

## مکان یابی کاربردهای چند لایه‌ای در بستر مرکز داده مجازی شده

کیوان رحیمی زاده\*<sup>۱</sup>، مرتضی آنالویی<sup>۲</sup>، پیمان کبیری<sup>۳</sup>

\*۱- نویسنده مسئول: استادیار، دانشکده فنی و مهندسی، دانشگاه یاسوج، یاسوج، ایران، RahimiZadeh@yu.ac.ir

۲- دانشیار، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران، AnaLoui@iust.ac.ir

۳- استادیار، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران، Peyman.Kabiri@iust.ac.ir

چکیده- رویکرد روز افزون فراهم کنندگان سرویس به پیاده سازی کاربردهای چند لایه‌ای (MTA) در بسترهای مجازی شده، ارزیابی کارایی در دستیابی به سرویس‌های مقیاس پذیر و انعطاف پذیر را مهم می‌سازد. فراهم سازی کارایی مناسب برای کاربردهای چند لایه‌ای مجازی شده (VMTA) که دارای پیچیدگی ساختاری و ارتباطی زیادی می‌باشند، نسبت به بسترهای سنتی بسیار سخت تر است. در این پژوهش، راهبردی برای مکان‌یابی ماشین‌های مجازی میزبان VMTAها پیشنهاد می‌شود. در گام نخست، رتبه بندی ماشین‌های مجازی میزبان لایه‌ها براساس تابع تولید کاب-داگلاس انجام می‌گیرد. سپس، VMTAها براساس میزان بهره‌وری منابع و کارایی اولویت بندی می‌شوند. در گام نهایی، VMTAها با توجه به نیازمندی منابع لایه‌ها، وابستگی بین آنان و وضعیت کارایی میزبانان فیزیکی، مکان یابی می‌شود. دست آوردها نشان می‌دهد که راهبرد پیشنهادی، کارکرد مناسبی در جایابی لایه‌های VMTAها در راستای کاهش تداخل کارکردی، ترازندی بار، کاهش میزان نقض سرویس و مصرف انرژی مرکز داده دارد.

واژه‌های کلیدی: کاربردهای چند لایه‌ای مجازی شده، مکان یابی، ارزیابی کارایی، محاسبات ابر، مرکز داده

### ۱- مقدمه

فرستاده می‌شود، مدیریت چگونگی پیکربندی و کنار هم قرارگیری سرویس‌ها بسیار مهم است. چنانچه یکپارچه سازی سرویس‌ها بدون در نظر گرفتن کارکرد ماشین‌های میزبان، وضعیت بهره‌وری منابع و چینش سرویس‌های میهمان انجام گیرد، این کار نه تنها ممکن است منجر به افزایش میزان تخطی از سطح سرویس توافقی شود، بلکه هزینه‌های پرپایی سرویس را نیز زیاد خواهد نمود. همچنین، مهاجرت‌های پی‌درپی ماشین‌های مجازی به دلیل یکپارچه سازی نادرست آنان، موجب کاهش پارامترهای کمی و کیفی سرویس‌های ارائه شده می‌گردد. از سوی دیگر، کاهش انرژی مصرفی یکی از اهداف سرویس‌دهندگان مراکز داده است. در این راه، مهاجرت ماشین مجازی که ابزاری کارا در یکپارچه سازی ماشین‌های مجازی و کاهش شمار میزبانان فیزیکی در یک مرکز داده است، نقش شایانی را در کاهش هزینه‌های مدیریتی و نگهداری مرکز داده بازی می‌کند [۱،۱۳،۱۴].

کاربردهای چند لایه‌ای (MTA) امکان برپایی سرویس‌های اینترنتی با مقیاس پذیری و انعطاف پذیری بالا را فراهم می‌سازند. با توجه به استفاده رو به افزایش فراهم کنندگان سرویس به برپایی MTAها، ارزیابی کارایی این سرویس‌ها در بسترهای مجازی شده

پیشرفت‌های فناوری مجازی سازی در زمینه فراهم سازی سرویس‌های گوناگون و پیاده سازی سیستم‌های اینترنتی، اهمیت ارزیابی این فناوری را برای دستیابی به کیفیت سرویس بهتر و کارایی مناسب سیستم برجسته می‌سازد. مجازی سازی سرور با یکپارچه سازی چندین ماشین مجازی، امکان استفاده اشتراکی از منابع فیزیکی را با انتزاعی سازی روی یک سرور فراهم می‌سازد. پایشگر ماشین مجازی (VMM یا Hypervisor) به عنوان نرم افزار مجازی ساز، نقش مدیریت و جداسازی کارکرد ماشین‌های مجازی را به عهده دارد. پایشگر ماشین مجازی با میانجی‌گری درخواست‌های فرستاده شده از سوی ماشین‌های مجازی، آنها را به منابع فیزیکی هدایت نموده و امکان هم نشینی همزمان چندین سرور مجازی شده را فراهم می‌سازد. در حقیقت، لایه مدیریت ماشین مجازی با نگاشت عملکرد اجرایی ماشین‌های مجازی شامل حالت اجرایی و دنباله عملیات اجرایی به لایه منابع فیزیکی این تسهیلات را فراهم می‌کند.

