

## اولویت قابل تقدم مبتنی بر تخصیص منابع پویا در محاسبه ابری با تحمل خطا

### چکیده

امروز، محاسبات ابری به عنوان یک مدل پاسخ درخواستی به خدمت گرفته می شوند که یک مشتری برای خدمات موجود مختلف در "پرداخت شما به عنوان پایه و اساس" درخواست می دهد. پردازش ابری یک پدیده تخصیص منابع انعطاف پذیر پویا را پیشنهاد می دهد. برای خدمات قابل اعتماد و تضمین شده باید یک مکانیسم برنامه ریزی وجود داشته باشد که تمام منابع با کارایی برای برآوردن درخواست مشتری اختصاص داده شوند. خدمات ابری بر اساس ویژگی های مقیاس پذیری هستند، در دسترس بودن، امنیت و تحمل خطا می باشند. تأمین خدمات در ابر بر اساس SLA است. توافقنامه سطح خدمات شرایط قرارداد ارائه دهنده ابری با مشتریان برای تعریف سطح (های) خدمات فروخته شده با اصطلاحات زبان ساده است. QoS (کیفیت خدمات) نقش مهمی در محیط ابری بازی می کند. برنامه ریزی منابع و استقرار خدمات با توجه به پارامترهای چندگانه SLA نظیر CPU مورد نیاز، پهنای باند شبکه، حافظه و ذخیره سازی اجرا می شود. در این مقاله ما یک الگوریتم را پیشنهاد می دهیم که پیشدستی منابع از اولویت کم به اولویت بالا و رزرو پیشرفته برای منابع را با توجه به متعدد پارامتر های SLA برای استقرار خدمات اجرا می کند. این الگوریتم همچنین برای مکانیزم تحمل خطا موثر است.

**کلمات کلیدی:** محاسبات ابری ، SLA، زیرساخت به عنوان خدمات (IaaS)، بستر های نرم افزاری به عنوان خدمات (PaaS)، مدیریت منابع، نرم افزار به عنوان خدمات (SaaS)، ماشین مجازی، مجازی سازی.

## 1. مقدمه

محاسبه ابری " ارائه منابع محاسباتی و برنامه های کاربردی از طریق اینترنت " با استفاده از مدل پرداخت به شما است. همچنین به عنوان محاسبات اینترنت شناخته شده است، در اینجا مخزنی از منابع از قبیل حافظه، پردازنده، شبکه و پهنای باند، عملاً در سراسر اینترنت توزیع شده است. اگر یک مشتری بخواهد از خدمات ارائه دهنده ابری استفاده کند و پس با توجه به خدمات با استفاده از زمان واقعی مجبور به پرداخت در ازای الزامات هر هزینه می باشد. محاسبات ابری به اشتراک گذاری جهانی منابع و ظرفیت های ذخیره سازی نامحدود را فراهم می کند.

همانطور که می دانیم هر تعداد از مشتریان می توانند ارائه دهنده ابری را درخواست دهند، اگر توافق بر اساس SLA رخ دهد، بدان معناست که ارائه دهنده ابری قادر به رسیدن به درخواست مربوطه از کاربر است، این امر با برنامه ریزی موثر منابع و استقرار برنامه های کاربردی در ماشین های مجازی مناسب اجرا می شود. برنامه ریزی منابع یعنی چندگانگی درخواست های کاربران در ساختار فیزیکی یکسان. در این زمان کار بیشتر در حال حاضر در مورد برنامه ریزی منابع در ابر های این روش بر استقرار منابع جهانی با در نظر گرفتن یک SLA بر اساس هدف از قبیل هزینه اجرا، زمان اجرا، حداقل منابع و غیره است.

زمانی که درخواستی توسط مشتری ارسال شد، ابتدا به چند زیروظیفه تقسیم می شود. در حال حاضر چهار مسئولیت اصلی زمانبندی ابری وجود دارد، (1) شیوه ای مناسب یا سفارش به منظور اجرای یک کار پیدا کند ، (2) برنامه تخصیص منابع مناسب برای کار بیابد، (3) زمانبندی باید تحمل خطا برای برنامه سربار و خاتمه کار را داشته باشد، (4) شیوه ای برای مهاجرت از وظایف به شیوه ای موثر پیدا کند. با استفاده از پدیده تخصیص منابع این مشکلات می توانند حل شوند.

برنامه ریزی می تواند دو نوع باشد، برنامه ریزی جهانی و برنامه ریزی محلی. برنامه ریزی محلی یک نوع برنامه ریزی است که در آن منابع بومی سازی ترجمه برای برآوردن درخواست کاربر در یک ابر استفاده می شوند. برنامه ریزی جهانی در جایی است که در آن تمام منابع از ابرهای متعدد به عنوان یک واحد برای تکمیل درخواست کاربر رفتار می شوند. به طور معمول تأمین کارآمد به دو فرآیند مجزا نیاز دارد، (1) برنامه ریزی ثابت اولیه - برنامه ریزی محلی

که در آن تمام ماشین های مجازی برای منابع فیزیکی نقشه برداری شوند. (2) تامین منابع پویا ماشین های مجازی جدید، مهاجرت در ماشین های مجازی، پاسخ پویا با توجه به حجم کار ایجاد می کند. در اینجا مرحله 1 ، مرحله استقرار است که به طور کلی در زمان راه اندازی یک ابر انجام می شود، و هنگامی که تعمیر و نگهداری توسط منبع انجام می شود. خواه گام 2 بطور مکرر در زمان تخصیص اجرا شود. چالش های مختلفی در این منطقه برای محققان وجود دارد، مانند مقیاس پذیری، چندگرایشی، امنیت، تخصیص منابع پویا و تحمل خطا.

در این مقاله بر تأمین منابع پویا متمرکز شده ایم، روش برنامه ریزی اکتشافی با توجه به اهداف متعدد SLA ، مانند مقدار مورد نیاز پردازنده، شبکه پهنای باند و هزینه برای گسترش برنامه های کاربردی در ابرها را ارائه داده ایم. این برنامه ریزی یک انعطاف پذیری در استراتژی تخصیص منابع درخواستی شامل رزرو پیشرفته و مکانیزم قابلیت پیشدستی برای منابع ارائه می کند. الگوریتم پیشنهادی ما به صورت پویا به منبع درخواست برای این کار پاسخ می دهد. نخست آن را به صورت محلی برای دسترس بودن منابع بررسی می کنیم؛ اگر منابع رایگان باشند، پس ماشین های مجازی جدید برای کار فعلی مستقر می کنند، اگر منابع در دسترس نباشد و پس VM جدید از منابع در دسترس در سطح جهان آن را ایجاد می کنند. اگر منابع جهانی در دسترس نباشند، پس برای منابع مورد بررسی قرار خواهند گرفت اگر قابل پیشدستی باشد، پس در غیر این صورت فرآیندهای مهاجرت در لیست انتظار قرار گرفته و طرح رزرو پیشرفته را اعمال می کنند.

