دوره نهم، شماره دوم، پاییز و زمستان ۱۳۹۵

مجله فناوری اطلاعات در طراحی مهندسی
Information Technology in Engineering Design
http://ited.sinaweb.net

یک الگوریتم کارامد برای زمانبندی کارها در محیط محاسبات ابری

سکینه رضایی‌زاده [(۱)] داوود بهره‌پور *[(۲)]

(۱) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

(۲) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران*

چکیده

محاسبات ابری یکی از پدیده نوظهور در محیط محاسبات توزیع شده است که با هدف ارائه سرویس‌ها به کاربران به صورت مجازی و بر اساس نیاز آنها به وجود آمده است. ابر روز به روز در حال توسعه است و از طرفی با چالش‌های زیادی رو به رو است. یکی از این چالش‌ها زمانبندی کارها است که خود یک مساله پیچیده در محیط ابر می‌باشد. برخی ازالگوریتم های زمانبندی از مرتب سازی پروسه‌ها برای زمانبندی آن‌ها استفاده می‌کنند. یکی از سریع‌ترین الگوریتم‌های مرتب سازی در محیط‌های موازی، الگوریتم مرتب سازی بایتونیک است کهبر خلاف بسیاری از الگوریتم‌های مرتب سازی دارای خروجی دو بعدی است. در این مقاله روشی مبتنی بر این الگوریتم با هدف ایجاد تعادل بار بر روی پردازنده‌ها ارائه می‌گردد که پروسه‌ها ابتدا وارد شبکه مرتب سازی بایتونیک می‌شوند و بعد از آن برای اجرا بر روی پردازنده‌ها ارسال می‌گردند. نتایج با استفاده شبیه ساز Cloudsim درسیستم عامل ویندوز نمایش داده شده است. نتایج شبیه سازی با الگوریتم زمانبندی STF مقایسه شده و بیانگر تعادل بار بر روی پردازنده‌ها است که خود منجر به کاهش زمان اتمام کار هر پردازنده و افزایش توان عملیاتی آنها گردیده است.

واژه‌های کلیدی: محاسبات ابری، ماشین مجازی، تعادل بار، زمانبندی پروسه‌ها، مرتب سازی بایتونیک

* عهده‌دارمکاتبات

نشانی: گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

تلفن: ۰۹۱۵۳۱٤٥۲۰۳ **پست الکترونیکی:** bahrepour@mshdiau.ac.ir

۱- مقدمه

محاسبات در حال تبدیل شدن به مدلی از سرویس‌ها است که به روشی مشابه سایر سرویس‌ها مدیریت (مانند آب، برق، گاز و تلفن) و به کاربران تحویل داده شود. در این مدل کاربران بر حسب نیاز به سرویس‌ها دست یابی پیدا می‌کنند، بدون اینکه بدانند این سرویس‌ها در کجا قرار گرفته و یا چگونه ارائه می‌شوند. در جهت ارائه این سرویس سودمند چندین مدل محاسباتی مانند محاسبات کلاستر ، محاسبات گرید و اخیراً محاسبات ابری معرفی شده است.

سیستم محاسبات ابری، زیر ساختی همانند یک ابر را نشان می‌دهد که کاربران در زمینه‌های مختلف تجاری، اقتصادی و... می‌توانند در هر زمان و هر جایی از دنیا به داده‌ها و کاربردهای مورد نیاز دسترسی داشته باشند. این زیر ساخت شامل مراکز داده‌ای است که در سرتاسر دنیا توزیع شده و از طریق اینترنت قابل دسترسی هستند. محاسبات ابرییک نوع سیستم توزیع شده با مقیاس بزرگ است که شامل تعداد زیادی منابع فیزیکی و مجازی است و بر اساس تقاضا این منابع را به صورت پویا و بر حسب یک سطح سرویس موافقت شده به مصرف کنندگان ارائه می‌دهد [۱]. هدف سیستم‌های محاسبات ابری به حداقل رساندن هزینه استفاده از منابع توسط تهیه کنندگان سرویس و نیز به حداکثر رساندن درآمد حاصل از سرویس دادن به برنامه‌های کاربردی مصرف کنندگان می‌باشد [۲]. در سیستم محاسبات ابری از زمانبندی، به منظور کاهش زمان انتظار درخواست‌ها و افزایش بهره وری از منابع استفاده می‌شود. زمانبندی وظایف یک فرآیند کلیدی در لایه زیرساخت ابر است که فرآیند نگاشت کارها به منابع در دسترس، بر پایه نیازمندی‌ها و ویژگی کارهاست و هدف آن، اجرای درخواست‌های وارد شده به سیستم بر روی منابع، به شیوه‌ای کارآمد و با درنظر گرفتن سایر خصوصیات محیط ابری می‌باشد. در محیط محاسبات ابری هر کاربر ممکن است برای اجرای هر کار با صدها منابع مجازی روبرو شود. در این صورت، تخصیص کارها به منابع مجازی توسط خود کاربر غیر ممکن می‌باشد. به دلیل خصوصیات پویای محاسبات ابری و نیازهای متفاوت برنامه‌های کاربردی مختلف از منابع، بحث زمانبندی وظایف به عنوان یک مسئله NP به شمارمی آید [۳]. هدف از زمانبندی کارها، تخصیص پردازنده‌ها به پروسه‌ها در طول زمان است به گونه‌ای که هدف‌های سیستم برآورده شود. یا در تعریفی دیگر، زمانبندی تصمیم گیری برای چگونگی اختصاص پروسه‌ها به پردازنده‌ها است. زمانبندی با اجرا متفاوت است. زمانبندی کارها، تخصیص منابع به درخواست‌ها است به گونه‌ای که بهره وری سیستم و سطح رضایتمندی کاربران افزایش یابد. الگوریتم‌های زمانبندی مختلفی در سیستم‌های ابری پیشنهاد شده است که فاکتورهایی مانند مانند کاهش انرژی، تعادل بار، کاهش زمان انتظارکارها و بهره‌وری عادلانه از منابع را مورد بررسی قرار داده‌اند [٤].

یکی از مهم‌ترین چالش‌ها در زمینه زمانبندی تخصیص منابع به پروسه‌ها، عدم تعادل بار بر روی پردازنده‌ها است. در این پژوهش به بررسی این چالش پرداخته و رویکردی برای متعادل کردن بار پردازنده‌ها ارائه می‌شود، به این صورت که با استفاده از یک الگوریتم مرتب سازی به نام بایتونیک ابتدا پروسه‌ها را مرتب نموده و سپس به پردازنده‌ها ارسال می‌شوند، به طوری که تا حد امکان پروسه‌ها به موقع از پردازنده‌ها استفاده کنند و بار پردازنده‌ها متعادل باشد و پردازش پروسه در یک زمان مقبول به پایان برسد. بر خلاف بسیاری از الگوریتم‌های زمانبندی که پروسه‌ها را در یک سیر صعودی یا نزولی مرتب می‌کنند در این رویکرد پروسه‌ها در یک ترکیب صعودی- نزولی و برعکس مرتب می‌شوند.