از آنجا که از سوی کاربران درخواست‌های متفاوتی از دید نیازمندی‌های منابع محاسباتی و ذخیره‌ای به یک مرکز داده

در راستای کاهش تداخل کارکرد ماشین‌های مجازی میزبان MTAها، ترازمندی بار، کاهش میزان نقض سرویس و مصرف انرژی مرکز داده، یک راهبرد مکان‌یابی در این پژوهش پیشنهاد و ارزیابی می‌شود. در گام نخست، رتبه‌بندی ماشین‌های مجازی، که میزبان لایه‌های VM TAها می‌باشند، براساس بهینه‌سازی خطی و استفاده از تابع تولید کب-داگلاس [۷] انجام می‌گیرد. در این گام براساس میزان تقاضای منابع و پارامترهای کیفی سرویس، ماشین‌های مجازی که ناکاراترند برای جایابی در گام بعدی برگزیده می‌شوند. سرانجام گام مکان‌یابی، با توجه به وابستگی بین لایه‌های VM TAها و با رویکرد کمینه کردن تداخل کارکردی ماشین مجازی و ترازمندی بار در مرکز داده انجام می‌گیرد.

ساختار مقاله در بخش‌های پیش‌رو چنین است. پیشینه پژوهش‌های انجام گرفته در بخش دوم بررسی می‌شود. در بخش سوم، تابع تولید کب-داگلاس برای رتبه‌بندی ماشین‌های مجازی ارائه می‌شود. با مدل‌سازی VM TAها براساس میزان کارایی آنان، در بخش چهارم ماشین‌های مجازی برای مکان‌یابی رتبه‌بندی می‌شوند. راهبرد مکان‌یابی آگاه به بارکاری در بخش پنجم و ارزیابی کارایی آن در بخش ششم بیان می‌شود. سرانجام، دست‌آورد این پژوهش و کارهای آینده در بخش هفتم ارائه خواهد شد.

## ۲- پیشینه پژوهش

بیشتر پژوهش‌های انجام گرفته به ارزیابی کارایی سرورهای تک لایه‌ای مجازی شده و اثر رقابت بین ماشین‌های مجازی در محیط اشتراکی مجازی شده می‌پردازند [۱۰-۸]. همچنین بخش زیادی از پژوهش‌های انجام گرفته در زمینه مکان‌یابی و مهاجرت کاربردهای یک لایه‌ای است [۱۱، ۱۲].

با توجه به پیچیدگی ساختاری کاربردهای چندلایه‌ای و همچنین بستر مجازی شده، ارزیابی کارایی این کاربردها و بهره‌گیری از دست‌آوردهای آن برای مکان‌یابی ماشین‌های مجازی وابسته بسیار پیچیده است. در [۵] به بررسی جابجایی تنگنا بین لایه‌های یک کاربرد چندلایه‌ای در بستر سنتی می‌پردازد. افزایش نرخ ورود درخواست‌ها به یک لایه منجر به کاهش کارایی یک لایه شده و این کاهش کارایی ممکن است به لایه‌های وابسته سرایت کند. در نهایت، کارایی کلی کاربرد کاهش چشم‌گیری می‌یابد.

فراهم‌سازی منابع برای MTAها از مهمترین چالش‌های کارایی است [۱۷-۱۵]. در [۱۸] زمان‌بندی و ارزیابی کارایی یک کاربرد سه لایه‌ای در مراکز داده ابری مورد بررسی قرار گرفته است. همچنین نویسندگان چگونگی توزیع کاربردهای سه لایه‌ای،

بسیار مهم است. هرچند ارزیابی کارایی، یکپارچه‌سازی و مکان‌یابی کاربردهای چند لایه‌ای مجازی شده (VM TA) به ندرت مورد بررسی قرار گرفته است، ارائه این سرویس‌ها به ویژه در سیستم‌های توزیع شده مانند بستر مبتنی بر محاسبات ابر، نیازمند بررسی دقیق است. بیشتر اینکه، ارزیابی کارایی کاربردهایی مانند VM TAها که دارای پیچیدگی ساختاری و ارتباطی زیادی می‌باشند در بستر مرکز داده مجازی‌شده، نسبت به بسترهای سنتی بسیار سخت‌تر است.