سازمان دهی مقالات به شرح زیر است: در بخش 2 کارهای تحقیقاتی مربوط به این موضوع را مطرح می کنیم، در بخش 3 مدلهایی برای تخصیص منابع و برنامه ریزی کاری در سیستم محاسبات ابری شامل الگوریتم های قبلی است که در الگوریتم ما استفاده می شود، بخش 4 در مورد روش ارائه شده بحث می کند، بخش 5 نتایج شبیه سازی را نشان می دهد ، بخش 6 نتیجه گیری مقاله و کار در آینده می باشد.

## 2. کار مرتبط

جیایین لی<sup>1</sup> [1] یک مکانیسم بهینه سازی منابع در سیستم های چند ابر IaaS ناهمگن متحد ارائه داد که برنامه ریزی مدل کار قابل پیش دستی را با تخصیص منابع، مدل سیستم ابر، نقشه برداری محلی و مصرف انرژی و مدل نرم افزاری را فعال می کند. این امر برای آینده خودکار در ابر و ماشین های مجازی مناسب است. آنها الگوریتم پویا برای تخصیص منابع آنلاین و برنامه ریزی کار پیشنهاد دادند. در منابع ابری خارق العاده پیشنهادی هر مرکز داده جدید سرور مدیری دارد، ارتباط و طرح تخصیص منابع بین سرور های مختلف از هر مرکز داده برای اشتراک گذاری تراکم کاری در میان سرور های داده های متعدد را تقسیم می کند. به اشتراک گذاری حجم کار ذخیره منبع بزرگی از منابع انعطاف پذیر و ارزان تر برای تخصیص منابع می سازد.

اس پاندی<sup>2</sup> [2] یک روش برنامه ریزی اکتشافی برای بهینه سازی هزینه های نقشه برداری منابع وظیفه بر اساس راه حل از روش بهینه سازی ازدحام عملی ارائه داد، PSO تاثیر الگوریتم هوشمند توسط رفتار اجتماعی حیوانات مانند یک گله از پرندگان در حال پیدا کردن یک منبع مواد غذایی و یا یک دسته ماهی در حال دفاع از خود در برابر شکارچی است. یک ذره در PSO شبیه به یک پرنده و یا یک ماهی است که در یک فضای جنبش تحقیق هر ذره از یک سرعت حرکت هر یک از ذرات تشکیل شده است که و با سرعتی هماهنگ است که دارای بزرگی و جهت است. هر موقعیت ذرات در هر نمونه از زمان تحت تاثیر بهترین موقعیت و بهترین جایگاه ذرات در یک فضای مسئله است. عملکرد یک ذره با یک مقدار تناسب اندازه گیری می شود که مختص مسئله است. ارزش تناسب در این الگوریتم استفاده می شود که بهترین موقعیت برای ذرات است. PSO یک روش اکتشافی برنامه ریزی برای نرم افزار برنامه ریزی پویا است.

وینسنت سی امیکاروا<sup>3</sup> [3] یک روش اکتشافی برنامه ریزی با هدف تخصیص برای ماشین های مجازی بر اساس مدت سطح SLA و شرایط آن ارائه داد. این امر بر استقرار منابع فیزیکی بر اساس قابلیت دسترسی منابع تمرکز دارد، به طوری که نقض SLA را با استفاده از این روش کاهش می دهد. یک متعادل کننده بار یکپارچه برای بهره برداری

<sup>1</sup> Jiayin Li

<sup>2</sup> S. Pandey

<sup>3</sup> Vincent C. Emeakaroha

از منابع بالا و کارآمد در محیط محاسباتی ابری فراهم می کند. الگوریتم استراتژی تعادل بار محلی فراهم می کند و پدیده تخصیص منابع جهانی برای بهره برداری از منابع بهتر در محیط ابری است.

چاندراشخار اس. پوار<sup>4</sup> [4] نشان دهنده یک الگوریتم برای اولویت مبتنی بر استراتژی تخصیص منابع پویا در محیط ابری است، برنامه ریزی اصلاح شده اکتشافی برای اجرای بالاترین اولویت را فراهم می کند که به عنوان AR (رزرو پیشرفته) برای ممانعت از بهترین کار تلاش پیشدستانه است. این الگوریتم اجرای مناسب در محیط چند ابری را ارائه می کند. می توان در مجموعه ای متفاوت از مشاغل بیش از اجرای ابر های متعدد آن را اجرا کرد. درخواست های مختلف را می توان در حالت AR برای اهداف SLA اجرا کرد.

ژن شیائو<sup>5</sup> [5] طراحی و پیاده سازی یک سیستم مدیریت منابع خودکار را ارائه می دهد که می تواند برای جلوگیری از بار بیش از حد در سیستم باشد، در حالی که با به حداقل رساندن تعداد سرور های استفاده شده، مفهوم اندازه گیری چولگی کارکرد تبدیل سرور و کارکرد اصلاحی محدودیت منابع چند بعدی را معرفی می کند. یک الگوریتم پیش بینی بار که می تواند آینده کاربردهای منابع برنامه بدون جزئیات داخلی را حدس بزند، طراحی کنید. دوریان مینارولی و برنند فریزلبن<sup>6</sup> تخصیص منابع VM در محاسبه ابری از طریق کنترل فازی چند عامل با متمرکز بر منبع پویایی تخصیص VM محلی بر روی هر دستگاه فیزیکی یک ارائه دهنده ابری و در نظر گرفتن حافظه و CPU به عنوان منبع ارائه دادند که می تواند مدیریت شود. کنترل فازی برای کوچک کردن یک تابع مطلوبیت جهانی به عنوان n یک محاسبه سریع اکتشافی بر اساس قوانین فازی اجرا شد. ملاحظات مسئله این است که چگونه منابع از یک ارائه دهنده ابری باید VM را به صورت پویا دوباره تخصیص دهد که تغییرات بارکاری برای نگه داشتن عملکرد با توجه به SLA است.

---

<sup>4</sup> Chandrashekhar S. Pawar

<sup>5</sup> Zhen Xiao

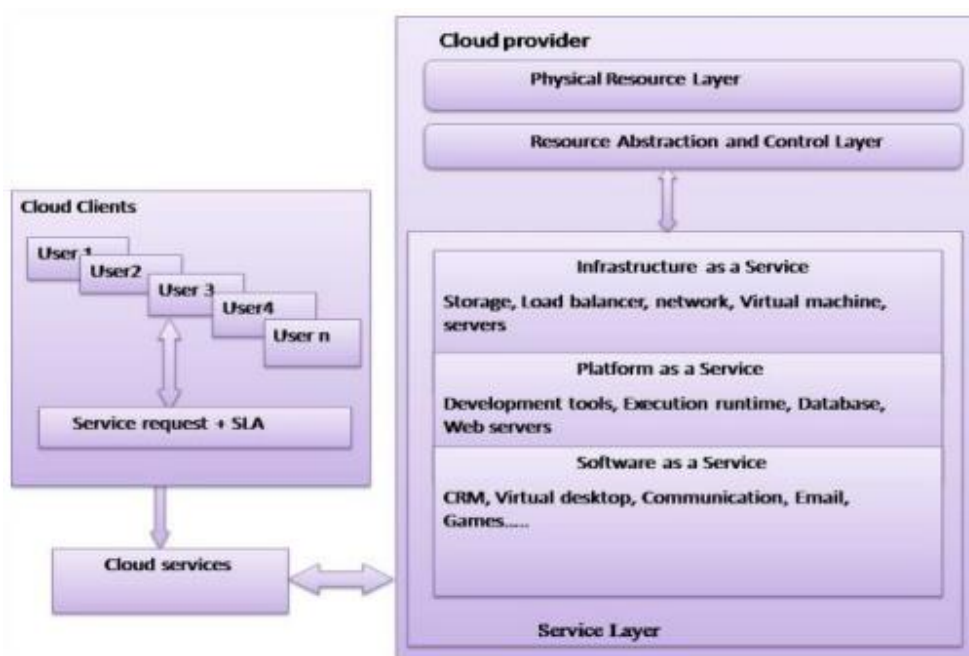
<sup>6</sup> Dorian Minarolli and Bernd Freisleben