در این مقاله، دربخش ۲ مروری بر ادبیات و کارهای مرتبط ارائه می‌شود. در بخش ۳ الگوریتم مرتب سازی بایتونیک، در بخش ٤ ساختار و تحلیل روش پیشنهادی و در بخش ۵ ارزیابی نتایج و تجزیه تحلیل روش پیشنهادی با استفاده از نرم افزار Cloudsim بیان می‌گردد.

۲- مرور ادبیات

۲-۱- محاسبات ابری

محاسبات ابری یکی از پدیده‌های نوظهور در دنیای کامپیوتر و ارتباطات است که می‌توان گفت از ترکیب علومی مانند سیستم‌های توزیع شده Grid، محاسبات عمومی و مجازی سازی مشتق شده است. سیر تکاملی محاسبات به گونه‌ای است که می‌توان آن را پس از آب، برق، گاز و تلفن به عنوان عنصر اساسی پنجم فرض نمود. هدف این سیستم‌ها ارائه خدمات به کاربران در بستر اینترنت است. سیستم ابر سه نوع خدمات ارائه می‌کند [۵]: زیر ساخت به عنوان سرویس[1] (IaaS): در این سرویس معمولاً یک کامپیوتر مجازی به کاربر ارائه می‌شود که به صورت کامل آن را کنترل می‌کند. پلت فرم به عنوان سرویس[2](PaaS): در این نوع از سرویس، سکوهای محاسباتی برای اجرای کاربردها فراهم می‌شوند. نرم افزار به عنوان سرویس[3] (SaaS) : این امکان را فراهم می‌کند که نرم افزارها به عنوان سرویسی بر حسب نیاز توسط مشتریان ارائه شوند. مرکز داده ابر، که محل ارائه خدمات است به دو دسته همگن و ناهمگن تقسیم می‌شود که در حالت همگن توان عملیاتی پردازنده‌ها با هم برابر است درحالیکه در حالت ناهمگن پردازنده‌ها از نظر توان عملیاتی با یکدیگر برابر نیستند [۵].

۲-۲- زمانبندی

زمان‌بندی در سیستم‌های توزیع شده یک مسئله NP-Complete می‌باشد[۵]. این مسئله نگاشت یک گراف با مجموعه‌ای از پروسه‌های محاسباتی و همچنین تعیین ترتیب حرکت داده‌ها بین آن‌ها، در محیطی با پردازنده‌های موازی می‌باشد. هدف الگوریتم زمانبند پروسه، تخصیص پروسه‌ها به پردازنده‌های در دسترس است به طوری که محدودیت تقدم و تأخر پروسه‌ها برآورده شود و همزمان طول اجرای پروسه‌ها یا کل زمان اتمام[4](TFT) پردازش کمتر شود [٦]. الگوریتم‌های زمانبندی بر مبنای پارامترهای مختلف و به روشهای مختلف به منظور برآورده شدن اهدافی همچون افزایش بهره وری از منابع، کاهش زمان انتظار پروسه‌ها و افزایش پارامترهای کیفیت سرویس تاکنون ارائه شده است.

الگوریتم‌های زمانبندی در حالت کلی به سه دسته زیر با توجه به معیارهای مختلف طبقه بندی شده‌اند [۷].

انحصاری و غیر انحصاری: در حالت انحصاری هنگامی که پردازنده در اختیاریک پروسه قرار گرفت، آنقدر در اختیارش باقی می‌ماند تا پروسه به اتمام برسد یا مسدود شود. در حالت غیر انحصاری هنگامی که پردازنده را به یک پروسه واگذار کرد، می‌تواند بر طبق الگوریتم زمانبندی بر خلاف میل پروسه، پردازنده را از پروسه باز پس گیرد.

زمانبندی ایستا و زمانبندی پویا: در ایستا زمان ورود کارها به سیستم و مدت زمان انجام آن‌ها قبل از شروع برای الگوریتم زمانبندی مشخص است و اگر کار جدیدی وارد سیستم شود تا پایان اجرای کلیه کارهای قبلی در زمانبندی دخالت داده نمی‌شود. در پویا ممکن است مشخصات کلیدی کارها از قبیل مدت زمان انجام آن‌ها در زمان اجرای برنامه تغییر یابد.اگر کار جدیدی وارد سیستم شود در هنگام اجرای کارهای قبلی ممکن است در زمانبندی دخالت داده شود. زمانبندی ایستا از نظر پیاده سازی آسان است در حالیکه زمانبندی پویا برای سناریوهای محیط‌های واقعی مناسب است. همچنین زمانبندی پویا هزینه‌های زمان اجرا را به حداقل می‌رساند [7].

در ادامه برخی از مقالات مرتبط در زمینه زمانبندی پروسه‌ها معرفی می‌گردد. در [۸] به منظور طراحی، پیاده سازی و ارزیابی یک الگوریتم زمانبند سبز به انضمام یک شبکه‌ی عصبی پیشگو برای بهینه سازی مصرف قدرت سرور در محاسبات ابری است که یک پیشگو برای پیشگویی تقاضای فراخوانی بعدی مبنی بر تقاضای تاریخی به کار می‌گیرد. بر طبق پیشگویی، الگوریتم سرورهای

[1] Infrastructure as a Service
[2] Platform as a Service
[3] Software as a Service