فارغ از مزایایی که مجازی‌سازی سرور دارد، این فناوری با چالش‌های نیز همراه است. نخست، مدیریت ماشین مجازی تنها به مجازی‌سازی سیستم عامل پرداخته و برنامه‌ها و سرویس‌های یکپارچه شده از اجرا در بستر مجازی شده اطلاعی ندارند. هر چند که هدف پایشگر ماشین مجازی ایجاد بستر اجرایی هم‌تراز با معماری سنتی در پیکربندی سرورهاست، اما استفاده هم‌زمان ماشین‌های مجازی از منابع سیستمی موجب بروز تداخل عملیاتی ماشین‌های مجازی و کاهش کارایی سرویس‌ها می‌شود [۲، ۳، ۳۱]. دوم، هیچ یک از فناوری‌های مجازی‌سازی برای همه کاربردها، کارایی یکسانی ندارند. زیرا که چگونگی کارکرد پایشگر ماشین مجازی، سیستم عامل و روش تخصیص منابع در پردازش بارهای کاری تاثیر مستقیم دارد. سوم، پیچیدگی‌های فناوری مجازی‌سازی، فراهم‌سازی کیفیت سرویس مناسب را با چالش‌های زیادی روبرو می‌کند. تعاملات لایه‌های مختلف سیستم مجازی شده (شامل لایه کاربرد، لایه سیستم عامل، مدیریت ماشین مجازی و منابع فیزیکی) می‌تواند جداسازی کارکرد ماشین‌های مجازی را با مشکل مواجه سازد [۴]. چهارم، با توجه به وابستگی عملیاتی بین لایه‌های کاربردهای چند لایه‌ای، افت کارایی یک لایه می‌تواند به لایه‌های وابسته سرایت کند و در نهایت کارایی کلی کاربرد چند لایه‌ای را کاهش دهد [۵، ۶، ۳۱]. مهم‌تر اینکه، پیاده‌سازی این کاربردها در بستر اشتراکی مجازی شده، می‌تواند این مشکل را بغرنج‌تر کند.

سرانجام، مکان‌یابی نامناسب ماشین‌های مجازی میزبان لایه‌های کاربردهای چندلایه‌ای نه تنها منجر به افزایش تداخل کارکرد عملیاتی کاربردها می‌شود، بلکه موجب افزایش شمار مهاجرت‌های پی‌درپی ماشین‌های مجازی می‌شود. از سوی دیگر، با افزایش پویایی بارهای کاری در بستر مرکز داده، مکان‌یابی نادرست به افزایش اثر پینگ پونگی منجر می‌شود. در این وضعیت، برخی از ماشین‌های مجازی بیشتر بین میزبانان فیزیکی جابجایی می‌شوند و کمتر درگیر پاسخ‌دهی درخواست‌های کاربران می‌باشند. در نهایت سیستم به سوی کوبیدگی و افت زیاد کارایی خواهد رفت.

رابطه (۲) فرم ریاضی تابع تولید تولید کب-داگلاس را نشان می‌دهد.

$$Q = X_0 \prod_{i=1}^m A_i^{x_i} \quad (2)$$

در رابطه (۲)،  $Q$  نشان دهنده بیشینه خروجی (تولید) به ازای ورودی‌های  $A_1, A_2, \dots, A_m$  و  $Q_j$  نشان دهنده خروجی مشاهده شده در بخش  $j$ ام با ورودی‌های  $A_{1j}, A_{2j}, \dots, A_{mj}$  است. از آنجایی که  $Q$  بیشترین خروجی است، پس رابطه (۳) برقرار است.

$$Q_j \leq X_0 \prod_{i=1}^m A_{ij}^{x_i} \quad (3)$$

با لگاریتم‌گیری طرفین رابطه (۳) خواهیم داشت:

$$\ln Q_j \leq \ln(X_0 \prod_{i=1}^m A_{ij}^{x_i}) \quad (4)$$

$$\ln Q_j \leq \ln X_0 + \sum_{i=1}^m x_i \ln A_{ij} \quad (5)$$

با قرار دادن  $\ln Q_j = q_j, x_0 = \ln X_0, a_{ij} = \ln A_{ij}$  بازنویسی رابطه (۳) به صورت رابطه (۶) خواهد بود:

$$q_j \leq x_0 + \sum_{i=1}^m x_i a_{ij} \quad (6)$$

حال مساله تخمین پارامترها در مدل اولیه را می‌توان به صورت زیر بیان کرد:

مشخص کردن  $x_m, \dots, x_1, x_0$  چنان که اختلاف بین مقادیر واقعی و مقادیر بدست آمده از تابع تولید را کمینه و امکان تولید  $x_0 + \sum_{i=1}^m x_i a_{ij}$  را بیشینه سازد. بنابراین مساله اولیه به مساله بهینه‌سازی خطی زیر تبدیل خواهد شد:

$$\text{maximize } z = \sum_{j=1}^n (x_0 + \sum_{i=1}^m x_i a_{ij} - q_j) \quad (7)$$

$$\text{s.t. } \begin{aligned} x_0 + \sum_{i=1}^m x_i a_{ij} &\geq q_j & j = 1, 2, \dots, n \\ x_i &\geq 0 & i = 1, 2, \dots, m \end{aligned}$$

بنابراین کارآیی واحد  $j$ ام به صورت زیر تعریف می‌شود:

$$\text{efficiency} = \frac{Q_j}{Q_j^*} = \frac{Q_j}{X_0 \prod_{i=1}^m A_{ij}^{x_i}} \quad (8)$$

در رابطه (۸)،  $Q_j$  مقدار خروجی واحد  $j$ ام و  $Q_j^*$  مقدار خروجی بهینه واحد  $j$ ام است. در اینجا برای روشن شدن چگونگی کارکرد این روش، نمونه‌ای براساس جدول (۱) ارائه می‌شود.

تخصیص خودکار منابع و گزینش زیرساخت ابر مناسب را براساس محدودیت‌های منبع، هزینه و توافق سطح سرویس ارائه داده‌اند. ما در [۱۹] ارزیابی کارایی Mediawiki [۲۶] را به عنوان کاربرد دولایه‌ای انجام داده‌ایم. پارامترهای کمی و کیفی وابسته به دو لایه وب و پایگاه داده مورد بررسی قرار گرفته‌اند. هرچند هم‌نشینی لایه‌ها بررسی نشده است.