### 3. استراتژی برنامه ریزی و مسائل طراحی

در این بخش ما نموداری بر اساس محیط ابری و برنامه ریزی اکتشافی پیشنهاد می دهیم. الگوریتم پیشنهادی ما به طور کلی در منابع SLA بر اساس تأمین و برنامه ریزی تطبیقی آنلاین برای قابلیت پیشدستی پیشرفته به منظور اجرای وظیفه است. دو گام اساسی برای استفاده موثر از منابع ابری مورد نیاز است که به اهداف SLA می رسد.

#### A. تخصیص منابع و استقرار نرم افزار

سه نوع لایه وجود دارد که ترکیبی برای استفاده در تأمین منابع برای سرویس درخواست کاربر است. به طور کلی به عنوان مدل های سرویس در محاسبات ابری شناخته شده است. PaaS و SaaS، IaaS.



شکل 1. لایه های خدمات ابری

در "شکل 1" چگونگی ارتباط با مشتری ابری با ارائه دهنده ابری برای استفاده از خدمات مختلف را نشان می دهد. شرایط SLA و شرایط باید برای اولین بار تأمین شود. سه نوع لایه در نمودار آورده شده وجود دارد: لایه خدمات، لایه کنترل و لایه های منابع فیزیکی. لایه کنترل منابع تخصیصی و مجازی سازی، لایه منابع شامل منابع فیزیکی درون یک ابر، لایه خدمات شامل سه نوع از خدمات IaaS، PaaS و SaaS را کنترل می کند.

SaaS (نرم افزاری به عنوان یک سرویس) برنامه های کاربردی و نرم افزار مورد در خواست کاربر را تحویل می دهد که توسط شخص ثالث اداره می شود. در هر مرحله شامل برنامه های کاربردی، مجازی سازی، زمان اجرا، QoS، سرور و مدیریت ذخیره سازی توسط فروشنده است.

PaaS (پلت فرمی به عنوان یک سرویس) پلت فرمی را به برای توسعه نرم افزار، با استفاده از سرویسهای Paas پلت فرم با توجه به درخواست کاربر برای هدف توسعه ارائه شده تحویل می دهد.

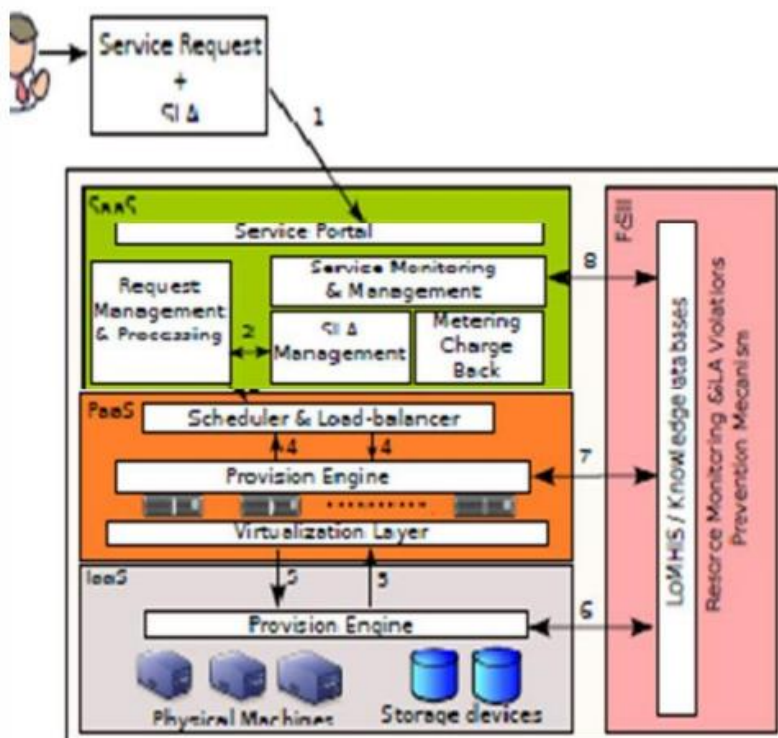
IaaS (زیرساخت به عنوان سرویس) منابع محاسباتی در مورد در خواست کاربر ارائه می کند؛ مانند فیزیکی یا VM ها، ذخیره سازی مبتنی بر فایل، متعادل کننده بار و فایروال.

تامین ابر و مدل استقرار نرم افزار [3] در شکل [2] نشان داده شده است. از سه نوع مدل خدمات ترکیب شده اند. در هر لایه، شامل مسئولیت هایی برای تأمین منابع برای رسیدن به اهداف SLA است. همانطور که در شکل [2] مشتری درخواست های خود را به پورتال خدمات می دهد (مرحله 1). پورتال خدمات درخواستی آنها را برای پردازش و بخش مدیریت برای بررسی اعتبار سنجیمی گذراند (مرحله 2). اگر درخواست اعتباری داشته باشد، پس آن را باید برای زمانبندی به جلو فرستاد و متعادل کننده بار را تأمین برای منابع مناسب فرستاد (مرحله 3). متعادل کننده بار موتور تأمین لایه PaaS برای انتخاب VM مناسب است و تأمین خدمات در حال اجرا مجازی ماشین آلات را متعادل می کنند (مرحله 4). لایه مجازی سازی برای مدیریت ماشین مجازی و تعامل با منابع فیزیکی با استفاده از لایه IaaS موتور تأمین استفاده می شود (مرحله 6).

ماتریکس منابع سطح پایین و منابع فیزیکی در لایه IaaS با چارچوب LOM2HiS نظارت می شوند (مرحله 6). اگر نقض SLA رخ دهد، اقدامات واکنشی توسط تکنیک های پایگاه داده دانش در FOSII (مرحله 7) ارائه می شوند. وضعیت خدمات درخواستی و اطلاعات SLA با پورتال خدمات مجدداً ارتباط برقرار می کنند (مرحله 8).