[4] Total Finish Time

بیکار وغیر استفاده را خاموش می‌کند و آن‌ها را برای کم کردن تعداد سرورهای درحال اجرا راه اندازی می‌کند. در [۹] یک مجموعه از کارهای مستقل بر روی یک مجموعه محدود از پردازنده‌ها که کاملاً متصل نیستند توزیع می‌شوند. اجرای کارها روی پردازنده‌های مختلف هزینه‌های متفاوت دارد و هر یک از لینک‌های بین پردازنده‌ها دارای نرخ شکست و پهنای باند متفاوت هستند. در این کار سعی شده رویکردی ارائه شود که قابلیت اطمینان اجرای کارها تا حد امکان بیشینه شده و کارها در یک زمان مقبول به پایان برسند. در[۱۰] در یک سیستم ناهمگن پردازنده‌ها سعی در نگاشت مجموعه‌ای از کارهای مستقل بر روی تعدادی پردازنده‌ی ناهمگن دارد به طوری که قابلیت اطمینان در طول اجرای کاربرد بیشینه شود. در [۱۵-۱۱] الگوریتم زمانبندی STF[5] مطرح شده است. در این الگوریتم پروسه‌ها بر اساس زمان اجرا به صورت صعودی مرتب می‌شوند، کوچک‌ترین کار در اولین پردازنده آزاد قرار می‌گیرد و کار بعدی در پردازنده آزاد بعدی الی آخر تا تمام کارها زمانبندی شوند. همچنین در این الگوریتم معیارهایی ارزیابی شده است که با الگوریتم پیشنهادی در بخش شبیه سازی مقایسه خواهیم نمود. در [16] به منظور توازن بار در محیط ابر الگوریتمی جدید بنام TBSLB-PSO[6] ارائه شد که از تکنیک مهاجرت کارها بر روی ماشین‌های مجازی بیکار استفاده کرد و شامل دو مرحله بود: PSO که در آن انجام بهینه‌سازی زمانبندی با هدف حداقل کردن زمان اجرا و انتقال task صورت می‌گرفت و TBSLB به منظور توازن بار سیستم با مهاجرت task از Overloaded VMs. هدف اصلی این الگوریتم کاهش مصرف انرژی در ماشین‌های مجازی بود. در [17] به منظور توازن بار الگوریتمی بر مبنای کلاستر ارائه شد. از معماری Master-Slave در این الگوریتم استفاده شده است وهدف آن پیدا کردن Master Node مناسب برای اختصاص Task به منظور تعادل بار بر روی کلاسترهاست. در [18] یک الگوریتم پویا برای تعادل بار پردازنده در محیط محاسبات ابر معرفی شد. این الگوریتم از تکنیک مهاجرت و همچنین الگوریتم‌های توازن بار و انتخاب یک دوره زمانی مناسب برای مهاجرت کارها و انتخاب منبع بهینه جهت مهاجرت کار و اجرای کار بر روی آن، به منظور ایجاد تعادل بار محاسبات پردازنده‌ها بهره برد. در [19] با استفاده از پیش‌بینی Fuzzy یک الگوریتم پویای زمانبندی کار به منظور برقراری تعادل معرفی شد. این الگوریتم (SALAF) از سیستم منطق فازی نوع یک و دو استفاده می‌کند و شامل دو بخش زیر می‌باشد: Availability fuzzy prediction برای بررسی توانایی سیستم که می‌تواند مسئله را بصورت فازی حل کند یا خیر. Load balance fuzzy prediction که در این بخش استنتاج وضعیت بار هرسرور، استنتاج وضعیت توازن بار مرکز داده‌ی مجازی، انتخاب مجموعه‌های نامزد بر طبق وضعیت توازن بار انجام می‌شد. در [20] با الهام از رفتار زنبور عسل الگوریتمی بنام HBB-LB برای توازن بار معرفی شد. در این الگوریتم به دو بحث اصلی توازن و اولویت پرداخته شد. در بحث توازن بار مراحل تصمیم توازن بار، گروه‌بندی Vms، انتقال Task و انتخاب Vm صورت می‌پذیرد.

۳- الگوریتم مرتب سازی بایتونیک

این الگوریتم، یک الگوریتم موازی مرتب سازی است که از آن برای ساخت شبکه‌های مرتب سازی استفاده می‌شود. روند کار مرتب سازی بایتونیک نیز به این صورت است: یک توالی از عناصر را بایتونیک گوییم هرگاه اعداد در ابتدا افزایش و سپس کاهش یابد یا برعکس. بطور مثال اعداد (۱،۲،۴،۷،۶،۰) به صورت بایتونیک مرتب شده‌اند. هر شیفت چرخشی از بایتونیک نیز یک توالی بایتونیک است، مانند(۸،۹،۲،۱،۰،۴) [۲۱].

۳-۱- شبکه مرتب سازی بایتونیک

وظیفه‌ی این شبکه، مرتب کردن توالی بایتونیک است.

[5] Shortest Task First

[6] Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization

بنابراین ورودی یک توالی n عنصری بایتونیک است. و دارای $\log n$ ستون است (تعداد گام‌ها) و هر ستون شامل n/2 مقایسه گر می‌باشد. شبکه مقایسه گرها جهت انجام عمل مقایسه استفاده می‌شود. از تعدادی ستون تشکیل شده است. هر ستون حاوی تعدادی مقایسه‌گر است که در هر ستون مقایسه‌گرها به صورت موازی به هم متصل هستند. جواب نهایی خروجی ستون آخر است. سرعت شبکه بستگی به عمق آن دارد [۲۱].

۳-۲- تحلیل پیچیدگی زمانی شبکه بایتونیک

پیچیدگی الگوریتم فوق به صورت $\theta(\log n^2)$ است که در رابطه ۱ نشان داده شده است.

$$\begin{cases} 1 & if\ n = 1 \\ 2T\left(\frac{n}{2}\right) + f(n), else \end{cases}$$

رابطه ۱- تحلیل پیچیدگی زمانی شبکه بایتونیک

در رابطه فوق $\boldsymbol{2\,T\left(\frac{n}{2}\right)}$ بیانگر هزینه‌ی حل دو زیر مسأله به صورت بازگشتی است و f(n) یک تابع مجانبی مثبت است و بیانگر هزینه‌ی ترکیب (ادغام) راه حل آنها می‌باشد که مقدار آن $\log n$ است. در نتیجه با توجه به اصل قضیه‌ی حل معادلات بازگشتی داریم:

$$T(n) = \theta(\log n^2)$$

رابطه (۲): هزینه اجرای الگوریتم بایتونیک

٤- روش پیشنهادی

در این مقاله یک روش جدید برای زمانبندی پروسه‌ها در محیط محاسبات ابری معرفی می‌شود که از یک الگوریتم مرتب‌سازی جهت مرتب کردن پروسه‌ها برای ورود به پردازنده بهره می‌برد. این رویکرد با هدف مرتب‌سازی پروسه و تقسیم پروسه بین ماشین‌های مجازی موازی، طراحی و پیاده سازی شده است. همانند بسیاری دیگر از الگوریتم‌های زمانبندی، پروسه‌ها ابتدا به صورت عادی و بدون هیچ سیاست یا الگوریتم مرتب‌سازی تعریف می‌شوند. سپس وارد مکانیزم زمانبندی شده و طبق الگوریتم تعریف شده به منظور استفاده از پردازنده مرتب و زمانبندی می‌شوند. این الگوریتم به این صورت است که ابتدا پروسه‌ها وارد یک شبکه‌ی مقایسه‌گر می‌شوند. این شبکه شامل یکسری مقایسه گر می‌باشد که با همدیگر عملیات مقایسه را انجام می‌دهند و سپس پروسه‌ها را به صورت صعودی- نزولی یا نزولی-صعودی مرتب می‌کنند که بتوانند از پردازنده استفاده کنند. در زمانبندی ارائه شده قصد داریم که با استفاده از یک الگوریتم مرتب‌سازی پروسه، پروسه‌ها را با شبکه مرتب سازی بایتونیک مرتب کنیم. علت استفاده از این شبکه این است که تعادل بار بر روی پردازنده‌ها را برقرار کند. با توجه به اینکه روش مرتب‌سازی ارائه شده، پروسه‌ها را به صورت ترکیب صعودی- نزولی یا بالعکس مرتب می‌کند، همه‌ی بار بر روی پردازنده‌ها به طور تقریباً متعادل تقسیم می‌شود اما در سایر روش‌ها به دلیل مرتب‌سازی یک بعدی (فقط صعودی یا نزولی) ممکن است پردازنده‌هایی فقط پروسه‌های با تعداد دستورالعمل زیاد را پردازش کنند و در مقابل پردازنده‌های دیگر پروسه‌های با تعداد دستورالعمل کم را پردازش کنند. هدف اصلی از الگوریتمی‌که ارائه خواهد شد، زمانبندی پروسه به منظور تعدیل بار کاری پردازنده‌هاست. به این صورت که پردازنده‌ها بار متعادل و تقریباً نزدیک به هم را پردازش کنند. با استفاده از زمانبندی میزان بار پردازنده‌ها به یک مقدار متعادلی می‌رسد و پردازنده‌ها از لحاظ تعداد پروسه‌هایی که پردازش می‌کنند و همچنین میزان TFT تقریباً برابر و نزدیک به هم هستند. زمانبندی به این صورت است که پروسه‌ها طبق الگوریتم مطرح شده به یک سری صعودی – نزولی یا نزولی – صعودی از پروسه‌ها تبدیل می‌شوند و سپس پروسه‌ها در بین پردازنده‌ها توزیع می‌گردند. در روش‌ها و الگوریتم‌های مختلف زمانبندی پروسه، پروسه‌ها اغلب فقط به صورت نزولی یا فقط به صورت صعودی مرتب شده و برای زمانبندی، پروسه بر اساس یک سیاست و خط مشی تعریف شده از پردازنده استفاده می‌کند که در این حالت‌ها ممکن است بار پردازنده‌ها متعادل نباشد به این صورت که،

بعضی پردازنده‌ها بار بیشتری را نسبت به سایر پردازنده‌ها تحمل کنند و در بعضی از الگوریتم‌های زمانبندی، پروسه‌ها ممکن است یک زمان بسیار طولانی در صف انتظار منتظر باشند تا بتوانند از پردازنده استفاده کنند، یا اینکه بعضی از پروسه‌ها ممکن است با حداقل زمان انتظار وارد پردازنده شده و سرویس خود را گرفته و پردازش شوند. در این موارد و الگوریتم‌ها ممکن است یکسری پروسه‌ها که مدت زمان کوتاهی به پردازنده نیاز دارند، مدت زمان بیشتری انتظار بکشند و یکسری پروسه‌ها که مدت زمان زیادی به پروسه نیاز دارند، زمان انتظار کمتری نسبت به بقیه داشته باشند. چگونگی تولید این سری‌های صعودی-نزولی یا نزولی- صعودی و جزئیات کامل مراحل مرتب سازی پروسه‌ها در ادامه بیان خواهد شد.

خصوصیات این رویکرد به شرح زیر است:

یکی از بهترین الگوریتم‌های مرتب سازی از نظر پیچیدگی و هزینه است.

در الگوریتم‌های دیگر خروجی به صورت یک بعدی بوده است یعنی سری خروجی یا کاملاً صعودی بود یا کاملاً نزولی؛ اما سری خروجی این الگوریتم ترکیبی از دو زیر سری صعودی و نزولی است.

۱. پروسه‌ها برای اجرا روی کامپیوترها ارسال می‌شوند. ارسال پروسه‌ها توسط Broker انجام می‌شودکه خود یک ماشین است و در واقع کلاسی است که کارگزار را مدل می‌کند.

٤- شبیه سازی و ارزیابی نتایج

جهت پیاده‌سازی از محیط برنامه‌نویسی Eclipse win64[7] استفاده شده است. همچنین از شبیه‌ساز CloudSim نسخه ۳,۰,۳ بهره برده شده است. در محیط شبیه‌سازی یک مرکز داده[8] تعریف شده است که در این مرکز داده امکان تعریف هر تعداد پروسه و ماشین مجازی با توجه به سلیقه‌ی کاربر وجود دارد که می‌توان مرکز داده را به صورت همگن یا ناهمگن تعریف کرد و پروسه‌های مختلف را بر روی آن شبیه‌سازی و اجرا نمود. الگوریتم پیشنهادی با الگوریتم زمانبندی STF[9] از نظر پارامترهای میزان موازی سازی، کاهش TFT، بهره‌وری CPU، بیشترین زمان اتمام پروسه، توان عملیاتی پردازنده‌ها و میانگین زمان انتظار پروسه‌ها در صف پردازنده‌ها در سناریوهای مختلف مورد بررسی و مقایسه قرار گرفت که در ادامه نتایج یکی از سناریوها بیان می‌گردد.

٥-١- سناریوی شماره یک

این سناریو [22 و 23] دو مرکز داده‌ی همگن و ناهمگن را با هم مقایسه می‌کند. از داده‌های این الگوریتم برای ارزیابی رویکرد پیشنهادی استفاده می‌شود. مشخصات کلی محیط شبیه‌سازی به صورت زیر است: در جدول ۱، مشخصات پروسه‌ها بیان می‌گردد.

جدول ۱- مشخصات پروسه‌ها

شماره پروسه	تعداد دستورالعمل (MI)	زمان اجرای تخمین زده شده (Sec)
۱	۳۰۷۲۰	۱۸۰۰
۲	۲۰٤۸۰	۱۸۰۰
۳	۳۰۷۲۰	۱۸۰۰
٤	۲۰٤۸۰	۲٤۰۰
٥	۳۰۷۲۰	۳۰۰۰
٦	۲۰٤۸۰	۲٤۰۰
۷	۳۰۷۲۰	۲٤۰۰
۸	۱۰۲٤۰	۱۸۰۰

درجدول ۲، مشخصات ماشین‌های مجازی در حالت همگن بیان می‌گردد.

[7] http://www.eclipse.org
[8] DataCenter
[9] Shortest Task First

جدول ۲- مشخصات ماشین‌های مجازی در حالت همگن

شماره ماشین	قدرت پردازش (MIPS)
۱	۳۰۰
۲	۳۰۰
۳	۳۰۰

در جدول ۳، مشخصات ماشین‌های مجازی در حالت ناهمگن بیان می‌گردد.

جدول ۳- مشخصات ماشین‌های مجازی در حالت ناهمگن

شماره ماشین	قدرت پردازش (MIPS)
۱	٤۷۰
۲	۲۲۰
۳	۳۰۰

در جدول ٤، مشخصات کلی محیط شبیه‌سازی بیان می‌شود.