در [۳۱] اثر هم‌نشینی لایه‌های گوناگون دو MTA به نام Xen و MediaWiki و RUBiS VA را در بستر مجازی‌ساز Xen مدل‌سازی تحلیلی و ارزیابی کمی و کیفی نمودیم. دست آوردها، هم‌نشینی لایه‌های VMهای را که از نظر نیازمندی و بدست‌گیری منابع با هم رقابت کمتری داشته و تداخل کارکرد عملیاتی کمتری روی میزبان فیزیکی دارند را پیشنهاد می‌کند. در [۲۰] ارزیابی بارهای انفجاری و تاثیر آن بر در دسترس بودن سرویس، زمان پاسخ و سطح توافق سرویس MTAها مورد بررسی قرار گرفته است.

با توجه به بررسی‌های انجام شده، اثر هم‌نشینی لایه‌های متفاوت MTAها و مکان‌یابی آنان کمتر در پژوهش‌ها مورد بررسی قرار گرفته است. ما در این پژوهش این به بررسی اثر هم‌نشینی MTAها و مکان‌یابی آنان در بستر مجازی شده می‌پردازیم.

### ۳- تابع تولید کب-داگلاس

در این بخش نخست مقدماتی در ارتباط با تابع تولید کب-داگلاس [۷] بیان می‌شود. از این تابع تولید برای رتبه‌بندی کارکرد ماشین‌های مجازی بهره خواهیم برد.

تعریف ۱: تابع تولید نشان دهنده بیشترین میزان تولید (خروجی) به ازای ترکیب‌های مختلف منابع تولید (ورودی) است. اگر فرض کنیم فقط دو نهاد (ورودی) نیروی کار ( $L$ ) و سرمایه ( $K$ ) داشته باشیم، تابع تولید به صورت زیر است:

$$Q = AL^\alpha K^\beta, \quad \alpha + \beta = 1 \quad (1)$$

$A$ : ضریب فنی یا تکنیکی تولید (که ثابت فرض می‌شود)

$\alpha$ : کشش تولید<sup>۱</sup> نهاد نیروی کار  $\alpha \in (0, 1)$

$\beta$ : کشش تولید نهاد سرمایه  $\beta \in (0, 1)$

کشش تولید نشان دهنده میزان تغییرات خروجی نسبت به تغییر در ورودی است. برای نمونه اگر  $\alpha = 0.35$  باشد، یک درصد افزایش در مقدار نهاد نیروی کار موجب افزایش تقریبی ۳۵ درصدی تولید می‌شود.

<sup>1</sup> Output elasticity

که پژوهش‌های انجام گرفته در زمینه مکان‌یابی ماشین‌های مجازی تنها پردازشگر را به عنوان پارامتر سنجش اولویت مهاجرت در نظر گرفته‌اند. هدف ما، استفاده از دیگر منابع سیستمی همانند دیسک و حافظه اصلی برای شناخت ماشین مجازی اولویت‌دار برای مکان‌یابی است.

لازم به یادآوری است رتبه‌هایی که براساس مدل‌سازی کب-داگلاس بدست می‌آید، می‌بایست براساس دیدگاه ما در رتبه‌بندی ماشین‌های مجازی هم‌خوانی داشته باشد. همان‌گونه که گفته شد زمان پاسخ هر ماشین مجازی به عنوان خروجی سیستم با توجه به مقادیر بهره‌وری منابع، به عنوان ورودی، در نظر گرفته می‌شود. هر چه که میزان زمان پاسخ در مدل مبتنی بر کب-داگلاس کمتر شود آنگاه سیستم کارآمدتر است. بنابراین مدل بهینه‌سازی شده را به صورت زیر بازنویسی می‌کنیم. مدل تابع تولید ماشین‌های مجازی را به صورت زیر بیان می‌کنیم:

$$R_{i,j} = \gamma \prod_{i=1}^{\Omega_j} A_{res,i,j}^{u_{res,i,j}} \begin{cases} i = 1, 2, \dots, |PM^j|, j = 1, 2, \dots, \Omega \\ res \in \Psi \end{cases} \quad (9)$$

در این رابطه  $R_{i,j}$ ، زمان پاسخ ماشین مجازی  $i$ ام روی ماشین فیزیکی  $j$ ام ( $PM^j$ ) است،  $\Omega$  مجموعه ماشین‌های مجازی روی  $PM^j$  است که منبع (منابع) آن در وضعیت بهره‌وری<sup>1</sup> بالاست.  $\Psi$  و  $\gamma$  به ترتیب مجموعه منابع سیستمی و ضریب فنی در تابع تولید کب-داگلاس می‌باشند.  $U_{res,i,j}$  نیز بهره‌وری منبع  $res$  ماشین مجازی  $i$ ام روی  $PM^j$  است. با توجه به مدل‌سازی در رابطه (۷)، مدل بهینه‌سازی ماشین مجازی  $i$ ام روی ماشین فیزیکی  $j$ ام ( $PM^j$ ) به صورت رابطه (۱۰) زیر بازنویسی می‌شود.

