاده شده باشد.
- شمارش تعداد دفعاتی که هر یک از کلیدواژه‌ها در برچسب‌های آیتم استفاده شده باشد.
- محاسبه‌ی وزن‌های متناظر با کلیدواژه‌ها بر حسب تی‌اِف‌آی‌دی‌اِف.

---

[140] *tag*

ما در این طرح پژوهشی از بین روش‌های نمایش ویژگی‌های آیتم‌ها بر حسب بردار ویژگی‌ها، روش وزن‌دهی کلیدواژه‌ها بر حسب تی‌اِف‌آی‌دی‌اِف را با توجه به قابلیت مقیاس‌پذیری بالای آن انتخاب می‌کنیم و در ادامه به طراحی بزرگ‌مقیاس این روش در ارایه‌ی سرویس‌های توصیه‌گر مبتنی بر محتوا در سیستم‌های یادگیری الکترونیکی می‌پردازیم.

همان‌طور که در بخش پیش اشاره شد، به منظور ارایه‌ی سرویس‌های توصیه‌گر مبتنی بر محتوا در مقیاس بزرگ، چارچوب پردازشی نگاشت‌کاهش را مبنای طراحی خود قرار می‌دهیم. مرحله‌هایی که بار محاسباتی سنگینی بر سیستم اعمال می‌کنند و از این رو باید به صورت مقیاس‌پذیری پیاده‌سازی شوند عبارتند از:

- محاسبه‌ی تعداد دفعات تکرار هر واژه در برچسب هر آیتم
- محاسبه‌ی تعداد واژه‌های موجود در برچسب هر آیتم
- به ازای هر واژه، محاسبه‌ی تعداد آیتم‌هایی که واژه در برچسب آن‌ها به‌کار رفته است.

بر این اساس، می‌توان سیستم توصیه‌گر مبتنی بر محتوا با استفاده از روش تی‌اِف‌آی‌دی‌اِف را در چهار لایه نگاشت و کاهش، طراحی و پیاده‌سازی نمود. جزییات طرح ارایه‌ی سرویس توصیه‌گر محتوا در سیستم‌های یادگیری الکترونیکی در شکل ۱۸ نشان داده شده است. معماری این طرح به شرح زیر است.

معماری نگاشت و کاهش در لایه‌ی اول:

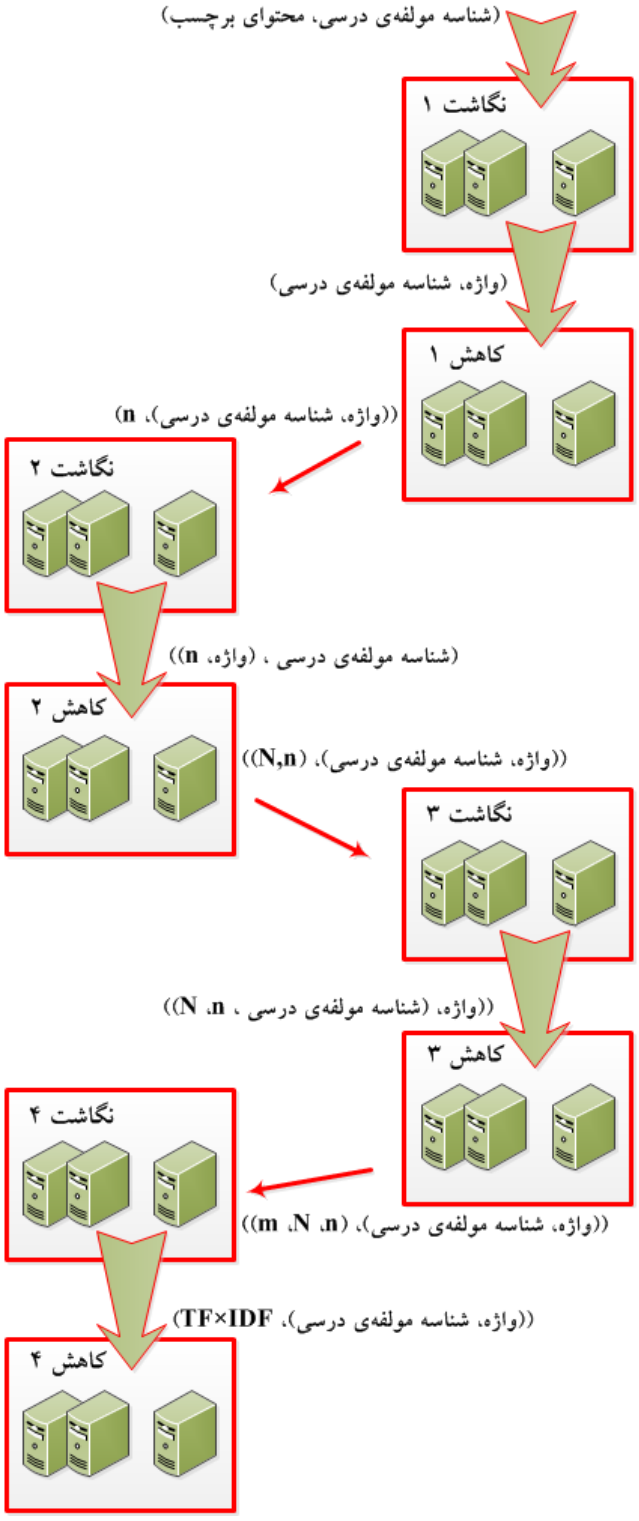
نگاشت ۱

- ورودی: زوج مرتب (شناسه مولفه‌ی درسی، محتوای برچسب)
- خروجی: زوج مرتب (واژه، شناسه مولفه‌ی درسی)

کاهش ۱

- ورودی: زوج مرتب (واژه، شناسه مولفه‌ی درسی)
- خروجی: زوج مرتب ((واژه، شناسه مولفه‌ی درسی)، n)

که به ازای هر (واژه، شناسه مولفه‌ی درسی)، n تعداد دفعاتی است که واژه در برچسب آن تکرار شده است. بنابراین در لایه‌ی اول معماری طراحی شده تعداد دفعاتی که هر واژه در هر مولفه‌ی درسی موجود در سیستم یادگیری الکترونیکی تکرار شده است مشخص می‌شود.



شکل ۱۸- طراحی سیستم توصیه‌گر مولفه‌های درسی مبتنی بر محتوا

معماری نگاشت و کاهش در لایه‌ی دوم:

نگاشت ۲

- ورودی: زوج مرتب ((واژه، شناسه مولفه‌ی درسی)، n)
- خروجی: زوج مرتب (شناسه مولفه‌ی درسی ، (واژه، n))

کاهش ۲

- ورودی: زوج مرتب (شناسه مولفه‌ی درسی ، (واژه، n))
- خروجی: زوج مرتب ((واژه، شناسه مولفه‌ی درسی)، (N,n))

که به ازای هر (واژه، شناسه مولفه‌ی درسی)، N تعداد کل واژه‌های به‌کار رفته در برچسب مولفه‌ی درسی و n تعداد دفعات تکرار واژه‌ی مورد نظر در آن است. بنابراین در لایه‌ی دوم معماری طراحی شده، با محاسبه‌ی مجموع nهایی که به ازای هر یک از واژه‌های موجود در برچسب یک مولفه‌ی درسی به‌دست آمده است (از لایه‌ی قبل)، تعداد کل واژه‌های به‌کار رفته در برچسب هر مولفه‌ی درسی موجود در سیستم یادگیری الکترونیکی مشخص می‌شود.