جدول ٤- مشخصات کلی محیط شبیه‌سازی

مشخصات پروسه‌ها	طبق جدول۱
قابلیت پردازش مرکز داده (MIPS)	۱۰۰۰
تعداد ماشین مجازی	طبق جداول ۲و ۳
میزان حافظه RAM ماشین‌های مجازی (MB)	۵۱۲
پهنای باند ماشین‌های مجازی (Mbit/Sec)	۱۰۰۰
میزان حافظه Ram مربوط به Host (MB)	۲۰٤۸
پهنای باند Host	۱۰۰۰۰

۵-۲- نتایج شبیه سازی سناریوی شماره یک

شکل ۱ میزان موازی سازی را نشان می‌دهد. در این شکل، محور عمودی بیانگر میزان موازی سازی و محور افقی نوع الگوریتم را نشان می‌دهد.

شکل ۱- میزان موازی سازی به ازای ۳ ماشین و ۸ پروسه

همانطور که در شکل مشخص است موازی‌سازی در رویکرد ارائه شده به ازای هر دو مرکز داده‌ی همگن و ناهمگن بهتر از الگوریتم STF است. با توجه به اینکه هدف از این الگوریتم تعادل بار پردازنده‌ها و کاهش TFT می‌باشد، TFT پردازنده‌ها و در نهایت بزرگ‌ترین TFT پردازنده‌ها کمتر از الگوریتم STF است. در نتیجه، میزان موازی سازی در رویکرد ارائه شده بیشتر از الگوریتم STF است.

شکل ۲ میانگین بهره‌وری از پردازنده‌ها را در مقایسه با هم نشان می‌دهد. در این شکل، محور عمودی بیانگر میزان بهره‌وری پردازنده‌ها و محور افقی نوع الگوریتم را نشان می‌دهد.

شکل ۲- میانگین بهره‌وری به ازای ۳ ماشین و ۸ پروسه

میزان همانطور که در شکل مشخص است، در هر دو الگوریتم و به ازای هر ٤ حالت میزان بهره‌وری برابر

است. دلیل آن است که چون پروسه‌ها و پردازنده‌هایی که در الگوریتم‌ها استفاده می‌شوند، مشابه هستند و با توجه به اینکه پروسه‌ها با هر الگوریتمی روی پردازنده‌ها توزیع شوند، هر چه زمان پردازش در پردازنده‌ها افزایش یا کاهش یابد به همان نسبت، زمان کل پردازش پروسه‌ها نیز افزایش یا کاهش می‌یابد. در نتیجه میانگین بهره‌وری از پردازنده‌ها در هر دو الگوریتم برابر است.

شکل‌های ۳، ٤ و ۵ کل زمان اتمام پردازش پروسه در پردازنده‌ها را در مقایسه با هم نشان می‌دهند. محور عمودی کل زمان اتمام پردازش پروسه بر روی هر پردازنده و محور افقی نوع الگوریتم را بیان می‌کند.

شکل ۳- TFTپردازنده شماره ۱

همانطور که در شکل مشخص است در رویکرد ارائه شده و در هر دو حالت مرکز داده‌ی همگن و ناهمگن مقدار بار پردازشی توسط پردازنده‌ها متعادل‌تر است.

شکل ٤- TFTپردازنده شماره ۲

همانطور که در شکل مشخص است در رویکرد ارائه شده و در هر دو حالت مرکز داده‌ی همگن و ناهمگن مقدار بار پردازشی توسط پردازنده‌ها متعادل‌تر است.

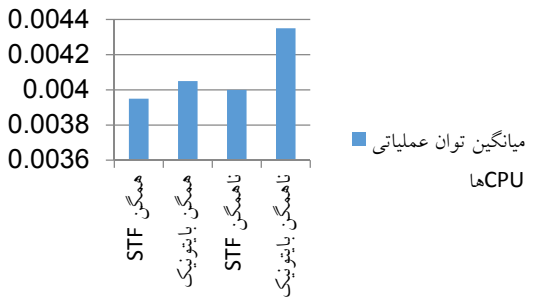
شکل ۵- TFTپردازنده شماره ۳

شکل ٦ بیشترین زمان اتمام پردازش پروسه، در دو الگوریتم را در مقایسه با هم نشان می‌دهد. محور افقی بیانگر بیشترین زمان اتمام پردازش پروسه و محور افقی بیانگر نوع الگوریتم است.

شکل ٦- TFT بیشترینTFTبه ازای ۳ ماشین و۸پروسه

با توجه به نتایج شبیه‌سازی، بیشترین TFT در رویکرد ارائه شده و در هر دو حالت مرکز داده‌ی همگن و ناهمگن، کمتر از بیشترین TFT در الگوریتم STF است و دلیل این است که در رویکرد ارائه شده پردازنده‌ها بار محاسباتی متعادل‌تری نسبت به هم دارند ولی در الگوریتم STF پردازنده‌ها متعادل عمل نکرده‌اند و بعضی از پردازنده‌ها بار بیشتر و بعضی دیگر بار کمتری را پردازش کرده‌اند.

شکل ۷ میانگین توان عملیاتی پردازنده‌ها را در مقایسه با هم نشان می‌دهد که محور عمودی بیانگر میانگین توان عملیاتی پردازنده‌ها و نمودار افقی بیانگر نوع الگوریتم است.



شکل ۷– میانگین توان عملیاتی به ازای ۳ ماشین و۸پروسه

شکل ۸– میانگین زمان انتظار به ازای ۳ ماشین و۸ پروسه

همانطور که در شکل مشخص است توان عملیاتی پردازنده‌ها در رویکرد ارائه شده در هر دوحالت مرکز داده‌ی همگن و ناهمگن بیشتر از الگوریتم STF بوده است. چون پردازنده‌ها متعادل‌ترند و پروسه‌ها متعادل‌تر توزیع می‌شوند، از حداکثر قدرت پردازش استفاده می‌کنند و پروسه‌ها زودتر و سریع‌تر اجرا می‌شوند. در نتیجه توان عملیاتی بالا می‌رود. هرچه پروسه‌های سنگین وزن بر روی پردازنده‌های قوی قرار بگیرند، توان عملیاتی بالاتر می‌رود. شکل ۸ میانگین زمان انتظار در صف پردازنده‌ها را در مقایسه با هم نشان می‌دهد. محور عمودی بیانگر میانگین زمان انتظار در صف پردازنده‌ها و محور افقی بیانگر نوع الگوریتم است.