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^{\Omega_j} (\gamma + \sum_{res \in \Psi} u_{res,i,j} a_{res,i,j} - R_{i,j}) \\ & \text{s.t.} \quad \gamma + \sum_{res \in \Psi} u_{res,i,j} a_{res,i,j} \leq R_{i,j} \quad j = 1, 2, \dots, \Omega^j, \forall M_{i,j} \in \Omega^j \\ & \quad \quad u_{res,i,j} \geq 0 \quad res \in \Psi \end{aligned} \quad (10)$$

دست آورد بهینه‌سازی بالا، رتبه‌بندی ماشین‌های مجازی روی ماشین فیزیکی  $j$ ام است تا ماشین‌های مجازی با کارکرد نامناسب به مجموعه مهاجرت شونده ( $M$ ) افزوده می‌شوند. روش پیشنهادی از دیدگاه پیچیدگی زمانی کارکرد مناسبی دارد. زیرا که، اگر شمار متغیرها در یک برنامه ریزی خطی با  $n$  محدودیت ثابت باشد، برنامه ریزی خطی دارای پیچیدگی زمانی  $O(n)$  است [۲۱]. مدل‌سازی مبتنی بر برنامه‌ریزی خطی پیشنهادی دارای  $\Omega \times \Psi$

جدول ۱: سیستم با دو ورودی و یک خروجی

سیستم	$A_1$	$A_2$	$Q$
۱	۵	۱۰	۷
۲	۱۰	۱۵	۸
۳	۲۰	۱۰	۱۶
۴	۳۰	۲۰	۱۲

در این نمونه، چهار سیستم با دو ورودی ( $A_1, A_2$ ) و یک خروجی ( $Q$ ) است را تحلیل می‌کنیم. پس از اعمال رابطه (۷) به جدول بالا، مساله بهینه‌سازی همانند زیر خواهد شد:

$$\begin{aligned} & \text{maximize} \quad z = 4x_0 + 10.28x_1 + 10.29x_2 - 9.94 \quad (7) \\ & \text{s.t.} \quad x_0 + 1.6x_1 + 2.3x_2 \geq 1.95 \\ & \quad \quad x_0 + 2.3x_1 + 2.7x_2 \geq 2.08 \\ & \quad \quad x_0 + 2.99x_1 + 2.3x_2 \geq 2.77 \\ & \quad \quad x_0 + 3.4x_1 + 2.99x_2 \geq 2.48 \\ & \quad \quad x_0, x_1, x_2 \geq 0 \end{aligned}$$

با حل مساله بهینه‌سازی بالا مقدار بهینه واحدها از رابطه زیر محاسبه می‌شود:

$$Q^* = X_0 A_1^{X_1} A_2^{X_2} \Rightarrow Q^* = 2A_1^{0.3} A_2^{0.6} \quad (8)$$

جدول (۲) نشان می‌دهد که سیستم ۳ دارای رتبه نخست کارکرد و سیستم ۲ رتبه آخر کارکردی را کسب کرده‌اند.

جدول ۲: کارایی و رتبه بندی سیستم

رتبه	کارایی	$Q^*$	$Q$	سیستم
۳	۰.۵۴	۱۳	۷	۱
۴	۰.۴	۲۰	۸	۲
۱	۰.۸	۱۹	۱۶	۳
۲	۰.۷۱	۱۷	۱۲	۴

#### ۴- رتبه‌بندی ماشین‌های مجازی

در این بخش، رتبه‌بندی ماشین‌های مجازی را با توجه به کارکردشان ارائه می‌دهیم تا از رتبه‌بندی آن‌ها در فاز مکان‌یابی استفاده کنیم. از آنجا که در این پژوهش فرض براین است که هر ماشین مجازی تنها یک لایه از یک MTA را اجرا (میزبانی) می‌کند، در این نوشتار، واژگان ماشین مجازی و لایه را جای هم استفاده می‌کنیم. از میزان بهره‌وری منابعی که در اختیار هر ماشین مجازی به عنوان ورودی‌های و میزان زمان پاسخ سرویس به عنوان خروجی تابع رتبه‌بندی استفاده خواهیم نمود. در این راه، از تابع تولید کب-داگلاس در ارزیابی واحدها (ماشین‌های مجازی) که در بخش پیش بیان شد، بهره می‌بریم. لازم به یادآوری است

<sup>1</sup> Utilization

در این رابطه،  $\overline{residual}_{res, PM_j}$  و  $\overline{demand}_{res, VM_i}$  به ترتیب بردار میزان درخواست (تقاضای) منابع<sup>۱</sup> ماشین مجازی  $\Lambda$  در مجموعه  $M$  و بردار منابع آزاد ماشین میزبان  $\Lambda$  در مجموعه  $\Lambda$  می‌باشند. چنانچه شرط بالا درست نباشد، می‌توان شماری از ماشین‌های فیزیکی را از مجموعه ماشین‌های فیزیکی آماده خدمت<sup>۲</sup> ( $\Theta$ ) به مجموعه ماشین‌های میزبان مقصد ( $\Lambda$ ) تا برقراری شرط بالا، افزود. آزادسازی منابع تا برقراری شرط (۱۳) در الگوریتم (۱) نشان داده شده است. همچنین، عملگر نامساوی ( $\leq$ ) برای مقایسه دو بردار مورد استفاده قرار می‌گیرد.