تأمین ذخیره را می توان در تک لایه به تنهایی انجام داد. اما روش و الگوریتم ما برای ارائه استراتژی برنامه ریزی منابع یکپارچه هدفمند شده است. بنابراین ما سه لایه را در نظر می گیریم. لایه LaaS مسئول مدیریت منابع

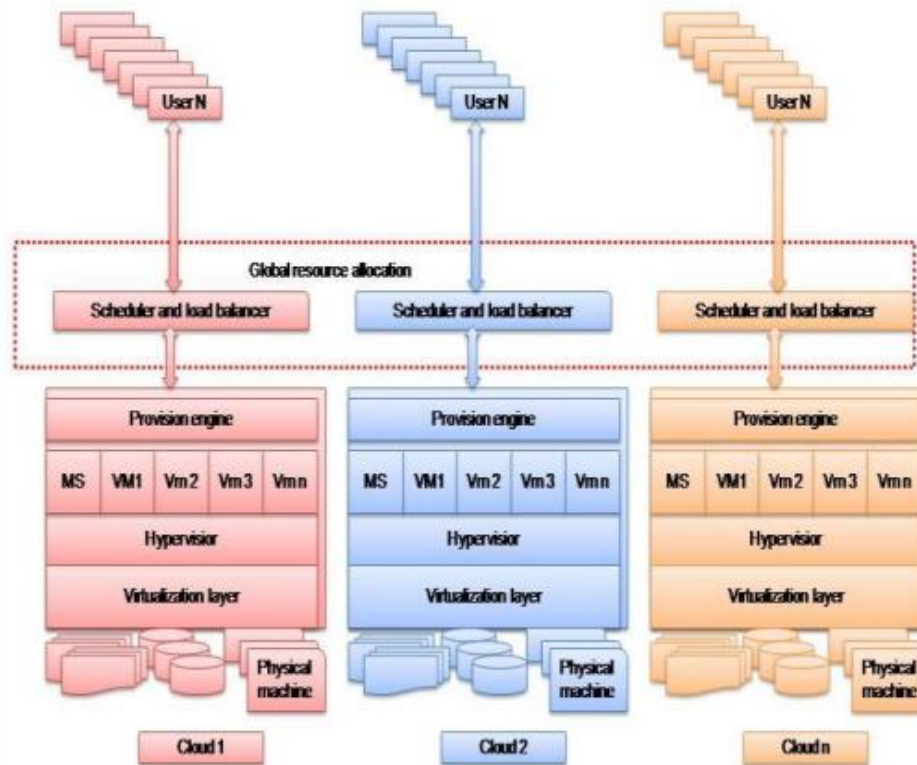
فیزیکی است. لایه PaaS مسئولیت گسترش و مدیریت VM است که با منابع فیزیکی با توجه به توافق SLA با مشتریان نقشه کشی می شود.



شکل 2. مدل تأمین و استقرار ابر

در شکل 3 برنامه ریزی منابع محلی و جهانی بین ابرها متعدد نشان داده شده است، همانطور که در برنامه ریزی ابر دیدیم،  $n$  عدد از کاربران وجود دارند که می تواند هر گونه درخواست زمان برای تامین کننده ابری ارائه دهند، اگر تمام منابع ابر فعلی در یک زمان مشغول باشند و درخواست جدید با اولویت بالا برسد، پس ارائه دهنده ابر باید درخواست را در لیست انتظار قرار دهد، بنابراین در اینجا یک راه حل ارائه شده در این الگوریتم برای برنامه ریزی و توزیع درخواست کاربر بطور موثر در سراسر ابرها و پیشدستی منابع از اولویت کم به اولویت بالا وجود دارد. الگوریتم ارائه شده همچنین قادر به بررسی تحمل خطا در سیستم ابری است.





شکل 3. تأمین منابع و برنامه ریزی بین ابرهای متعدد

## B. زمانبندی توضیحات اکتشافی

هنگامی که زمانبندی کننده درخواست خدمات کاربر را دریافت کند، آن اولین پارتیشن خواهد بود که وظیفه در چند کار فرعی و ساخت یک DAG خواسته می شود. این فرایند تخصیص منابع ثابت نامیده می شود. در [1] نویسندگان الگوریتم حریصانه ای برای تولید برنامه ریزی استاتیک تخصیص پیشنهاد دادند: CLS (الگوریتم زمان بندی لیست ابری)

1. برنامه ریزی لیست ابری (CLS)

این CLS مشابه [14] CPNT است. تعاریف مورد استفاده برای لیست وظایف به صورت زیر ارائه شده اند. اولین زمان شروع (EST) و آخرین زمان شروع (LST) از یک وظیفه در (1) و (2) نشان داده شده اند. وظایف ورودی EST برابر با O دارند. و LST وظایف خروجی با EST آنها برابر است.

$$EST(v_i) = \max_{v_m \in pred(v_i)} \{EST(v_m) + AT(v_m)\} \dots \dots (1)$$

$$LST(v_i) = \max_{v_m \in succ(v_i)} \{LST(v_m) - AT(v_m)\} \dots \dots (2)$$

همانطور که سیستم ابری مربوطه در [II] با زمان اجرای هر وظیفه ناهمگن است، در ماشین های مجازی ابرهای مختلف یکسان نیستند.  $AT(V_i)$  متوسط زمان اجرای وظیفه  $V_i$  است. گره بحرانی (CN) مجموعه ای از رئوس با EST و LST برابر در DAG هستند. الگوریتم 1 یک تابع تشکیل یک لیست وظایف را بر اساس اولویت ها نشان می دهد.

### الگوریتم 1 تشکیل یک لیست کار بر اساس اولویت

- نیاز (ورودی): یک DAG ، میانگین زمان اجرای AT از هر وظیفه در DAG  
 اطمینان (خروجی): لیستی از وظایف P بر اساس اولویت
1. EST برای هر وظیفه محاسبه شده است
  2. LST برای هر کار محاسبه شده است
  3.  $T_p$  و  $B_p$  هر کار محاسبه شده است
  4. لیست خالی P و پشته S و همه وظایف در فهرست وظیفه U آمده است
  5. فشار کار CN به پشته S ترتیب LST را کاهش می دهد
  6. زمانی که پشته S بطور خالی اجرا نمی شود
  7. اگر (S) بالا پیشینیان فوری پس از آن انباشته دارد پس
  8.  $S \leftarrow$  روند فوری با حداقل LST است
  9. مورد دیگر
  10. (S) بالا  $P \leftarrow$
  11. پاپ تاپ (S)
  12. پایان در صورتی که
  13. پایان در حالی که

هنگامی که در الگوریتم بالا 1 فرم از لیست وظایف با توجه به اولویت وجود دارد، ما می توانیم منابع را برای وظایف در جهت تخصیص لیست تشکیل شده اختصاص دهیم. هنگامی که تمام وظایف کار اختصاص داده شده به پایان رسید و منابع ابری در دسترس برای آنها اختصاص داده شد، کار اختصاص داده شده شروع اجرا خواهد کرد. این وظیفه از لیست پس از انتساب آن حذف شده است. این روش تا زمانی که لیست خالی است، تکرار می شود.