معماری نگاشت و کاهش در لایه‌ی سوم:

نگاشت ۳

- ورودی: زوج مرتب ((واژه، شناسه مولفه‌ی درسی)، (N ،n))
- خروجی: زوج مرتب ((واژه، (شناسه مولفه‌ی درسی ، n، N))

کاهش ۳

- ورودی: زوج مرتب ((واژه، (شناسه مولفه‌ی درسی ، n، N))
- خروجی: زوج مرتب ((واژه، شناسه مولفه‌ی درسی)، (n، N، m))

که به ازای هر واژه m تعداد دفعاتی است که واژه در برچسب‌های کل مولفه‌ی درسی به‌کار رفته و یا تکرار شده است. به این ترتیب در لایه‌ی سوم معماری طراحی شده، با محاسبه‌ی مجموع nهایی که به ازای هر یک از واژه‌ها در برچسب‌های مولفه‌ی درسی مختلف به‌دست آمده است، تعداد دفعاتی که واژه در کل مولفه‌های درسی موجود در سیستم یادگیری الکترونیکی تکرار شده است مشخص می‌شود.

معماری نگاشت و کاهش در لایه‌ی چهارم:

نگاشت ۴

- ورودی: زوج مرتب ((واژه، شناسه مولفه‌ی درسی)، $(n، N، m)$)
- خروجی: زوج مرتب ((واژه، شناسه مولفه‌ی درسی)، $TF \times IDF$)

کاهش ۴

- تابع کاهش در این لایه تنها نقش جمع‌آوری نتایج را بر عهده دارد.

به این ترتیب با فرض این‌که تعداد کل مولفه‌های درسی موجود در سیستم یادگیری الکترونیکی معلوم باشد (در صورت نامعلوم بودن این تعداد، می‌توان آن را با استفاده از الگوی نگاشت‌کاهش به‌دست آورد)، سیستم توصیه‌گر مبتنی بر محتوا، بر اساس معماری ارایه شده، می‌تواند مولفه‌های بردار کلیدواژه‌های متناظر با هر مولفه‌ی درسی را بر حسب تی‌اِف‌آی‌دی‌اِف وزن‌دهی کرده و با استفاده از این بردارها و بردارهای اولویت کاربران (همان‌طور که پیش‌تر توضیح داده شد) به پیشنهاد مولفه‌های درسی مناسب به افراد یادگیرنده بپردازد.

۵-۲-۲- ارایه‌ی سرویس‌های توصیه‌گر مبتنی بر پالایش مشترک

یکی از روی‌کردهایی که در ارایه‌ی سرویس‌های توصیه‌گر از موفقیت بسیار بالایی برخوردار بوده و در سطح گسترده‌ای استفاده شده است، به‌کارگیری فنون پالایش مشترک در این سرویس‌هاست [۶۳-۶۸]. در این بخش به طراحی مقیاس‌پذیر سرویس‌های توصیه‌گر مبتنی بر پالایش مشترک در سیستم یادگیری الکترونیکی می‌پردازیم.

فنون پالایش مشترک به روش‌های گوناگونی در سیستم‌های توصیه‌گر به‌کار گرفته می‌شوند، اما بسیاری از این روش‌ها را می‌توان به دو دسته‌ی کلی تقسیم‌بندی نمود:

- پالایش مشترک مبتنی بر کاربر
- پالایش مشترک مبتنی بر آیتم

اساس کار فنونی که در دسته‌ی اول قرار می‌گیرند به این ترتیب است که ابتدا کاربرانی که با کاربر مورد نظر (کاربری که سیستم قصد ارایه‌ی پیشنهاد به آن را دارد) الگوی نظرسنجی مشابهی دارند شناسایی می‌شوند. سپس با استفاده از نظرسنجی‌های این دسته از کاربران شناسایی شده، پیشنهاد مناسب به کاربر مورد نظر پیش‌بینی و ارایه می‌شود. در دسته‌ی دوم از فنون پالایش مشترک، ابتدا ماتریسی که روبط بین هر جفت از آیتم‌ها را تعیین می‌کند ساخته می‌شود. سپس با مقایسه و بررسی

میزان تطابق پروفایل کاربر مورد نظر با ماتریس به‌دست آمده، پیشنهاد مناسب به کاربر استنتاج می‌شود.

فنون پالایش مشترک مبتنی بر آیتم در مقایسه با فنون پالایش مشترک مبتنی بر کاربر از مقیاس‌پذیری بالاتری برخوردارند و از این رو در این طرح پژوهشی ما در طراحی سرویس‌های توصیه‌گر در سیستم یادگیری الکترونیکی بر روی این دسته از فنون تمرکز می‌کنیم. به بیان دقیق‌تر، اغلب با بزرگ شدن مقیاس سیستم با این مساله مواجه خواهیم شد که تعداد نظرسنجی‌های کاربران در مقایسه با مجموعه‌ی عظیم مولفه‌های درسی موجود در سیستم یادگیری الکترونیکی اندک است و از این‌رو در بسیاری موارد امکان ارایه‌ی پیشنهاد مناسب بر اساس پالایش مشترک مبتنی بر کاربر در مقیاس بزرگ وجود ندارد. از سوی دیگر با توجه به این‌که اغلب تغییرات مجموعه‌ی کاربران نسبت به مجموعه‌ی مولفه‌های درسی بسیار بیشتر است، ارایه‌ی پیشنهادهای مناسب در تطبیق با این تغییرات، نیازمند صرف هزینه و تحمیل بار محاسباتی بیشتری خواهد بود.

یکی از مراحل اصلی در فنون پالایش مشترک، انجام محاسبات مورد نیاز برای یافتن میزان مشابهت میان آیتم‌ها (مولفه‌های درسی) و سپس انتخاب مشابه‌ترین آیتم‌هاست. ایده‌ی پایه و کلی در محاسبه‌ی شباهت بین دو آیتم i و j این است که ابتدا کاربرانی که در مورد هر دو آیتم نظر داده‌اند را از سایر کاربران جدا کرده و سپس با در پیش گرفتن یکی از روی‌کردهای محاسبه‌ی شباهت، میزان شباهت s_{ij} تعیین شود. در این مقاله ما از روش همبستگی پیرسون[۱۴۱] برای اندازه‌گیری میزان شباهت i و j استفاده می‌کنیم. فرض می‌کنیم کاربرانی که در مورد هر دو آیتم i و j نظر داده‌اند با U نمایش داده شود، در این صورت میزان شباهت i و jبر حسب همبستگی پیرسون برابر خواهد بود با:

که R_{ui} نشان‌دهنده‌ی نظرسنجی کاربر u در مورد آیتم i و $\bar{R}_i$ میانگین نتایج نظرسنجی روی آیتم i است. پس از این که بر اساس این معیار مشابه‌ترین آیتم‌ها به آیتم i شناسایی شدند، باید میزان مناسب بودن پیشنهاد این آیتم به کاربر مورد نظر پیش‌بینی شود. به این منظور از جمع وزن‌دار نتایج نظرسنجی کاربر مورد نظر در مورد مجموعه آیتم‌های U (که مشابه‌ترین آیتم‌ها به i هستند) استفاده می‌کنیم. در

[141] *Pearson correlation*

این جمع وزن‌دار، وزن اختصاص‌یافته به نظرسنجی هر آیتم در U برابر میزان مشباهت محاسبه شده بین آن آیتم و i است [۷۰].