تعریف ۲: اگر هر درایه از بردار  $x$  کوچکتر یا مساوی درایه متناظرش در بردار  $y$  باشد.

برای هر درخواست فرض براین است که فراهم‌کننده سرویس برای تضمین کیفیت سرویس مقداری بیش از درخواست کاربر  $\eta \in (0, 1]$  تخصیص می‌دهد. این مقدار وابسته به مواردی همچون تحمل‌پذیری در برابر خطا و سیاست هزینه‌ای فراهم‌کننده سرویس دارد. چگونی تخمین این مقدار، به بهینه‌سازی براساس سیاست‌های تخصیص منابع، هزینه، نوع بار کاری بستگی دارد که آن را برای کارهای آینده در نظر می‌گیریم. مقدار  $\eta$  را در آزمایش‌های انجام گرفته صفر در نظر گرفته‌ایم تا کارکرد روش پیشنهادی در حالت واقعی سیستم (بدترین حالت) را ارزیابی شود.

الگوریتم ۱: آزادسازی منابع مورد نیاز برای مکان یابی

```

1. Input: PM, VM,  $\Lambda$ , and  $\Theta$  sets and VM resources demand
2. Output:  $\Lambda$  with adequate resources to initiate migration phase
3. Flag  $\leftarrow$  false
4.  $\bar{r} \leftarrow \sum_{j=1}^{|\Lambda|} \overline{residual}_{PM_j} // PM_j \in \Lambda$ 
5.  $\bar{d} \leftarrow \sum_{i=1}^{|M|} \overline{demand}_{VM_i} // VM_i \in M$ 
6. while (Flag == false)
7.   for ( $k \leftarrow 1$  to  $|\Theta|$ )
8.      $\Theta \leftarrow \Theta - PM_k$ 
9.      $\Lambda \leftarrow \Lambda \cup PM_k$ 
10.     $\bar{r} \leftarrow \bar{r} + \overline{residual}_{PM_k}$ 
11.    if ( $\bar{d} \leq (\bar{r} + \bar{1}) \times \eta$ )
12.      flag  $\leftarrow$  true
13.      break
14.    end if
15.  end for
16. end while
17. return
    
```

<sup>1</sup> Resource demands

<sup>2</sup> Standby

متغیر است. بنابراین، پیچیدگی زمانی برای محاسبه رتبه‌بندی کارایی ماشین‌های مجازی برابر  $O(\Omega \times \Psi)$  است.

## ۵- مکان‌یابی ماشین‌های مجازی VMTAها در مرکز داده

تا اینجا، با بهره‌گیری از تابع تولید کب-داگلاس کارایی ماشین‌های مجازی رتبه‌بندی شده و ماشین‌های مجازی ناکارا به مجموعه ماشین‌های مهاجر ( $M$ ) افزوده می‌شوند. در این بخش با توجه به مدل‌سازی که در بخش‌های پیشین انجام شد، به ارائه راهبردی برای مکان‌یابی ماشین‌های مجازی خواهیم پرداخت. در این راستا، نخست الگوریتمی برای آزادسازی منابع مورد نیاز پیش از انجام مکان‌یابی ارائه می‌شود. سپس الگوریتم اولویت‌بندی VMTAها را برای مکان‌یابی بیان خواهیم کرد. همچنین، الگوریتمی برای زمان‌بندی مکان‌دهی لایه‌های VMTAها ارائه خواهد شد و در نهایت نمونه‌ای آزمایشگاهی برای هم‌نشینی لایه‌های VMTAها، ارائه می‌شود.

### ۵-۱- آزادسازی منابع

در مکان‌یابی ماشین‌های مجازی، میزان منابع موجود در مرکز داده بایستی پیش از انجام امکان‌سنجی شود. در این راستا، الگوریتمی پیشنهاد می‌شود. پیش از آن مقدماتی ارائه می‌شود. الگوریتم مکان‌یابی نگاشتی است از  $M \rightarrow \Lambda$  چنانچه:

$$\forall VM_j \in M, \exists PM_i \in \Lambda, x_{ij} = 1 \quad (11)$$

$$\Rightarrow \sum_{k=1}^P x_{kj} = 1, k=1, 2, \dots, P$$

در این رابطه،  $P$  شمار همه ماشین‌های فیزیکی در مرکز داده،  $M$  مجموعه ماشین‌های مجازی مهاجر و  $\Lambda$  مجموعه ماشین‌های فیزیکی مقصد است. شرط بالا تضمین می‌کند که هر ماشین مجازی پس از مهاجرت تنها روی یک ماشین فیزیکی جایابی می‌شود.  $x$  یک متغیر دودویی است که بودن یا نبودن یک ماشین مجازی را در یک ماشین فیزیکی نشان می‌دهد. این شرط را می‌توان به عنوان یک قید به صورت زیر نوشت.