همانطور که در شکل مشخص است در رویکرد ارائه شده ودر هر دوحالت مرکز داده‌ی همگن و ناهمگن، هر پردازنده تعدادی از پروسه‌های با تعداد دستورالعمل کم (سبک) و تعدادی از پروسه‌های با تعداد دستورالعمل زیاد (سنگین) را پردازش می‌کند. اما در الگوریتم STF ممکن است بعضی از پردازنده‌ها پروسه‌های سنگین و بعضی دیگر، پروسه‌های سبک را پردازش کنند. در نتیجه در پردازنده‌هایی که پروسه‌های سبک را بیشتر پردازش می‌کنند زمان انتظار کمتر از پردازنده‌ی متقابل آن پردازنده، در رویکرد ارائه شده است. بنابراین میانگین زمان انتظار در رویکرد ارائه شده بیشتر از الگوریتم STF می‌شود. در جدول شماره ۵ نتایج شبیه سازی برای برای سناریوی شماره یک بصورت مقایسه‌ای بیان می‌شود.

جدول ۵– مقایسه نتایج شبیه سازی سناریوی یک

	میانگین زمان انتظار	میانگین توان عملیاتی	بیشترین زمان اتمام پروسه	میانگین بهره وری پردازنده‌ها	میزان موازی سازی
STFهمگن	۱۲٤	۰,۰۰٤۱۰	۲٦٥	۰,۳۳	۲,۳۸
بایتونیک همگن	۱۳٦	۰,۰۰٤۲۰	۲٤۰	۰,۳۳	۲,۷۲
STFناهمگن	۱۱٤	۰,۰۰٤۳۹	۲۷۳	۰,۳۳	۲,۲٥
بایتونیک ناهمگن	۱۲۰	۰,۰۰٤٤۲	۲٤۰	۰,۳۳	۲,٦۳

همانطور که از جدول مشخص است رویکرد پیشنهادی در پارامترهای میزان موازی سازی، بیشترین زمان اتمام پروسه، میانگین توان عملیاتی پردازنده‌ها در هر دو حالت همگن و ناهمگن بهتر از الگوریتم STF عمل نموده است. اما در پارامتر میانگین زمان انتظار پردازنده الگوریتم STF بهتر از رویکرد پیشنهادی است چرا که در رویکرد پیشنهادی در هر دوحالت مرکز داده‌ی همگن و ناهمگن، هر پردازنده تعدادی از پروسه‌های با تعداد دستورالعمل کم (سبک) و

تعدادی از پروسه‌های با تعداد دستورالعمل زیاد (سنگین) را پردازش می‌کند. اما در الگوریتم STF ممکن است بعضی از پردازنده‌ها پروسه‌های سنگین و بعضی دیگر، پروسه‌های سبک را پردازش کنند. در نتیجه در پردازنده‌هایی که پروسه‌های سبک را بیشتر پردازش می‌کنند، زمان انتظار کمتر از پردازنده‌ی متقابل آن پردازنده، در رویکرد ارائه شده است. بنابراین، میانگین زمان انتظار در رویکرد ارائه شده بیشتر از الگوریتم STF می‌شود.

۵-۳- سناریوی شماره دو

در این سناریو [24] عناصر پردازشی مختلف برای اجرای یک عملیات پردازش تصاویر، با همدیگر تعامل دارند. ٤٠ تصویر مختلف در قالب ٤ دسته‌ی ١٠ تایی به ٤ پردازنده‌ی مختلف ارسال می‌شوند. هر تصویر در حالت غیر فشرده به عناصر پردازشی ارسال می‌شود. دستورالعمل‌های مورد نیاز برای پردازش در قالب یک کاربرد بسته‌بندی شده و به عناصر پردازشی ارائه می‌شود. در این مقاله تصاویر به ٤ روش پردازش می‌شوند و خروجی هر پردازنده در قالب تعداد دستورالعمل‌های پردازش شده (MI) نمایش داده می‌شود. در هر روش ٤ خروجی به ازای هر ٤ پردازنده خواهیم داشت. بنابراین ١٦ خروجی در کل تولید می‌شود. ١٦ خروجی تولید شده از این عملیات به عنوان ١٦ پروسه جهت استفاده در سناریو در نظر گرفته می‌شوند. در این سناریو، جهت تخمین زمان پردازش هر پروسه بدترین حالت در نظر گرفته شده است که پروسه بر روی ضعیف‌ترین پردازنده قرار گیرد. به این ترتیب ورودی‌ها به شرح زیر هستند. از داده‌های این الگوریتم برای ارزیابی رویکرد ارائه شده استفاده می‌شود. در شبیه سازی مشخصات محیط، به صورت زیر است:

جدول ٦- مشخصات ماشین‌های مجازی

شماره ماشین	قدرت پردازش (MIPS)
١	٣٠
٢	٤٠
٣	٥٠
٤	٣٥

جدول ٧- مشخصات پروسه‌ها

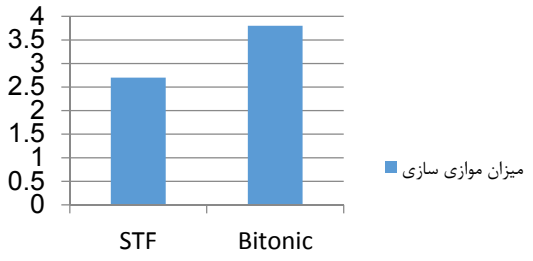
شماره پروسه	تعداد دستورالعمل (MI)	زمان اجرای تخمین زده شده (Sec)
١	١٧٠٤	٥٧
٢	٨٠٧	٢٧
٣	١٠٤٧	٣٥
٤	٧٦٢	٢٥
٥	١١٣١	٣٨
٦	٧٦٢	٢٥
٧	١٢٢٦	٤١
٨	١٢٠١	٤٠
٩	٢٥٨٨	٨٦
١٠	٢٩٧٨	٩٩
١١	٣٤٣٨	١١٥
١٢	٣٢٩٠	١١٠
١٣	١٧٦٠	٥٩
١٤	٢٩١٦	٩٧
١٥	٤٤٤٧	١٤٨
١٦	٣١٦٧	١٠٦

جدول ۸- مشخصات کلی محیط شبیه سازی

مشخصات پروسه‌ها	طبق جدول۷
قابلیت پردازش مرکز داده (MIPS)	۱۰۰۰
تعداد ماشین مجازی	طبق جدول٦
میزان حافظهRAM ماشین‌های مجازی(MB)	۵۱۲
پهنای باند ماشین‌های مجازی (Mbit/Sec)	۱۰۰۰
میزان حافظه RAM مربوط به Host (MB)	۲۰٤۸
پهنای باند Host	۱۰۰۰۰

٥-٤- نتایج شبیه سازی سناریوی شماره دو

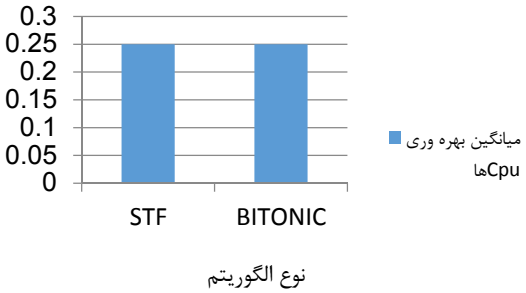
شکل ۹ میزان موازی‌سازی در دو الگوریتم را نشان می‌دهد.