با توجه به هزینه و بار محاسباتی بالای فنون پالایش مشترک، مساله‌ی مقیاس‌پذیری در طراحی سیستم‌های توصیه‌گر مبتنی بر این فنون از اهمیت قابل ملاحظه‌ای برخوردار است [۶۹-۶۸]. مرحله-هایی که بار محاسباتی سنگینی بر سیستم اعمال می‌کنند و از این رو باید به صورت مقیاس‌پذیری پیاده‌سازی شوند عبارتند از:

- محاسبه‌ی میانگین نظرسنجی‌ها برای هر آیتم
- محاسبه‌ی مشابهت بین هر جفت از آیتم‌ها
- محاسبه‌ی آیتم‌های پیش‌بینی شده برای پیشنهاد به کاربر مورد نظر

همان طور که پیش‌تر بحث شد، به منظور ارایه‌ی سرویس‌های توصیه‌گر مبتنی بر پالایش مشترک در مقیاس بزرگ، چارچوب پردازشی نگاشت‌کاهش را مبنای طراحی خود قرار می‌دهیم. بر این اساس، می‌توان سرویس فوق را در چهار لایه نگاشت و کاهش، طراحی و پیاده‌سازی نمود. جزییات طرح ارایه‌ی سرویس توصیه‌گر مبتنی بر پالایش مشترک در سیستم‌های یادگیری الکترونیکی در شکل ۱۹ نشان داده شده است. معماری این طرح به شرح زیر است.

معماری نگاشت و کاهش در لایه‌ی اول:

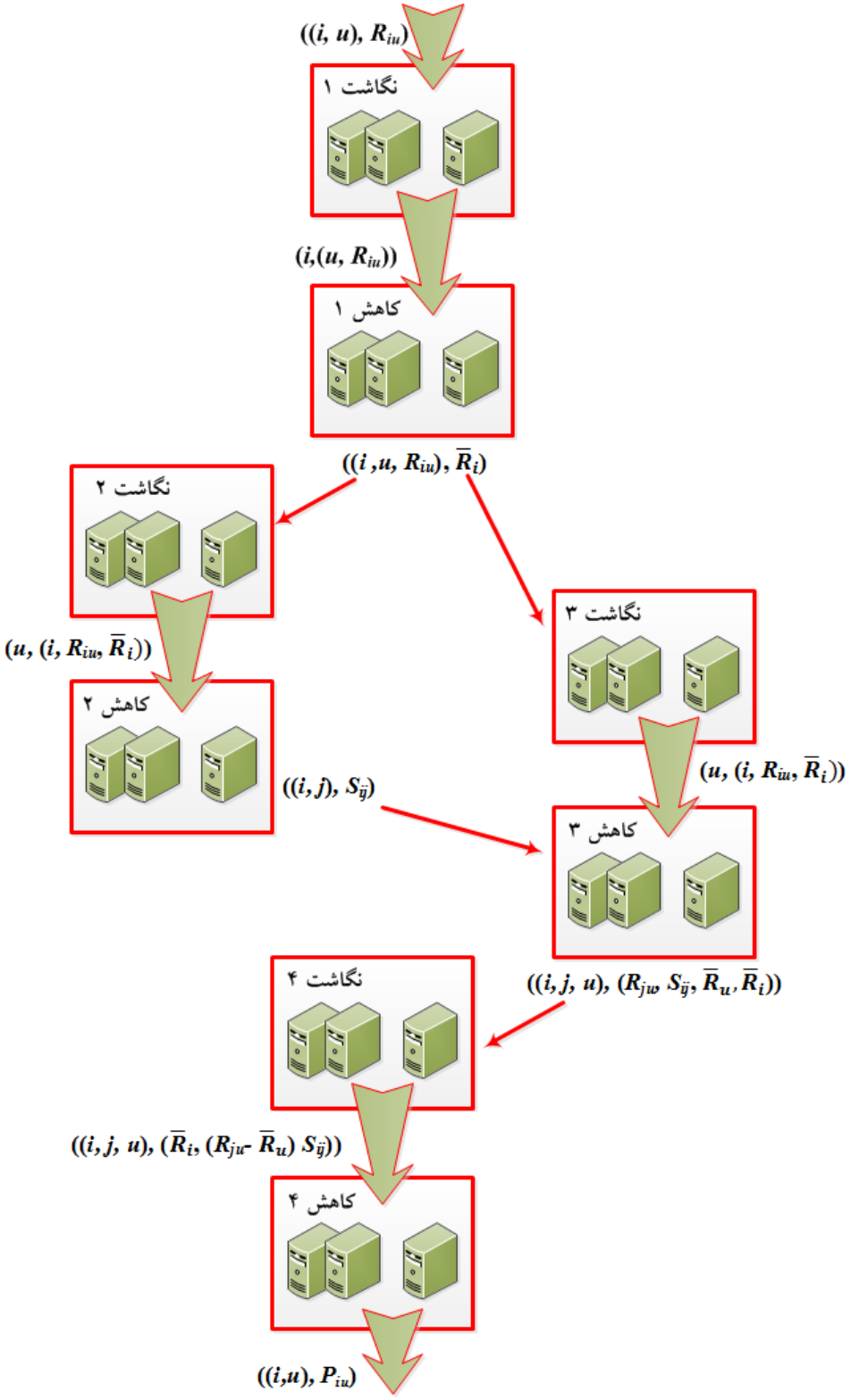
نگاشت ۱

- ورودی: زوج مرتب $((i, u), R_{iu})$
- خروجی: زوج مرتب $(i,(u, R_{iu}))$

کاهش ۱

- ورودی: زوج مرتب $(i,(u, R_{iu}))$
- خروجی: $((i ,u, R_{iu}), \bar{R}_i)$

در این لایه از معماری، مقادیر نظرسنجی که به ازای هر کاربر u در مورد هر مولفه‌ی درسی i وجود دارد به عنوان ورودی گرفته می‌شود و پس از دسته‌بندی این مقادیر به ازای هر یک از مولفه‌های درسی، میانگین مقدار نظرسنجی کاربران مختلف به ازای هر مولفه محاسبه می‌شود. سپس مقادیر به



شکل ۱۹ – طراحی سیستم توصیه‌گر مولفه‌های درسی مبتنی بر پالایش مشترک

دست آمده در اختیار لایه‌ی دوم قرار می‌گیرد. در لایه‌ی دوم، ابتدا مولفه‌های درسی بر حسب کاربرانی که در مورد آن‌ها نظر داده‌اند دسته‌بندی می‌شوند تا به این ترتیب جفت آیتم‌های (i,j) ای که کاربر مشترکی در مورد هر دوی آن‌ها نظر داده است مشخص شوند. سپس میزان شباهت i و j بر حسب همبستگی پیرسون محاسبه می‌شود.