#### 4. الگوریتم زمان بندی

در این بخش یک الگوریتم برای رسیدگی به اولویت درخواست مشتری و ارائه رزرو پیشرفته و پیش دستی بر منابع پیشنهاد می شود که نسخه اصلاح شده از الگوریتم قبلی در [4] است. در اینجا بالاترین اولویت کار بیش از وظیفه AR و وظیفه مرتبط به بالاترین هزینه پرداخت توسط مشتریان تعریف می شود

---

#### الگوریتم 2 اولویت بر اساس الگوریتم زمان بندی

---

1. ورودی:  $AP (R, AR)$

2. // الگوریتم فراخوان 1 برای شکل دادن به لیست وظیفه بر اساس اولویت های

3. دریافت جهانی فهرست VM موجود و دریافت فهرست VM جهانی و همچنین فهرست منابع در دسترس هر زمانبندی ابری

4. // پیدا کردن فهرست VM مناسب هر زمان بند ابری

5. اگر  $AP (R, AR) = \emptyset$  ، پس

6. دریافت لیست VM موجود در محل و فهرست منابع موجود در محل و دریافت لیست VM محلی استفاده شده  
7. بررسی برای منابع مناسب ، در غیر این صورت پیدا کردن در لیست های محلی در دسترس

سپس شروع جدید VM نمونه

اضافه کردن به لیست VM برای VM استفاده شده

اعزام سرویس در ماشین مجازی جدید

مستقر شده = درستی

8. سایر موارد اگر،

منابع جهانی قادر به میزبانی فوق العاده VM باشند پس شروع جدید VM نمونه

اضافه کردن VM به لیست VM موجود

اعزام سرویس در ماشین مجازی جدید

مستقر شده = درستی

9. دیگری در صورتی که،

اگر  $R =$  وظیفه هزینه بالا" و یا " $R =$  رزرو پیشرفته" && "VM دسترس => درخواست VM

فراخواندن الگوریتم 3 برای پیشدستی و رزرو پیشرفته

10. به روز رسانی لیست محلی و جهانی موجود و VM

11. مورد دیگر

خدمات صف درخواست تا زمانی که

زمان صف < زمان انتظار

12. مستقر شده = نادرست

13. پایان در صورتی که

14. اگر مستقر شده پس

15. بازگشت موفقیت آمیز

16. خاتمه

17. مورد دیگر

18. شکست بازگشت

19. خاتمه

در الگوریتم [2] R درخواست خدمات به مشتریان است، A اطلاعات نرم افزار، S بر اساس SLA و شرایط است. این موارد به عنوان ورودی برای زمانبندی (مرحله 1) فراهم می شوند. وقتی که درخواست برای یک سرویس برای زمانبندی ابری ارسال می شود، پس زمانبندی تقسیم در بسیاری از کارهای زیر به عنوان وابستگیهای آنها هستند، برای این الگوریتم هدف 1 نامیده می شود. الگوریتم 1 نیز برای تشکیل یک لیست از وظایف بر اساس وابستگی های آن استفاده می شود (گام 2). در (مرحله 3) زمانبندی دریافت لیست VM های موجود جهانی، و لیست کامل منابع، برای استقرار خدمات از هر ابری برای کاربر در دسترس است. لیست VM نیز برای اضافه کردن اطلاعات VM مستقر شده اند. در (مرحله 4-5) از شرایط SLA برای یافتن یک لیست از ماشین های مجازی مناسب استفاده می شود که قادر به تأمین درخواست خدمات R است. در (مرحله 6) زمانبندی تمام اطلاعات محلی و لیست ها با همه اطلاعات ردیافت می شود سپس متعادل کننده بار به صورت محلی تصمیم می گیرد که VM به درخواست خدمات اختصاص دارد (مرحله 7). وقتی که هیچ VM و منبع درخواست وجود ندارد که در آن زمان در دسترس باشد، پس از آن زمانبندی در سطح جهانی برای منابع رایگان بررسی می شود اگر هر منبع در سطح جهان رایگان است پس از آن استقرار خدمات در منابع توسط ایجاد یک نمونه VM جدید صورت می گیرد (مرحله 8). دیگری یعنی اگر هیچ منابع اضافی در دسترس محلی و جهانی برای اولویت کار وجود ندارد، اگر کار هزینه های بالایی دارد یا آن رزرو پیشرفته دارد پس زمانبندی الگوریتم (3) اجرا می شود (مرحله 9). در (مرحله 10) زمانبندی لیست خود را در صورت هر گونه تغییر، در طول تخصیص منابع به روز رسانی می کند. در هر مورد دیگر درخواست حکم به صف لیست انتظار اضافه می شود تا VM با منابع مطلوب رایگان دریافت شود (مرحله 11). اگر پس از یک دوره معینی از

زمان درخواست خدمات زمانبندی و مستقر شد پس زمانبندی به عنوان وضعیت استقرار موفقیت آمیز و در غیر این صورت با شکست آن را به مدیریت برمی گرداند (مرحله 14-19).

### الگوریتم 3 رزرو پیشرفته و تقدم مبتنی بر الگوریتم min-min

ورودی: مجموعه ای از وظایف،  $m$  ابرهای مختلف ماتریس ETM

خروجی: یک برنامه تولید شده توسط CMMS یا گزارش تقدم

1. برای یک مجموعه کار قابل طراحی مجموعه  $P$

2. در حالی که وظایف واگذار شده وجود دارد اجرا شود

3. به روز رسانی مجموعه وظیفه  $P$  قابل طراحی

4. برای  $i = 1$  وظیفه  $V_i \in P$  اجرا شود

5. ارسال لیست وظیفه درخواستی از  $V_i$  به تمام موارد دیگر زمانبند ابری

6. دریافت اولین منابع زمان در دسترس پاسخ و لیستی از کار با اولویت های آنها

ساخت تمام زمانبندی ابری دیگر

7. یافتن ابر  $C_{min}(V_i)$  با دادن اولین زمان پایان  $V_i$ ، با فرض اینکه هیچ وظیفه مقدم دیگر  $V_i$  وجود ندارد

8. اگر منابع "قابلیت تقدم  $R_i =$ "

به روز رسانی صف دایره ای برای منبع خاص بعلاوه وظیفه  $V_i$  زمانبندی ابری صف و تنظیم آماده یک تایمر برای آن را پس از 2 برش زمانی و اعزام آن قطعی کند.