شکل ۹- نمودار میزان موازی سازی به ازای ٤ ماشین و ١٦ پروسه

همانطور که در شکل مشخص است که موازی‌سازی بهتر از الگوریتم STF است. با توجه به اینکه هدف از این الگوریتم تعادل بار پردازنده‌ها و کاهش TFT می‌باشد، TFT پردازنده‌ها و در نهایت بزرگ‌ترین TFT پردازنده‌ها کمتر از الگوریتم STF است. در نتیجه، میزان موازی‌سازی در رویکرد ارائه شده بیشتر از الگوریتم STF است.

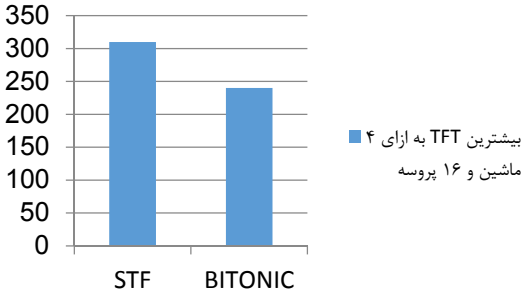
شکل ۱۰ میانگین بهره‌وری پردازنده‌ها در دو الگوریتم را نشان می‌دهد.



شکل ۱۰- نمودار میانگین بهره‌وری پردازنده‌ها به ازای ٤ ماشین و ١٦ پروسه

همانطور که در شکل مشخص است، میانگین بهره‌وری از پردازنده‌ها برابر است. دلیل آن است که چون پروسه‌ها و پردازنده‌هایی که در الگوریتم‌ها استفاده می‌شوند، مشابه هستند و با توجه به اینکه پروسه‌ها با هر الگوریتمی روی پردازنده‌ها توزیع شوند، هر چه زمان پردازش در پردازنده‌ها افزایش یا کاهش یابد به همان نسبت، زمان کل پردازش پروسه‌ها نیز افزایش یا کاهش می‌یابد. در نتیجه میانگین بهره‌وری از پردازنده‌ها در هر دو الگوریتم برابر است.

شکل ۱۱ بیشترین TFT در دو الگوریتم را نشان می‌دهد.

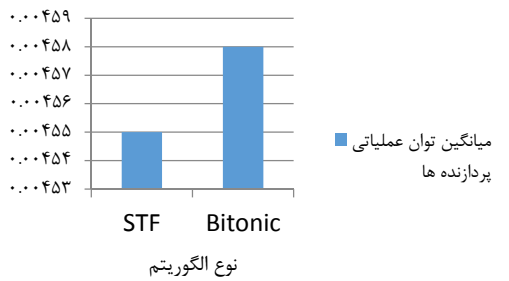


شکل ۱۱- بیشترین TFT به ازای ٤ ماشین و١٦ پروسه

همانطور که در شکل مشخص است، با توجه به نتایج شبیه سازی، بیشترین TFT در رویکرد ارائه شده، کمتر از بیشترین TFT در الگوریتم STF است و دلیل این است که در رویکرد ارائه شده پردازنده‌ها بار محاسباتی متعادل‌تری نسبت به هم دارند ولی در الگوریتم STF پردازنده‌ها متعادل عمل نکرده‌اند و بعضی از پردازنده‌ها بار بیشتر و

بعضی دیگر بار کمتری را پردازش کرده‌اند.

شکل ۱۲ میانگین توان عملیاتی پردازنده‌ها در دو الگوریتم را نشان می‌دهد.

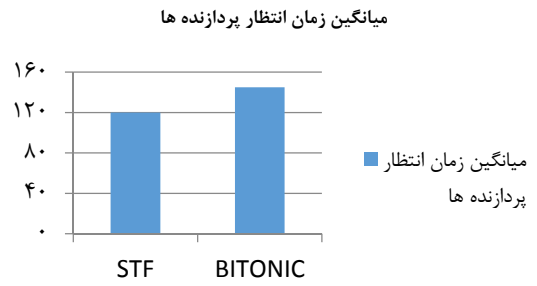


نوع الگوریتم

شکل ۱۲- میانگین توان عملیاتی پردازنده‌ها به ازای ٤ ماشین و ١٦ پروسه

همانطور که در شکل مشخص است، توان عملیاتی پردازنده‌ها در رویکرد ارائه شده بیشتر از الگوریتم STF بوده است. چون پردازنده‌ها متعادل‌ترند و پروسه‌ها متعادل‌تر توزیع می‌شوند، از حداکثر قدرت پردازش استفاده می‌کنند و پروسه‌ها زودتر و سریع‌تر اجرا می‌شوند. در نتیجه توان عملیاتی بالا می‌رود. هرچه پروسه‌های سنگین وزن بر روی پردازنده‌های قوی قرار بگیرند، توان عملیاتی بالاتر می‌رود.

شکل ۱۳ میانگین زمان انتظار پردازنده‌ها در دو الگوریتم را نشان می‌دهد.



شکل ۱۳- میانگین زمان انتظار پردازنده‌ها به ازای ٤ ماشین و ١٦ پروسه

در رویکرد ارائه شده هر پردازنده تعدادی از پروسه‌های با تعداد دستورالعمل کم (سبک) و تعدادی از پروسه‌های با تعداد دستورالعمل زیاد (سنگین) را پردازش می‌کند. اما در الگوریتم STF ممکن است بعضی از پردازنده‌ها پروسه‌های سنگین و بعضی دیگر، پروسه‌های سبک را پردازش کنند. در نتیجه در پردازنده‌هایی که پروسه‌های سبک را بیشتر پردازش می‌کنند زمان انتظار کمتر از پردازنده‌ی متقابل آن پردازنده، در رویکرد ارائه شده است. بنابراین میانگین زمان انتظار در رویکرد ارائه شده بیشتر از الگوریتم STF می‌شود.

در جدول شماره ۹ نتایج شبیه‌سازی برای پارامترهای میزان موازی‌سازی، میانگین بهره‌وری پردازنده‌ها، بیشترین زمان اتمام پروسه، میانگین توان عملیاتی پردازنده‌ها و میانگین زمان انتظار در صف پردازنده‌ها برای سناریوی شماره دو بصورت مقایسه‌ای بیان می‌شود.