معماری نگاشت و کاهش در لایه‌ی دوم:

نگاشت ۲

- ورودی: زوج مرتب $((i, u, R_{iu}), \bar{R}_i)$
- خروجی: زوج مرتب $(u, (i, R_{iu}, \bar{R}_i))$

کاهش ۲

- ورودی: زوج مرتب $(u, (i, R_{iu}, \bar{R}_i))$
- خروجی: $((i, j), S_{ij})$

در لایه‌ی سوم و چهارم معماری، عملیات لازم برای محاسبه‌ی مولفه‌های درسی پیش‌بینی شده برای پیشنهاد به کاربر مورد نظر انجام می‌گیرد. به این منظور معماری این دو لایه به شکل زیر است.

معماری نگاشت و کاهش در لایه‌ی سوم:

نگاشت ۳

- ورودی: زوج مرتب $((i, u, R_{iu}), \bar{R}_i)$
- خروجی: زوج مرتب $(u, (i, R_{iu}, \bar{R}_i))$

کاهش ۳

- ورودی: زوج مرتب $((u, (i, R_{iu}, \bar{R}_i))$ و $((i, j), S_{ij})$
- خروجی: $((i, j, u), (R_{ju}, S_{ij}, \bar{R}_u, \bar{R}_i))$

باید به دو نکته در طراحی این معماری دقت کرد. نکته‌ی اول این است که نگاشت لایه‌ی سوم ورودی خود را از خروجی لایه‌ی اول دریافت می‌کند و مشابه نگاشت لایه‌ی دوم مولفه‌های درسی بر حسب کاربرانی که در مورد آن‌ها نظر داده‌اند دسته‌بندی می‌شوند. نکته‌ی دوم این است که کاهش

لایه‌ی سوم، علاوه بر خروجی نگاشت لایه‌ی سوم، خروجی لایه‌ی دوم را نیز به عنوان ورودی دریافت می‌کند و با ترکیب این دو ورودی، خروجی مورد نظر را تولید می‌نماید.

در نهایت معماری نگاشت و کاهش در لایه‌ی چهارم:

نگاشت ۴

- ورودی: زوج مرتب $((i, j, u), (R_{ju}, S_{ij}, \bar{R}_u, \bar{R}_i))$
- خروجی: زوج مرتب $((i, j, u), (\bar{R}_i, (R_{ju} - \bar{R}_u)\, S_{ij}))$

کاهش ۴

- ورودی: زوج مرتب $((i, j, u), (\bar{R}_i, (R_{ju} - \bar{R}_u)\, S_{ij}))$
- خروجی: $((i,u), P_{iu})$

به این ترتیب بر اساس معماری چهار لایه‌ی نگاشت‌کاهش ارایه شده، پالایش مشترک مبتنی بر آیتم در سرویس توصیه‌گر مولفه‌های درسی با قابلیت مقیاس‌پذیری بالایی در سیستم یادگیری الکترونیکی پیاده‌سازی می‌شود.

در این زیربخش (۵-۲)، به چگونگی به‌کارگیری فنون و ابزارهای دسته‌ی دوم - که به طور خاص برای اجرا در محیط‌های رایانشی بزرگ‌مقیاس طراحی شده‌اند - برای ارایه‌ی سرویس‌های توصیه‌گر در سیستم یادگیری الکترونیکی پرداختیم. به این منظور این دسته از سرویس‌ها را با در نظر گرفتن دو سناریوی مطرح (ارایه‌ی سرویس توصیه‌گر مقیاس‌پذیر ۱. مبتنی بر محتوا و ۲. مبتنی بر پالایش مشترک مولفه‌های درسی) طراحی نمودیم. صرف‌نظر از سناریوی انتخابی، طراحی سرویس‌ها بر اساس چارچوب رایانشی نگاشت‌کاهش انجام گرفت تا به این ترتیب با توزیع عملیات رایانشی در ارایه‌ی سرویس‌ها در محیط‌های رایانشی توزیع‌شده، از قابلیت مقیاس‌پذیری به طور کامل پشتیبانی شود. همان‌طور که در بخش چهارم این پژوهش مطرح و بررسی شد، پیاده‌سازی مدل‌های طراحی شده در این دسته، نیازمند دسترسی به خوشه‌های سخت‌افزاری است[۱۴۲] و با فراهم شدن این امکان دسترسی، مدل‌های ارایه شده در هر یک از سناریوهای مطرح شده آماده رسیدن به مرحله‌ی اجرا خواهند بود.

---

[۱۴۲] یادآوری می‌شود که توزیع عملیات رایانشی مورد نیاز در سرویس‌های طراحی شده، نیازمند دسترسی به خوشه‌های سخت‌افزاری است.

٦- جمع‌بندی و پژوهش‌های آتی

مدل‌سازی و شبیه‌سازی سیستم‌های رایانشی بزرگ مقیاس یکی از مراحل جاافتاده در حوزه‌ی تحقیق و توسعه‌ی کاربردها و الگوریتم‌ها به‌شمار می‌رود. با افزایش پیچیدگی چنین محیط‌هایی، فناوری‌های مربوط به شبیه‌سازی آن‌ها نیز تغییرات قابل ملاحظه‌ای را تجربه کرده است تا توان رویارویی با چالش‌هایی مانند چندبستری بودن محیط‌های توزیع‌یافته و یا چالش‌های دیگری از این دست را بیابد.

در این طرح پژوهشی، تاکید اصلی بر روی ابزارهای شبیه‌سازی است که به طور خاص در حوزه‌ی عملیات رایانشی بزرگ‌مقیاس به‌کار گرفته می‌شوند. با توجه به گستردگی تنوع این فنون و ابزارها، به منظور مطالعه و بررسی دقیق‌تر آن‌ها، ابتدا به ارایه‌ی یک طبقه‌بندی کلی در این حوزه پرداختیم و فنون و ابزارهای شبیه‌سازی در این حوزه‌ی کاربردی را بر اساس «نحوه‌ی به‌کارگیری» به دو دسته‌ی کلی تقسیم نمودیم، ۱) فنون و ابزارهایی که بر مبنای شبیه‌سازی محیط‌های رایانشی بزرگ مقیاس توسعه داده شده‌اند و ۲) فنون و ابزارهای شبیه‌سازی که بر روی بسترهای رایانشی بزرگ مقیاس قابل به‌کارگیری هستند (که ما در این پژوهش این دسته را فنون وابزارهای شبیه‌سازی بزرگ‌مقیاس - نامیدیم). در راستای همین طبقه‌بندی، ابتدا به بررسی و استخراج ویژگی‌های مشترک بین این دو دسته وهمچنین ویژگی‌های خاص فنون و ابزارهای مطرح در هر یک از این دسته‌ها پرداختیم. سپس نیازمندی‌های فنی مختلف (مانند نیازمندی‌های سخت‌افزاری و نرم‌افزاری) برای به‌کارگیری آن‌ها را به تفکیک هر دسته مورد بررسی قرار دادیم.