پاسخ: اگر  $2 \leq$  زمان حلقوی کار

1. پردازش در حالی که منابع پس از تکمیل کنار گذاشته می شوند

2. پردازش بعدی در صف آماده  $B$  یگیری

اگر کار پشت سر هم باشد  $2 <$

1. تایمر  $= 1 -$

2. اجرای روند و قرار دادن آن در انتهای صف دایره ای

3. تکرار گام 1,2,3 تا صف دایره  $=$  "خالی"

C.استقرار = درست برای این کار

9. پایان برای

10. دیگری اگر

11. پیدا کردن جفت وظیفه ابری ،  $(V_k C_{min}(V_k))$  با اولین زمان پایان در جفت تولید شده در حلقه

12. اختصاص وظیفه  $V_k$  برای ابر  $D_{min}(V_k)$

13. حذف  $V_k$  از  $P$

14. به روز رسانی مجموعه کار  $P$  قابل طراحی

## 15. دیگری اگر

همه کارهای در حال اجرا بر روی ابر "AR" = وظایف

سپس

1. مشاهده تمام منابع  $R_i$  با اولین کار پایان  $V_j$  و رزرو پیشرفته = 'پوچ'،

2. رزرو پیشرفته در منبع خاص

$R_i$  توسط کار  $V_j$

3. پس از اتمام کار قبلی  $V_j$ ،  $R_i$  اختصاص داده می شود به کار  $V_i$

16. پایان زمانی که

یک برنامه زمانبندی ابری اجرای تمام منابع را با استفاده از یک حافظه ثبت می کند. مجموعه ای کار های طراحی شده به الگوریتم اختصاص داده شده است، مجموعه کار قابل طراحی تنها مجموعه ای است که از کار برای VM و منابع مربوطه اختصاص داده شده پشتیبانی کرده است. اگر یک اولویت بالا (هزینه بالا) باشد و پس آن باید با رزرو پیشرفته ارائه شده با توجه به الگوریتم داده شده وجود داشته باشد. اگر وظیفه ای با منبع مناسب در شرایط آزاد وجود ندارد و زمانبندی یک اولویت بالا است، پس از آن این الگوریتم را با این نسخه های وظایف در سه مرحله خواهد داشت

1) نخست آن را برای اولین منبع زمان موجود از تمام ابرها بررسی کرده سپس آن را با درخواست فعلی منابع قابل تقدم بودن یا نه بررسی می کنیم، اگر جواب مثبت است پس از آن تعویض اجرا بین وظایف در همان منبع با استفاده از یک زمان کوانتوم اجرا می شود. بنابراین وضعیت استقرار وظیفه فعلی درست است و تمام وظایف می توانند با موفقیت در ماشین مجازی مستقر شوند.

2) اگر منبع درخواست شده قابل تقدم باشد، و AR وظیفه ای برای یک ابر باشد، ابتدا در دسترس بودن منابع در این ابر اختصاص داده خواهد شد و زمانبندی ابر بررسی می شود. از آنجا که بهترین تلاش کار می تواند با کار AR تقدم یابد، تنها مورد زمانی است که بیشترین منابع توسط برخی از کارهای AR دیگر رزرو باشد. از این رو منابع کافی برای این کار AR در زمان مورد نیاز وجود ندارد. اگر وظیفه AR رد نشود، که معنی آن این است که به اندازه کافی منابع در دسترس برای این کار وجود دارد، یک مجموعه ای از ماشین های مجازی مورد نیاز به دلخواه انتخاب شده اند.

3) اگر تمام وظایف در حال اجرا در وظیفه AR ابرباشند، پس الگوریتم ما رزرو پیشرفته در اولین منابع کار برای پایان ارائه می دهد.

در اولویت قابلیت تقدم اگر منابع قابل تقدم باشند پس فقط برای نزدیکترین منابع بررسی شده و اختصاص آن به کار پس از تعویض بین وظایف انجام می شود. زمانی که یک کار کامل شود، پس آن را از صف دایره ای و نتیجه بازگشت حذف می کند. در رزرو پیشرفته، اگر یک منبع غیر قابل تقدم باشد پس زمانبندی فقط کنترل کار درخواستی را به همه ارائه دهنده ابری دیگر و دریافت اولین زمان در دسترس از منابع مربوطه می فرستد و پس از آن، سرور مدیر این ابر ابتدا منابع در دسترس در این ابر را بررسی می کند. از آنجا که وظایف AR می تواند از وظایف با تلاش بهتر تقدم یابد، اگر منابع توسط برخی از AR دیگر در زمان مورد نیاز محفوظ شوند، پس از آن، وظیفه AR منابع توسط رزرو پیشرفته در زمانی است که منابع رایگان دریافت خواهد کرد.

---

#### الگوریتم 4 برای تحمل خطا

---

1. اگر وظایف در حال اجرا در حال حاضر،  $T_i = \{T_1, T_2, \dots, T_n\}$

2. منابع کنونی برای کار  $T_n$  اختصاص یافته اند  $T_n = \{R_1, R_2, \dots, R_n\}$

3. خطا روی داد

4. وضعیت منابع؛  $R = "F AILED"$

5. سپس

6. اجرای اختصاص بالاترین اولویت به  $T_i$

7.  $T_i$  (اولویت) = "بالاترین"

8. اجرای الگوریتم 2

9. اگر مستقر شد پس

10. استقرار = "درست"

11. بازگشت = "موفقیت آمیز"

12. دیگری

13. تعلیق تمام وظایف موفق

14. قرار دادن وظیفه  $\{T_i, T_{i+1}, T_{i+2}, \dots, T_n\}$  در لیست انتظار

15. استقرار = "نادرست"

16. بازگشت = "ناموفق"

17. پایان

الگوریتم 4 برای مکانیسم تحمل خطا استفاده می شود. همه درخواست های خدمات و استقرار آنها در VM در محیط پویا انجام می شود، در محاسبات ابری هنوز هم شانس وجود دارد که منابع را در طول اجرای شکست خورده وظایف دریافت کنید. پس از آن یک مسئولیت برای زمانبندی ابری وجود دارد که آن منابع را به عنوان وضعیت شکست خورده، و بلافاصله مهاجرت وظیفه بر روی ماشین مجازی جدید نشان داده می شود. برای این الگوریتم هدف 2 می تواند یک گزینه بهتر داشت، زیرا بطور کارآمد منابع جدید اختصاص داده شده است. در الگوریتم 4 وظیفه فعلی در حال اجرا =  $T_i$ ، منابع جاری برای کار  $T_i$  اختصاص داده می شود  $R_i$  است (در مرحله 1-2). اگر هر مورد از منابع دلیل نبود سخت افزار شکست خورده باشد (مرحله 3) پس بالاترین اولویت زمانبندی را برای آن کار  $T_i$  اختصاص می دهد؛ (مرحله 6) و اجرا الگوریتم 2 (مرحله 7). در (مراحل 8 تا 10) اگر استقرار کار مربوطه با استفاده از الگوریتم 2 انجام شود و پس آن را می توان موفق انجام داد، در غیر این صورت زمانبندی تمام وظایف موفق تعلیق یافته و در لیست انتظار قرار داده می شوند.

## 5. نتیجه شبیه سازی

در این بخش در مورد نتیجه آزمایش ما با توجه به الگوریتم بحث می کنیم. عملکرد اکتشافی از طریق شبیه سازی داده ها را با استفاده از مجموعه وظایف مختلف در 10 اجرا می شود، ارزیابی می کنیم. با مجموعه ای از 60 درخواست خدمات مختلف و هر درخواست خدمات تشکیل شده در 10 تا 15 زیرکار آزمایشی انجام می دهیم. 4 ابر در شبیه سازی را در نظر می گیریم. درخواست های مختلف را می توان در ابر دلخواه اجرا کرد. زمان ورود درخواست ها با همدیگر متفاوت است. برخی از کارها به عنوان حالت AR اجرا می شوند و بقیه آنها را به عنوان بهترین حالت تلاش اجرا می شوند. ما شبیه سازی را به صورت محلی با استفاده از این پارامترها بدون اجرای آن را در هر سیستم از ابر های موجود و یا با استفاده از VM مرتبط با API انجام می دهیم .