جدول شماره ۹- نتایج شبیه‌سازی

	میانگین زمان انتظار در صف پردازنده‌ها	میانگین توان عملیاتی پردازنده‌ها	بیشترین زمان اتمام پروسه	میانگین بهره وری پردازنده‌ها	میزان موازی سازی
STF	۱۲۰	۰,۰۰٤٥٥	۳۰۰	۰,۲٥	۲,٦۸
بایتونیک	۱٤٥	۰,۰۰٤٥۸	۲۳۰	۰,۲٥	۳,۷٥

٦- نتیجه‌گیری

محاسبات ابری مجموعه‌ای گسترده از انواع منابع را به صورت مجازی در اختیار کاربران قرار می‌دهد. تخصیص منابع به درخواست‌ها به صورت پویا، موجب گردیده که زمانبندی کارها به یکی از چالش برانگیزترین مباحث این مقوله تبدیل گردد. در این مقاله، روشی جدید برای

زمانبندی کارها مطرح شد که از فاکتورهایی مانند زمان اتمام تخمینی و تعداد دستورالعمل‌های هر پروسه استفاده شد. نتایج نشان می‌دهد که در هر دو حالت مرکز داده‌ی همگن و مرکز داده‌ی ناهمگن الگوریتم بایتونیک بهتر از الگوریتم STF می‌باشد. هر چند که مرکز داده‌ی ناهمگن به عنوان یکی از خصوصیات معماری و زیر بنای محاسبات ابری می‌باشد اما باز هم در مرکز داده‌ی ناهمگن، نتایج الگوریتم پیشنهادی بهتر از الگوریتم STF است. شبیه‌سازی انجام شده نشان می‌دهد که در هر دو الگوریتم، میزان بهره‌وری پردازنده‌ها و در نتیجه میانگین بهره‌وری پردازنده‌ها برابر است. میزان موازی‌سازی و توان عملیاتی پردازنده‌ها در الگوریتم ارائه شده بهتر از الگوریتم STF است. میزان انتظار پروسه‌ها در الگوریتم پیشنهادی بیشتر از الگوریتم STF بوده است. در الگوریتم ارائه شده، پردازنده‌ها مقدار بار پردازشی متعادل‌تری را داشتند و پروسه‌ها سریع‌تر و متعادل‌تر شده‌اند.

۷- مراجع

[1] Foster, Ian, Yong Zhao, Ioan Raicu, and Shiyong Lu. "Cloud computing and grid computing 360-degree compared." In *2008 Grid Computing Environments Workshop*, pp. 1-10. Ieee, 2008.

[2] Karagiannis and C.N. Höfer · G, "Cloud computing services: taxonomy and comparison", open access at Springerlink,Vol. II, 2011.

[3] Sun. H et al., Research and Simulation of Task Scheduling Algorithm in Cloud Computing, *TELKOMNIKA Indonesian Journal of Electrical Engineering,* Vol.11, No.11, pp. 6664-6672,2013

[4] Shi, Daji Ergu · Gang Kou · Yi Peng · Yong Shi · Yu, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment", DOI 10.1007/s11227-011-0625-1, Springer Science+Business Media, 2013.

[5] D. Duncan, X. Chu, C. Vecchiola, and R. Buyya, The Structure of the New IT Frontier: Cloud Computing, Strategic Facilities Magazine, Issue 9, Pages: 67-72, Pacific & Strategic Holdings Pte Ltd, Singapore, and August/September 2009.

[6] Q. Li and Y. Guo, Optimization of Resource Scheduling in Cloud Computing, 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2010.

[7] Peter Mell, Timothy Grance, "The NIST definition of Cloud Computing(September, 2011)", Accessed on May, 2014.

[8] A E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, M. Livny, and Pegasus: Mapping scientific workflows onto the grid, in: European across Grids and Conference, 2004.

[9] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, I.Brandic, Cloud computing emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility , Future Generation Computer Systems 25(6)(2009),pp. 599-616.

[10] T. V. Truong Duy, Y. Sato, Y. Inoguchi, Performance Evaluation of a Green Scheduling Algorithm for Energy Savings in Cloud Computing, 1-1 Asahidai, Nomi, Ishikawa, 923-1292 Japan, 2010.

[11] **P**. Yin, S. Yu, P. Wang and Y. Wang, Task allocation for maximizing reliability of a distributed system using hybrid particle swarm optimization, Journal of Systems and software 80(5)(2007), PP 724-735

[12] Q. Kang, H. He, H. Song and R. Deng, Task allocation for maximizing reliability of distributed computing systems using honeybee mating optimization, Journal of Systems and Software 83(11), PP 2165-2174(2010).

[13] L.M. Zhang and K. Li,Green Task Scheduling Algorithms with Speeds Optimization on Heterogeneous Cloud Servers, 2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing, CPSCom.2010.

[14] S. Chen, P. Stephens, H. B. Chen, A Two-Step Policy Approach used in Cloud Web Service Scheduling Problems, New Mexico, Final Report, April 3rd, 2013.

[15] S. Nesmachnow, S. Iturriaga, B. Dorronsoro, E. Talbi, and P. Bouvry, List scheduling heuristics for virtual machine mapping in cloud systems, Universidad de la República, Uruguay LIFL, University of Lille, France University of Luxembourg, Luxembourg, July 29-30, 2013.
[16] M. Gahlawat, P. Sharma, Analysis and Performance Assessment of CPU Scheduling Algorithms in Cloud using Cloud Sim, International Journal of Applied Information Systems (IJAIS)
ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5 – No. 9, July 2013.
[17] Ramezani, Fahimeh, Jie Lu, and Farookh Khadeer Hussain. "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization." International Journal of Parallel Programming 42.5 (2014): 739-754.
[18] Dhurandher, Sanjay K., et al. "A cluster-based load balancing algorithm in cloud computing." *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014.
[19] Rajavel, Rajkumar, Thamarai Selvi Somasundaram, and Kannan Govindarajan. "Dynamic Load Balancer Algorithm for the Computational Cloud Environment." Information and Communication Technologies. Springer Berlin Heidelberg, 2010.
[20] Krishna, P. Venkata. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." *Applied Soft Computing* 13.5 (2013): 2292-2303.
[21] R. Rajavel, T. Mala, Achieving Service Level Agreement in Cloud Environment using Job Prioritization in Hierarchical Scheduling, 2010.
[22] S. Chen, P. Stephens, H. B. Chen, A Two-Step Policy Approach used in Cloud Web Service Scheduling Problems, New Mexico, Final Report, April 3rd, 2013.
[23] R. Rajavel, M. T, Achieving Service Level Agreement in Cloud Environment using Job Prioritization in Hierarchical Scheduling, Anna University, Chennai, Tamil Nadu, India, 2012.
[24] K. Dutta, a Smart Job Scheduling System for Cloud Computing Service Providers and Users: Modeling and Simulation, Computer Science & Engineering Department, Kalyani Government Engineering College, Kalyani, Nadia- 741235, West Bengal, India, 1st Int'l Conf. on Recent Advances in Information Technology (RAIT)-2012.