ما در این پژوهش تاکید اصلی خود را بر روی «آموزش و یادگیری الکترونیکی» به عنوان یکی از حوزه‌های خدمات رایانشی پژوهشگاه قرار داده‌ایم. از این رو در این گزارش به چگونگی به‌کارگیری فنون و ابزارهای هر یک از دو دسته‌ی اشاره شده در تحلیل و ارزیابی سناریوهای مختلف در حوزه‌ی خدماتی «آموزش و یادگیری الکترونیکی» پرداخته‌ایم. به این منظور، در راستای طبقه‌بندی ارایه شده، در بخشی از گزارش، سناریوهایی را تحلیل و ارزیابی نمودیم که به توسعه‌ی سیستم یادگیری و آموزش الکترونیکی در محیط‌های رایانشی بزرگ‌مقیاس می‌پردازند (پوشش دسته‌ی اول از فنون و ابزارها) و در بخش دیگری از گزارش نیز به طور خاص به چگونگی به‌کارگیری فنون و ابزارهای بزرگ‌مقیاس در ارایه‌ی سرویس‌های هوشمند یادگیری و آموزش الکترونیکی – که از جمله

سرویس‌های مهمی هستند که ارایه‌ی آن‌ها تنها با انجام عملیات رایانشی بزرگ‌مقیاس امکان‌پذیر است- پرداختیم (پوشش دسته‌ی دوم از فنون).

برای پژوهش‌های آتی که در ادامه‌ی این طرح پژوهشی انجام می‌گیرند، روی‌کردهای متفاوتی را می‌توان پیشنهاد داد. در روی‌کرد نخست می‌توان پژوهش‌های مشابهی را انجام داد که ساختاری مشابه به این طرح پژوهشی دارند، با این تفاوت که تاکید اصلی خود را بر روی دیگر حوزه‌های خدمات رایانشی پژوهشگاه قرار می‌دهند. به عنوان نمونه می‌توان در پژوهشی به تفصیل به چگونگی به‌کارگیری این فنون وابزارها در عملیات رایانشی بزرگ‌مقیاس در حوزه‌ی خدمات «عملیات محاسباتی پربازده» و یا «ارایه‌ی طرح‌ها و پایان‌نامه‌های الکترونیکی» پرداخت. روی‌کرد دیگری که در پژوهش‌های آتی می‌توان در پیش گرفت این است که در ادامه‌ی طرح کنونی و با تاکید بر روی حوزه‌ی «آموزش و یادگیری الکترونیکی» به طراحی و راه‌اندازی یک نمونه‌ی آزمایشگاهی پرداخت و سناریوهای مورد بررسی قرار گرفته در این طرح را در مقیاس محدود پیاده‌سازی و ارزیابی نمود. پژوهش‌هایی که چنین روی‌کردی داشته باشند در واقع گام بعدی در نقشه راهی هستند که در نهایت به ارایه‌ی عملی این حوزه‌ی خدماتی در مقیاس واقعی منتهی خواهد شد.

۷- مراجع

[1] Foster, C. Kesselman, and S. Tuecke, The anatomy of the grid: Enabling virtual organizations, International Journal of High Performance Computing Applications, Vol 15, Issue 3, pages 200-222, 2001.

[2] L.M. Vaquero, L. Rodero-Merino, J. Caceres, et al., A break in the clouds: Towards a cloud definition, ACM SIGCOMM Computer Communication Review, Vol. 39, Issue 1, pages 50-55, 2009.

[3] Florian Feldahaus, Stefan Freitag, Chaker EI Amrani, State-of-the-Art Technologies for Large-Scale Computing, Wiley Series on Parallel and Distributed Computing, (Editor) Albert Y. Zomaya, 2012.

[4] Atsuko Takefusa, Satoshi Matsuoka, Kento Aida, Hidemoto Nakada, and Umpei Nagashima, Overview of a performance evaluation system for global computing scheduling algorithms, In Proceedings of the The Eighth IEEE International Symposium on High Performance Dis- tributed Computing, page 11, Washington, DC, USA, 1999.

[5] H. Casanova. Simgrid: a toolkit for the simulation of ap- plication scheduling. In Proceedings of the IEEE Interna- tional Symposium on Cluster Computing and the Grid (CC- Grid'01),Brisbane, Australia, pages 430–437, 2001.

[6] Henri Casanova, Arnaud Legrand, and Loris Marchal, Scheduling distributed applications: the simgrid simulation framework. In Proceedings of the third IEEE Interna- tional Symposium on Cluster Computing and the Grid (CC- Grid'03), may 2003.

[7] Henri Casanova, Arnaud Legrand, and Martin Quinson, SimGrid: a Generic Framework for Large-Scale Distributed Experiments. In proceedings of the 10th IEEE International Conference on Computer Modeling and Simulation (UKSim), April 2008, Cambridge, UK.

[8] http://simgrid.gforge.inria.fr, accessed Aug 2013

[9] Rajkumar Buyya and Manzur Murshed, Gridsim: A toolkit for the modeling and simulation of distributed resource man- agement and scheduling for grid computing. The Journal of Concurrency and Computation: Practice and Experience (CCPE), 14(13-15), 2002.

[10] C. Dumitrescu and I. Foster, Gangsim: A simulator for grid scheduling studies, In Proceedings of the IEEE Interna- tional Symposium on Cluster Computing and the Grid (CC- Grid'05), Cardiff, UK, may 2005.

[11] William H. Bell, David G. Camerona, Luigi Capozza, A. Paul Millar, Kurt Stockinger, and Floriano Zini. OptorSim - a grid simulator for studying dynamic data replication strategies, International Journal of High Performance Com- puting Applications,, 17(4), 2003.

[12] EU DataGrid Project. The DataGrid Architecture. Technical Report DataGrid-12-D12.4-333671-3-0, CERN, Geneva, Switzerland, 2001.

[13] Simon Coakley, Marian Gheorghe, Mike Holcombe, Shawn Chin, David Worth, Chris Greenough, Exploitation of High Performance Computing in the FLAME Agent-Based Simulation Framework, IEEE 14th International Conference on High Performance Computing and Communications, 2012.

[14] Matthias Scheutz and Jack J. Harris, An Overview of the SimWorld Agent-based Grid Experimentation System, Wiley Series on Parallel and Distributed Computing, (Editor) Albert Y. Zomaya, 2012.

[15] Nick Collier, Repast HPC Manual, February 6, 2012, Available at http://repast.sourceforge.net/docs.html

[16] Collier, Nicholson, and Michael North, Repast HPC: A Platform for Large-Scale Agent-Based Modeling, Wiley Series on Parallel and Distributed Computing, (Editor) Albert Y. Zomaya, 2012.

[17] J. T. Murphy, "Computational social science and high performance computing: A case study of a simple model at large scales," in Proceedings of the 2011 Computational Social Science Society of America Annual Conference, Santa Fe, 2011.

[18] Ciprian Dobre, Florin Pop, Valentin Cristea, New Trends in Large Scale Distributed Systems Simulation, In proceedings of International Conference on Parallel Processing Workshops, 2009.

[19] Benjamin Quetier and Franck Cappello, A survey of Grid research tools: simulators, emulators and real life platforms, In Proceedings of the 17th IMACS World Congress, 2005.

[20] Saurabh Kumar Garg and Rajkumar Buyya, NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations, Proceedings of the 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011, IEEE CS Press, USA), Melbourne, Australia, December 5-7, 2011.

[21] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50,

ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011. (Preferred reference for CloudSim)

[22] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros, Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities, Proceedings of the 7th High Performance Computing and Simulation Conference (HPCS 2009, ISBN: 978-1-4244-4907-1, IEEE Press, New York, USA), Leipzig, Germany, June 21-24, 2009.

[23] Kliazovich D, Bouvry P, Khan SU, GreenCloud: a packet-level simulator of energy- aware cloud computing data centers. The Journal of Supercomputing. pp 1–21, 2010.