$$\sum_{k=1}^{|\Lambda|} x_{kj} = 1 \quad (12)$$

برای اینکه همه ماشین‌های مجازی مکان‌یابی شوند، فرض بر این است که ماشین‌های فیزیکی به اندازه کافی منابع آزاد دارند. در رابطه (۱۳)، امکان وجود منابع مورد نیاز ماشین‌های مجازی پیش از مکان‌یابی به عنوان یکی از قیود باید بررسی شود.

$$\sum_{i=1}^{|\Lambda|} \overline{demand}_{res, VM_i} < \sum_{j=1}^{|\Lambda|} \overline{residual}_{res, PM_j} \quad (13)$$

$$R_{VMTA_k} = \sum_{t=1}^l R_{VMTA_k, tier t} \quad (16)$$

VMTAها براساس رابطه (۱۵) اولویت‌بندی مکان‌یابی می‌شوند و لایه‌هایی از آنان که در مجموعه ماشین‌های مهاجر می‌باشند، در فاز بعدی روی میزبانان فیزیکی در مجموعه  $\Lambda$  مکان‌یابی می‌شوند. روشن است هر چه زمان پاسخ  $VMTA_K$  به سطح سرویس توافقی نزدیک‌تر باشد یا از آن بیشتر باشد، لایه‌های  $VMTA_K$  دارای وضعیت کارایی نامناسبی می‌باشند و باید با توجه به اولویت‌شان جایابی شوند.

### ۵-۳- الگوریتم زمان‌بندی VMTAها

همه لایه‌های (ماشین‌های مجازی) گزینش‌شده برای مهاجرت در گام پیشین و ماشین‌های مجازی تازه تدارک داده شده، مجموعه مهاجرتی  $M$  را می‌سازند. مدیریت سراسری ( $GM^2$ ) براساس کارکرد یک لایه و وابستگی بین لایه‌ها در  $M$ ، ماشین‌های مجازی برای مکان‌یابی زمان‌بندی می‌کند. الگوریتم (۲) زمان‌بندی مهاجرت لایه‌های VMTAها را نشان می‌دهد. در این الگوریتم پیشنهادی، VMTAها براساس رابطه (۱۵) اولویت‌بندی شده و سپس براساس الگوریتم مرتب‌سازی سریع<sup>۳</sup> مرتب می‌شوند (خط ۳).

#### الگوریتم ۲: زمان‌بندی مکان‌یابی VMTAها

```

1. Input: set  $\Lambda$ , set  $M$ , VMtAs list,
   resources demand vectors of tiers in  $\Lambda$  and  $M$ 
2. Output: Virtual machine migration scheduling
3. Sort (VMtAs of in descending order based on Eq.(15))
4. for ( $i \leftarrow 1$  to  $|\Phi|$ ) //  $\Phi$ : number of VMTA in datacenter
5.   for ( $t \leftarrow 1$  to  $\omega_{VMTA_i}$ )
     //  $\omega_{VMTA_i}$ : number of overloaded tiers of  $VMTA_i$ 
6.     candidateVM  $\leftarrow$  NextOverloadedVM( $VMTA_i$ )
7.     // returns the next overloaded VM of  $VMTA_i$  in set  $M$ 
8.     Placement(candidateVM)
9.   end for
10. end for
11. return
    
```

پیچیدگی زمان اجرایی الگوریتم زمان‌بندی وابسته به شمار VMTAهای در حال اجرا در مرکز داده ( $\Phi$ )، شمار لایه‌های ناکاری VMTA یعنی ( $\omega$ ) و پیچیدگی الگوریتم مکان‌یابی است که در بخش پیش‌رو بیان می‌شود. بخش مرتب‌سازی براساس الگوریتم مرتب‌سازی سریع انجام می‌گیرد، که پیچیدگی آن در بدترین حالت  $O(\Phi \log \Phi)$  است. پس پیچیدگی این الگوریتم

در گام‌های ۴ و ۵ این الگوریتم، میزان منابع آزاد ماشین‌های فیزیکی مقصد ( $\vec{r}$ ) و میزان درخواست منابع ماشین‌های مجازی ( $\vec{d}$ ) محاسبه می‌شوند. سپس، از مجموعه ماشین‌های فیزیکی آماده به خدمت  $\Theta$ ، تا رسیدن به وضعیتی که میزان منابع آزاد ماشین‌های فیزیکی مقصد بیشتر از میزان درخواست‌های ماشین‌های مجازی شود، ماشین فیزیکی به مجموعه  $\Lambda$  افزوده می‌شود (خطوط ۶ تا ۱۶). در خط ۸ نیز ماشین فیزیکی از مجموعه ماشین‌های فیزیکی آماده به خدمت  $\Theta$  حذف شده و به مجموعه  $\Lambda$  افزوده می‌شود (خط ۹).

پیچیدگی زمانی این الگوریتم  $O(\max(\Lambda, M, \Theta))$  است.