آزمایش ما در دو حالت آزاد و وضعیت تنگنا کار می کند؛ در هر مورد نتیجه بهتر از الگوریتم غیر پیشگیرانه است. این شبه کد بهتر از الگوریتم های موجود است. [3] [4] زیرا رزرو در منابع و وظیفه مهاجرت بیش از منابع قابل



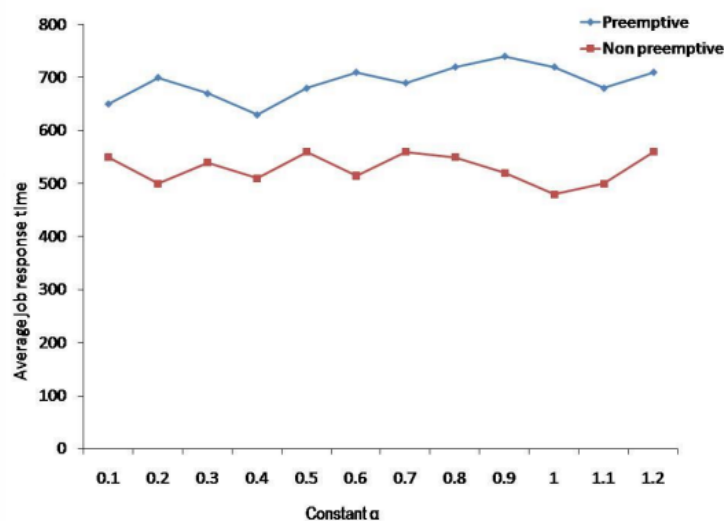
تقدم دارد. زمانبندی مجدد این وظایف با پیش تعریف احتمال  $a$  صورت می گیرد. پارامترها در جدول 1.1 در مجموعه شبیه سازی به طور تصادفی با حداکثر و حداقل مقادیر خود ارائه شده اند. بنابراین ما بر ساز و برنامه ریزی تمرکز می کنیم.

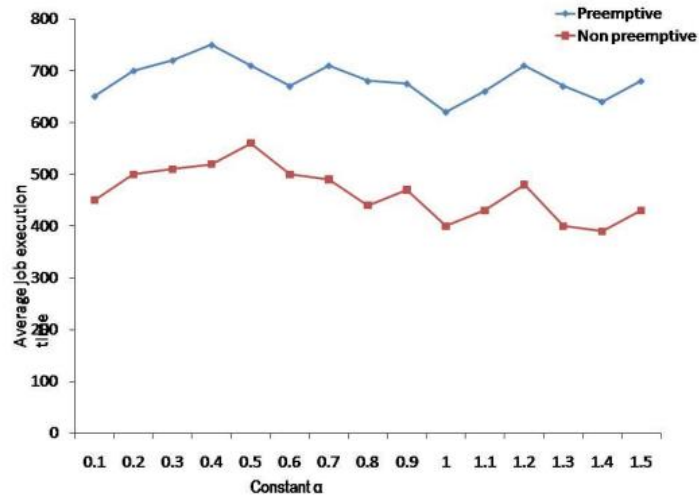
جدول 1.1 محدوده پارامترها

Parameter	Minimum	Maximum
ETMi,j	27	120
Number of VMs in a cloud	22	120
Number of CPU in a VM	4	10
Memory in a VM	40	2048
Disk space in VM	5000	10000
Speed of copy in disk	100	1000

### نتیجه

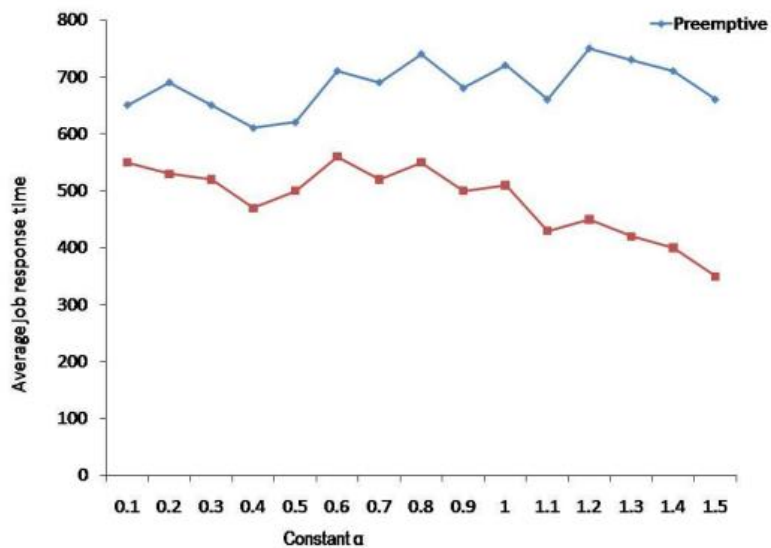
شکل 3 میانگین زمان پاسخ (الف) و متوسط زمان اجرا در وضعیت آزاد را نشان می دهد، وضعیت آزاد جای است که در آن ما زمان ورود درخواست کنونی را با یکدیگر تنظیم می کنیم، بنابراین درگیری منابع کمتر وجود دارد. اگر هر گونه درگیری منابع رقابتی رخ دهد پس بهترین کار تلاش برای تقدم توسط وظایف AR است. در نتیجه آزمایش درمی یابیم که آن پاسخ بهتری به زمان و حداقل متوسط زمان اجرا می دهد.