[24] The Network Simulator ns-2. Available at: http://www.isi.edu/nsnam/ns/, Accessed Aug 2013

[25] Fan X, Weber W-D, Barroso LA (2007) Power provisioning for a warehouse-sized computer. In: ACM international symposium on computer architecture, San Diego, CA, June 2007

[26] Chen G, He W, Liu J, Nath S, Rigas L, Xiao L, Zhao F (2008) Energy-aware server provisioning and load dispatching for connection-intensive internet services. In: The 5th USENIX symposium on networked systems design and implementation, Berkeley, CA, USA

[27] Shang L, Peh L-S, Jha NK (2003) Dynamic voltage scaling with links for power optimization of interconnection networks. In: Proceedings of the 9th international symposium on high- performance computer architecture (HPCA '03). IEEE Computer Society, Washington, DC, USA, pp 91–102.

[28] Bart Jacob, Michael Brown, Kentaro Fukui, and Nihar Trivedi, "Introduction to Grid Computing", IBM International Technical Support Organization, 2005.

[29] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I.Stoica, M.Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report No.UCB/EECS-2009-28, University of California at Berkeley, 2009.

[30] Sulistio A. Yeo C. S, Buyya R., "A taxonomy of computer-based simulations and its mapping to parallel and distributed systems", software-practice and experience, vol. 34:653-673, 2004.

[31] Ciprian Dobre, Florin Pop, Valentin Cristea, "New Trends in Large Scale Distributed Systems Simulation", Journal of Algorithms & Computational Technology, Vol. 5, No. 2, 2011.

[32] Santiago B. Robinson, "A Modeling Process to Understand Complex System Architectures", PhD Thesis, Chap II, Georgia Institute of Technology August, 2009.

[33] Michael A. Murphy, "Virtual Organization Clusters: Self-Provisioned Clouds on the Grid", PhD Thesis, Chap 2, Clemson University, 2010.

[34] Gilberto Cunha Filho, Francisco Jos´e da Silva, "OGST: an Opportunistic Grid Simulation Tool", Latin American Grid (LAGrid) workshop, 2008.

[35] Jens Gustedt, Emmanuel Jeannot, Martin Quinson, "Experimental Methodologies for Large-Scale Systems: a Survey", Parallel Processing Letters, Vol. 23, No. 32, November 20, 2010.

[36] James William Smith, Ali Khajeh-Hosseini, Jonathan Stuart Ward, Ian Sommerville, "CloudMonitor: Profiling Power Usage", in IEEE Cloud 2012 Work In Progress Track, https://github.com/jws7/CloudMonitor

[37] John Cartlidge and Dave Cliff, "omparison of Cloud Middleware Protocols and Subscription Network Topologies using CReST, the Cloud Research Simulation Toolkit", In Proc. of 3rd International Conference on Cloud Computing, 2013 http://sourceforge.net/projects/cloudresearch/

[38] GNU Project. Free Software Foundation:"What is Free Software" https://www.gnu.org/philosophy/free-sw.html

[39] Open-Source Foundation, http://opensource.org/docs/osd

[40] J. Dean and S. Ghemawat, "Mapreduce:Simplified data processing on large clusters," Procedings of OSDI 2004: Sixth Sixth Symposium on Operating System Design and Implementation, 2004.

[41] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Commun. ACM, vol. 51, pp. 107- 113, 2008.

[42] A. Fern´andez, D. Peralta, F. Herrera, and J.M. Ben´ıtez, "An Overview of E-Learning in Cloud Computing," Workshop on Learning Technology for Education in Cloud, Advances in Intelligent Systems and Computing, vol. 173, 2012, pp 35-46.

[43] T. Dillon, C. Wu and E. Chang, "Cloud Computing: Issues and Challenges", 24th IEEE International Conference on Advanced Information Networking and Appications, 2010.

[44] Kwan, R., Fox, R., Chan, F., Tsang, P.: Enhancing Learning Through Technology: Research on Emerging Technologies and Pedagogies. World Scientific, 2008.

[45] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition", ACM SIGCOMM Computer Communication Review, Volume 39 Issue 1, pages 50-55, January 2009.

[46] A.E. Youssef , "Exploring Cloud Computing Services and Applications," Journal of Emerging Trends in Computing and Information Sciences, vol. 3, no. 6, July 2012.

[47] M. Nasr, S. Ouf, "An Ecosystem in e-Learning Using Cloud Computing as platform and Web2.0," The Research Bulletin of Jordan ACM, vol.II(IV).

[48] Vouk, M., Averitt, S., Bugaev, M., Kurth, A., Peeler, A., Shaffer, H., Sills, E., Stein, S., Thompson, J., "Powered by VCL - using virtual computing laboratory (VCL) technology to power cloud computing'" In: 2nd Intl. Conference on the Virtual Computing Initiative (ICVCI), Research Triangle Park, North Carolina, USA, 2008

[49] Dong, B., Zheng, Q., Qiao, M., Shu, J., Yang, J.: BlueSky Cloud Framework: An ELearning Framework Embracing Cloud Computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp. 577–582. Springer, Heidelberg, 2009.

[50] Sulistio, A., Reich, C., Doelitzscher, F.: Cloud Infrastructure & Applications – CloudIA. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp. 583–588. Springer, Heidelberg, 2009.

[51] M. H. Masud, X. Huang, An E-learning System Architecture based on Cloud Computing, World Academy of Science, Engineering and Technology, Issue 62: 2012.

[52] A. M. Gamundani, T. Rupere, "A cloud computing architecture for e-learning platform, supporting multimedia content", International Journal of Computer Science and Information Security, Vol. 11, No.3, March 2013.

[53] P. Pocatilu, F. Alecu, M. Vetrici, "Using Cloud Computing for E-learning Systems", Recent advances on data networks, communications, computers: proceedings of the 8th WSEAS International Conference on Data Networks, Communications, Computers, Italy, 2009.

[54] A. Jain, S. Chawla, "E-Learning in the Cloud", International Journal of Latest Research in Science and Technology, Vol. 2, Issue 1, pages 478-481, 2013.

[55] G. Vakili, "Designing an E-learning System based on Cloud Computing Models" (in Farsi), In Proc. of 7th National and 4th International Conference on E-Learning, Shiraz, Iran, 2013.

[56] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and

Opportunities", In Proc. of International Conference on High Performance Computing Simulation, HPCS '09, pp. 1-11, 2009.

[57] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software: Practice and Experience, Vol. 41, Issue 1, pages 23–50, 2011.

[58] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I.Stoica, M.Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report No.UCB/EECS-2009-28, University of California at Berkeley, 2009.

[59] Slavomir Stankov, Vlado Glavinic, Marko Rosic, "Intelligent Tutoring Systems in E-Learning Environments," IGI Global, July 31, 2010.

[60] Gang Zhou, Jason T. L. Wang, Meter A. Ng. Curriculum knowledge representation and manipulation in Knowledge based tutoring systems. IEEE Transactions on knowledge and data engineering, vol. 8, no 5, 1996.

[61] Salton, Wong and Yang, "A Vector Space Model for Automatic Indexing", CACM (18)11.

[62] M. Bhandarkar, "MapReduce programming with apache Hadoop," in Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on, 2010, pp. 1-1.

[63] Z. D. Zhao and M. S. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," in International Workshop on Knowledge Discovery and Data Mining, 2010, pp. 478-481.

[64] T. HengSong and Y. HongWu, "A Collaborative Filtering Recommendation Algorithm Based on Item Classification," in Circuits, Communications and Systems. PACCS '09. Pacific-Asia Conference on, 2009, pp. 694-697.

[65] G. Song-Jie, Y. HongWu, and T. HengSong, "Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System," in Circuits, Communications and Systems. PACCS '09. Pacific-Asia Conference on, 2009, pp. 690-693.

[66] S. Xiaoyuan, T. M. Khoshgoftaar, and R. Greiner, "Imputed Neighborhood Based Collaborative Filtering," in Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on, 2008, pp. 633-639.

[67] S. Ping and Y. HongWu, "An Item Based Collaborative Filtering Recommendation Algorithm Using Rough Set Prediction," in Artificial Intelligence, 2009. JCAI '09. International Joint Conference on, 2009, pp. 308-311.

[68] Jing Jiang, Jie Lu, Guangquan Zhang, Guodong Long, "Scaling-up Item-based Collaborative Filtering Recommendation Algorithm based on Hadoop," 2011 IEEE World Congress on Services, 2011.

[69] Z. D. Zhao and M. S. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," in International Workshop on Knowledge Discovery and Data Mining, 2010, pp. 478-481.

[70] Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web, Hong Kong, Hong Kong, 2001, pp. 285-295.

[71] W. Fajardo Contreras1, E. Gibaja Galindo, E. Marín Caballero1 and G. Marín Caballero, "An Intelligent Tutoring System for a Virtual E-learning Center," Current Developments in Technology-Assisted Education, 2006.

[72] I-Han Hsiao and Peter Brusilovsky, "The role of community feedback in the student example authoring process: An evaluation of AnnotEx," British Journal of Educational Technology, Vol. 42, No, 3, 482–499, 2011.

[73] John Champaign and Robin Cohen, "A Model for Content Sequencing in Intelligent Tutoring Systems Based on the Ecological Approach and Its Validation Through Simulated Students," Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS), 2010.

[74] Niels Pinkwart, Vincent Aleven, Kevin Ashley and Collin Lynch, "Using Collaborative Filtering in an Intelligent Tutoring System for Legal Argumentation," In Proc. of workshops at 4th Int. Conf. on Adaptive Hypermedia and Adaptive Web based Systems, 2006.

واژه‌نامه انگلیسی به فارسی

agent	عامل
Amazon Web Services (AWS)	سرویس های وب آمازون
analyzer	تحلیل گر
Application Programming Interface (API)	واسط برنامه نویسی کاربرد
broker	دلال
chat	گپ
Collaborative filtering	پالایش مشترک
Computationally intensive	با حجم پردازشی بالا
Concurrency	هم زمان سازی
Concurrent Sequential Process (CSP)	فرآیندهای متوالی هم زمان
Content-based	مبتنی بر محتوا
context	زمینه
Correlation	وابستگی
coupled	به هم پیوند خورده
Data intensive	با حجم داده ی بالا
Deferred Queue	صف تاخیر
Development interface	واسط های توسعه
documents	سندها
DVFS (Dynamic Voltage/Frequency Scaling)	مقیاس دهی پویای ولتاژ/ فرکانس
Elastic Compute Cloud (EC2)	ابر محاسباتی کشسان
Entity-based	مبتنی بر موجودیت
Event-based	مبتنی بر رویداد

flops (floating point operations per second)	عملیات نقطه شناور بر ثانیه
forum	انجمن
Future Queue	صف آینده
Hadoop Distributed File System (HDFS)	فایل سیستم توزیع شده ی هدوپ
High Performance Computing	محاسبات پربازده
hybrid cloud	ابر هایبرید
Infrastructure as a Service (IaaS)	بستر به عنوان سرویس
instance	نمونه
Instant messaging	ارسال بلادرنگ پیغام
Intelligent Tutoring System	سیستم های هوشمند آموزش
Job tracker	ردگیر کار
Learner-centered	یادگیرنده محور
MapReduce	نگاشت کاهش
Master	گره اصلی
migration	مهاجرت
Model discovery	اکتشاف مدل
MOOC (Massive Online Open Course)	درس های برخط باز و فراگیر
MPI (Message Passing Interface)	واسط تبادل پیغام
multiplatform	چندبستری
open-source	متن باز
passive	کنش پذیر
Platform as a Service (PaaS)	بستر به عنوان سرویس
portability	انتقال پذیری
Power saving	حفظ توان
presentations	ارایه ها
private cloud	ابر خصوصی
Proactive	کنش گر

Profile	نمایه
projection	نقشه
public cloud	ابر عمومی
Recommender	توصیه گر
registration	ثبت
Replication	تکرار
repository	مخزن
Rule-based	مبتنی بر قانون
Scalability	مقیاس پذیری
scalable	مقیاس پذیر
Shuffling	جابه جایی
Simple Storage Service (S3)	سرویس ذخیره‌سازی ساده
Single core	تک هسته
SLA (service level agreement)	توافق نامه ی کیفیت سرویس
Slave	گره کارگر
Smart agents	عامل های هوشمند
Software as a Service (SaaS)	نرم افزار به عنوان سرویس
Sorting	مرتب سازی
Space-shared	اشتراک زمانی
Stand-alone	مستقل
state machine	ماشین حالت
state transition function	توابع گذر از حالت
Statistics-based	مبتنی بر آمار
tag	برچسب
task	کار
task graph	گراف کار

Term Frequency Inverse Document Frequency	فرکانس واژه عکس فرکانس آیتم
thread	رشته
Time zone	زمان منطقه ی جغرافیایی
Time-shared	اشتراک زمانی
transparent	شفاف
Utilization model	مدل بهره وری
Value-added services	سرویس های ارزش افزوده
Virtual Organization (VO)	سازمان های مجازی
visualization	تجسم
visualizer	نمایان گر
Development interface	واسط توسعه
Elastic Compute Cloud (EC2)	ابر محاسباتی کشسان
Recommender	توصیه گر
Slave	گره کارگر
Software as a Service (SaaS)	نرم افزار به عنوان سرویس
agent	عامل
Amazon Web Services (AWS)	سرویس های وب آمازون
analyzer	تحلیل گر
Application Programming Interface (API)	واسط برنامه نویسی کاربرد
broker	دلال
chat	گپ
Collaborative filtering	پالایش مشترک
Computationally intensive	با حجم پردازشی بالا
Concurrency	هم زمان سازی
Concurrent Sequential Process (CSP)	فرآیندهای متوالی هم زمان
Content-based	مبتنی بر محتوا

context	زمینه
Correlation	وابستگی
coupled	به هم پیوند خورده
Data intensive	با حجم داده ی بالا
Deferred Queue	صف تاخیر
documents	سندها
DVFS (Dynamic Voltage/Frequency Scaling)	مقیاس دهی پویای ولتاژ / فرکانس
Entity-based	مبتنی بر موجودیت
Event-based	مبتنی بر رویداد
flops (floating point operations per second)	عملیات نقطه شناور بر ثانیه
forum	انجمن
Future Queue	صف آینده
Hadoop Distributed File System (HDFS)	فایل سیستم توزیع شده هدوپ
High Performance Computing	محاسبات پربازده
hybrid cloud	ابر هایبرید
Infrastructure as a Service (IaaS)	بستر به عنوان سرویس
instance	نمونه
Instant messaging	ارسال بلادرنگ پیغام
Intelligent Tutoring System	سیستم های هوشمند آموزش
Job tracker	ردگیر کار
Learner-centered	یادگیرنده محور
MapReduce	نگاشت کاهش
Master	گره اصلی
migration	مهاجرت
Model discovery	اکتشاف مدل
MOOC (Massive Online Open Course)	درس های برخط باز و فراگیر
MPI (Message Passing Interface)	واسط تبادل پیغام