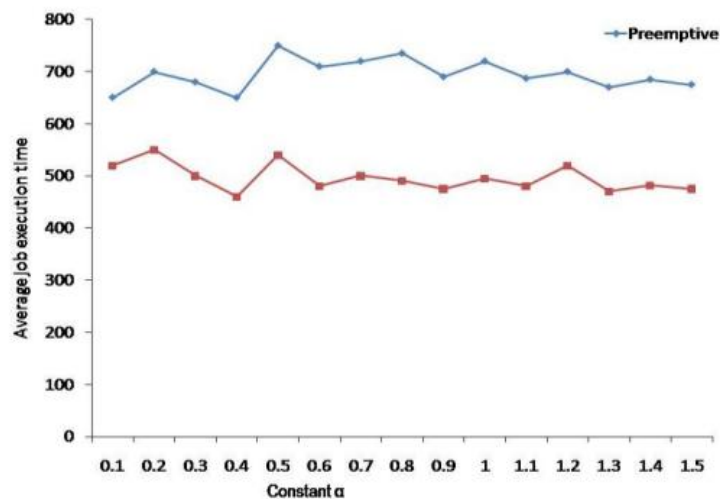




شکل 3. (a) و (b) متوسط زمان پاسخ و اجرا در وضعیت آزاد

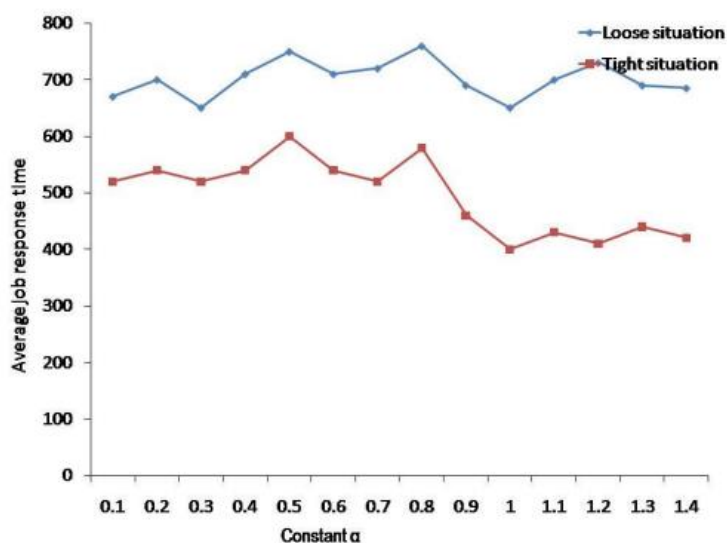
شکل 4 نتیجه وضعیت تنگنا را نشان می دهد که در آن الگوریتم داده شده بهتر از الگوریتم های موجود اجرا می شود [4] [3]. برای هدف تجربی زمان ورود منابع را بسیار نزدیک به یکدیگر تنظیم کردیم. روش تطبیقی در وضعیت تنگنا موثر تر است. زمان پاسخ متوسط (C) و زمان اجرای متوسط (d) بهتر از الگوریتم های قبلی است.





شکل 4. (c) و (d) زمان اجرای پاسخ متوسط و در شرایط تنگنا

شکل 5 نتیجه الگوریتم آزمایش تحمل خطا را در وضعیت آزاد و تنگنا در زمان پاسخ متوسط را نشان می دهد. این الگوریتم در هر دو شرایط بهتر کار می کند. زمانی که یک شکست در منابع در طول اجرای یک روند رخ می دهد، پس از آن زمان پاسخ باید با زمانبندی بسیار سریع باشد. این الگوریتم سعی می کند تا منابع را در بدترین شرایط تعیین کند، اگر هیچ گونه منابع رایگان و قابل الویت بندی پس از رزرو آن در منابع با تضمین وجود ندارد که باید منابع را بسیار سریع دریافت کنند.



شکل 5. متوسط زمان پاسخ در وضعیت آزاد و تنگنا

## 6. نتیجه گیری

تحمل خطا، QOS، دسترسی و مقیاس پذیری هنوز چالش ها باز در زمینه محاسبات ابری هستند. در این مقاله، برنامه ریزی اکتشافی برای مکانیزم تخصیص منابع پویا با منابع پیشدستی در ابر را ارائه کرده ایم. برنامه ریزی محلی و جهانی با توجه به درخواست خدمات کاربر ارائه کرده ایم. همچنین یک طرح جدید برای اولویت بالای آن نیز برای مکانیزم تحمل خطا در مدیریت منابع ارائه دادیم. هنگامی که یک منبع شکست می خورد، بلافاصله منبع جدید برای کار فراهم می شود. در این الگوریتم اولویت کار در مورد هزینه و محدودیت زمان تعریف شده است. الگوریتم زمان بندی خود را با استفاده از شبیه سازی محلی ارزیابی کردیم. سناریوهای ارزیابی و آزمون را برای بهره برداری از منابع، استقرار خدمات و تحمل خطا استفاده کردیم. کار آینده- ما می توانیم بر روی چندین پارامتر کار کنیم که اولویت وظایف مانند حداقل ملزومات منابع، زمان CPU، هزینه و شبکه می باشد. همچنین میتوانیم هدف انرژی کارآمد در تخصیص و استفاده از منابع را بررسی کنیم. بحث اکتشافی ارائه شده نیز می تواند در محیط محاسبات FOG شبیه سازی شوند. طرح جدید می تواند برای بکار گرفتیم در لیست انتظار توسعه یابد. ما همچنین می توانیم بر روی مکانیسم هایی کار کنیم که چگونه مجرای وظایف را تعیین کنیم.

## REFERENCES

- [1] Jiayin Li, Meikang Qiu, Jian-Wei Niu, Yu Chen, Zhong Ming, "Adaptive Resource Allocation for Preemptible Jobs in Cloud Systems," in 10th International Conference on Intelligent System Design and Application, Jan. 2011, pp. 31-36.
- [2] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in AINA '10: Proceedings of the 2010, 24th IEEE International Conference on Advanced Information Networking and Applications, pages 400-407, Washington, DC, USA, 2010, IEEE Computer Society.
- [3] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer, Ivan Breskovic, "SLA-Aware Application Deployment and Resource Allocation in Clouds", 35th IEEE Annual Computer Software and Application Conference Workshops, 2011, pp. 298-303.
- [4] Chandrashekhar S. Pawar , Rajnikant B. Wagh "Priority Based Dynamic resource allocation in Cloud Computing", 2013 International Conference on Intelligent Systems and Signal Processing (ISSP).
- [5] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen , " Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment".
- [6] Dorian Minarolli and Bernd Freisleben Department of Mathematics and Computer Science, University of Marburg," Virtual Machine Resource Allocation in Cloud Computing via Multi-Agent Fuzzy Control", 2013 IEEE Third International Conference on Cloud and Green Computing
- [7] 1. Brandic.Towards self-manageable cloud services. In 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC'09), 2009
- [8] R. N. Calheiros, R Ranjan, ABeloglazov, C.A FD.Rose, and R. Buyya. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. In Software: Practice and Experience. Wiley Press, New York, USA, 2010.
- [9] O.H.Ibarra and C.E.Kim, "Heuristic Algorithms for Scheduling Independent Tasks on Non-identical Processors," Journal of the ACM, pp. 280-289, 1977.
- [10]Ravi Jhavar( Graduate Student Member IEEE), Vincenzo Piuri( Fellow, IEEE)and Marco Santambrogio( Senior Member, IEEE), "Fault Tolerance Management in Cloud Computing: A System-Level Perspective" 2013 IEEE International Conference.
- [11]Y. c. Emeakaroha, 1. Brandic, M. Maurer, and S. Dustdar, "Low level metrics to high level SLAs - LoM2His framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," In High Performance Computing and Simulation.
- [12]y' C. Lee, C. Wang, A Y. Zomaya, and B. B. Zhou.Profit-driven service request scheduling in clouds. In 10th IEEE ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), 2010, pages 15 -24, may. 2010.
- [13]Shi J.Y., Taifi M., Khreishah A,"Resource Planning for Parallel Processing in the Cloud," in IEEE 13th International Conference on High Performance and Computing, Nov. 2011, pp. 828-833.
- [14]T. Hagraas and J. Janecek,"A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems," Parallel Computing, vol. 31, no. 7, pp. 653-670, 2005.
- [15]G. lung, and K. M. Sim, -Agent-based Adaptive Resource Allocation on the Cloud Computing Environment,1 in 40th International Conference on Parallel Processing Workshops (ICPPW'11), Taipei City, 2011, pp. 345 - 351.