چندبستری	*multiplatform*
متن باز	*open-source*
کنش پذیر	*passive*
بستر به عنوان سرویس	*Platform as a Service (PaaS)*
انتقال پذیری	*portability*
حفظ توان	*Power saving*
ارایه ها	*presentations*
ابر خصوصی	*private cloud*
کنش گر	*Proactive*
نمایه	*Profile*
نقشه	*projection*
ابر عمومی	*public cloud*
ثبت	*registration*
تکرار	*Replication*
مخزن	*repository*
مبتنی بر قانون	*Rule-based*
مقیاس پذیری	*Scalability*
مقیاس پذیر	*scalable*
جابه جایی	*Shuffling*
سرویس ذخیره‌سازی ساده	*Simple Storage Service (S3)*
تک هسته	*Single core*
توافق نامه ی کیفیت سرویس	*SLA (service level agreement)*
عامل های هوشمند	*Smart agents*
مرتب سازی	*Sorting*
اشتراک زمانی	*Space-shared*
مستقل	*Stand-alone*
ماشین حالت	*state machine*

state transition function	توابع گذر از حالت
Statistics-based	مبتنی بر آمار
tag	برچسب
task	کار
task graph	گراف کار
Term Frequency Inverse Document Frequency	فرکانس واژه عکس فرکانس آیتم
thread	رشته
Time zone	زمان منطقه ی جغرافیایی
Time-shared	اشتراک زمانی
transparent	شفاف
Utilization model	مدل بهره وری
Value-added services	سرویس های ارزش افزوده
Virtual Organization (VO)	سازمان های مجازی
visualization	تجسم
visualizer	نمایان گر

واژه‌نامه فارسی به انگلیسی

ابر خصوصی	*private cloud*
ابر عمومی	*public cloud*
ابر محاسباتی کشسان	*Elastic Compute Cloud (EC2)*
ابر هایبرید	*hybrid cloud*
ارایه ها	*presentations*
ارسال بلادرنگ پیغام	*Instant messaging*
اشتراک زمانی	*Space-shared*
اشتراک زمانی	*Time-shared*
اکتشاف مدل	*Model discovery*
انتقال پذیری	*portability*
انجمن	*forum*
با حجم پردازشی بالا	*Computationally intensive*
با حجم داده ی بالا	*Data intensive*
برچسب	*tag*
بستر به عنوان سرویس	*Infrastructure as a Service (IaaS)*
بستر به عنوان سرویس	*Platform as a Service (PaaS)*
به هم پیوند خورده	*coupled*
پالایش مشترک	*Collaborative filtering*
تجسم	*visualization*
تحلیل گر	*analyzer*
تک هسته	*Single core*
تکرار	*Replication*
توابع گذر از حالت	*state transition function*

توافق نامه ی کیفیت سرویس	*SLA (service level agreement)*
توصیه گر	*Recommender*
ثبت	*registration*
جابه جایی	*Shuffling*
چندبستری	*multiplatform*
حفظ توان	*Power saving*
درس های برخط باز و فراگیر	*MOOC (Massive Online Open Course)*
دلال	*broker*
ردگیر کار	*Job tracker*
رشته	*thread*
زمان منطقه ی جغرافیایی	*Time zone*
زمینه	*context*
سازمان های مجازی	*Virtual Organization (VO)*
سرویس ذخیره‌سازی ساده	*Simple Storage Service (S3)*
سرویس های ارزش افزوده	*Value-added services*
سرویس های وب آمازون	*Amazon Web Services (AWS)*
سندها	*documents*
سیستم های هوشمند آموزش	*Intelligent Tutoring System*
شفاف	*transparent*
صف آینده	*Future Queue*
صف تاخیر	*Deferred Queue*
عامل	*agent*
عامل های هوشمند	*Smart agents*
عملیات نقطه شناور بر ثانیه	*flops (floating point operations per second)*
فایل سیستم توزیع شده ی هدوپ	*Hadoop Distributed File System (HDFS)*
فرآیندهای متوالی هم زمان	*Concurrent Sequential Process (CSP)*

فرکانس واژه عکس فرکانس آیتم	*Term Frequency Inverse Document Frequency*
کار	*task*
کنش پذیر	*passive*
کنش گر	*Proactive*
گپ	*chat*
گراف کار	*task graph*
گره اصلی	*Master*
گره کارگر	*Slave*
ماشین حالت	*state machine*
مبتنی بر آمار	*Statistics-based*
مبتنی بر رویداد	*Event-based*
مبتنی بر قانون	*Rule-based*
مبتنی بر محتوا	*Content-based*
مبتنی بر موجودیت	*Entity-based*
متن باز	*open-source*
محاسبات پربازده	*High Performance Computing*
مخزن	*repository*
مدل بهره وری	*Utilization model*
مرتب سازی	*Sorting*
مستقل	*Stand-alone*
مقیاس پذیر	*scalable*
مقیاس پذیری	*Scalability*
مقیاس دهی پویای ولتاژ/ فرکانس	*DVFS (Dynamic Voltage/Frequency Scaling)*
مهاجرت	*migration*
نرم افزار به عنوان سرویس	*Software as a Service (SaaS)*

نقشه	*projection*
نگاشت کاهش	*MapReduce*
نمایان گر	*visualizer*
نمایه	*Profile*
نمونه	*instance*
هم زمان سازی	*Concurrency*
وابستگی	*Correlation*
واسط برنامه نویسی کاربرد	*Application Programming Interface (API)*
واسط تبادل پیغام	*MPI (Message Passing Interface)*
واسط های توسعه	*Development interface*
یادگیرنده محور	*Learner-centered*

Abstract

According to wide-spread use of large-scale computing, most of the computing services potentially provided by IRANDOC[*] in a near future, such as Electronic Theses and Dissertation, eLearning, and High Performance Computing, should adopt these technologies in order to satisfy the stakeholders' requirements. Considering the raised issue, the main concern in our current research is investigating how to employ large-scale simulation techniques and tools as the first step in a roadmap for developing large-scale computing services based on the state-of-the art technologies. To this end, we explore different techniques and tools in the field of large-scale computing and categorize them to study their features and requirements in more detail. After identifying the requirements of each category (e.g. in software and hardware aspects), we choose eLearning services specifically to focus our research. We scrutinize the employment of the tools and techniques in each category separately and evaluate some common scenarios of providing eLearning services based on large-scale computing technologies.

[*] *Iranian Research Institute for Information Science and Technology*

Ministry of Science, Research and Technology
Iranian Research Institute for Information Science and Technology

Research Report

Applying Simulation Techniques and Tools for Large-scale Computing

Golnaz Vakili

Spring 1393