### ۵-۲- اولویت‌دهی VMTAها برای مکان‌یابی

در این بخش، VMTAها را براساس میزان تخطی از سطح سرویس توافقی اولویت‌بندی می‌کنیم. هرچه میزان تخطی از سرویس VMTAی بیشتر باشد، لایه‌هایی از آن که در مجموعه  $M$  هستند می‌بایست زودتر مکان‌دهی شوند. هر VMTA را می‌توان با استفاده از ساختمان داده‌ای همانند لیست پیوندی دو طرفه ذخیره کرد که هر گره آن اطلاعات یک لایه شامل، آدرس ماشین فیزیکی میزبان، میزان بهره‌وری کنونی منابع، بیشینه منابع درخواستی، آدرس لایه بعدی و لایه پیشین را ذخیره می‌کند. همچنین، وضعیت کارکردی VMTA با توجه به اینکه ماشین‌های (های) مجازی وابسته به آن در وضعیت نرمال می‌باشند ذخیره می‌شود. بدین معنی که اگر لایه‌ای از آن به مجموعه  $M$  تعلق داشته باشد فیلد باینری آن گره یک می‌شود. رابطه (۱۴) برای بیان وابستگی بین دو لایه استفاده می‌شود.

$$VM_{p,j}^A \text{ is preceding tier of } VM_{s,k}^A \text{ in } VMTA^A \quad (14)$$

$$\text{iff } (x_{p,j} \times x_{s,k} = 1) \text{ and } (\text{next}(VM_{p,j}^A) == VM_{s,k}^A)$$

رابطه (۱۵) اولویت  $VMTA_K$  در مکان‌یابی لایه‌های آن را محاسبه می‌کند.

$$\text{Priority}^{VMTA_k} = \frac{R_{VMTA_k} - R_{VMTA_k}^{SLA}}{R_{VMTA_k}^{SLA}} \quad (15)$$

در این رابطه  $R_{VMTA_k}^{SLA}$  و  $R_{VMTA_k}$  به ترتیب زمان پاسخ و میزان سطح توافق سرویس  $VMTA_K$  است.  $R_{VMTA_k}$  نیز برابر مجموع جمع<sup>۱</sup> زمان‌های سکونت ( $R_{VMTA_k, tier t}$ ) لایه‌های  $VMTA_K$  است (رابطه ۱۶). در این رابطه  $l$  شمار لایه‌های  $VMTA_K$  است.

<sup>2</sup> Global Manager (GM)

<sup>3</sup> Quick sort

<sup>1</sup> Aggregated residence time

براساس رابطه (۱۷) محاسبه می‌شود.

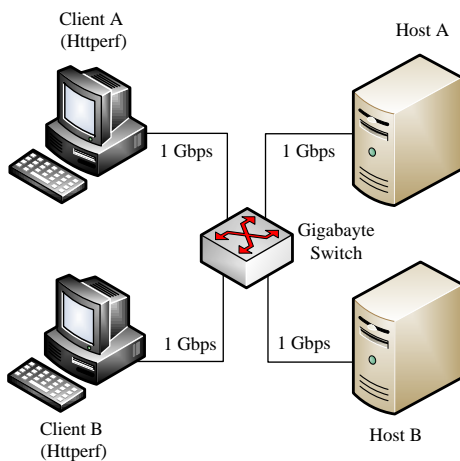
$$O(\Phi \times \omega_{VMTA} \times O(Placement Alg.)) + O(\Phi \log \Phi) \quad (17)$$

با توجه به پیاده‌سازی‌های تجربی، VMTAها غالباً سه لایه‌ای می‌باشند، بنابراین بیشینه تعداد لایه‌های با کارایی نامناسب VMTAها یعنی  $\omega$  برابر ۳ است. لایه‌هایی از VMTA اولویت‌دار توسط الگوریتم مکان‌یابی (۳) جاده می‌شوند. این الگوریتم در بخش ۵.۵ ارائه می‌شود. پیش از آن، به بیان نمونه‌ای از دست‌آوردهای پیاده‌سازی عملی هم‌نشینی لایه‌های کاربردی که بصورت چندلایه‌ای پیکربندی شده می‌پردازیم تا براساس آن الگوریتم مکان‌یابی را پیشنهاد دهیم.

#### ۵-۴- نمونه آزمایشگاهی بررسی هم‌نشینی لایه‌ها

هدف این پژوهش مکان‌یابی ماشین‌های مجازی میزبان لایه‌های VMTAها با توجه به کارکرد آنها و بررسی میزان تداخل کارکرد ماشین‌های مجازی میزبان و ماشین‌های مجازی در حال اجرا روی ماشین فیزیکی است. در این راه، پایه اصولی برای هم‌نشینی لایه‌های VMTAها را با یک نمونه آزمایشگاهی مورد بررسی قرار می‌دهیم. در این بخش، ما آزمایش‌های تجربی را برای یکپارچه‌سازی بار کاری RUBiS [۲۲] برای بررسی هم‌نشینی لایه‌های این دو کاربرد چندلایه‌ای مجازی شده انجام خواهیم داد. پیکربندی سه لایه‌ای شامل Apache، Tomcat و MySQL برای RUBiS مورد استفاده قرار خواهد گرفت. RUBiSVA امکان پیاده‌سازی RUBiS را در بستر Xen فراهم نموده و کارکرد سایت خرید و فروش eBay را فراهم می‌کند [۲۲]. سه نمونه کاربران، نه جدول در پایگاه داده و ۲۶ نوع تراکنش از طریق مرورگر وب قابل انجام است. از تولید کننده بار Httpperf [۲۳] از سوی دو کلاینت A و B به ترتیب برای دو سرور فیزیکی A و B که روی آنها RUBiSVA پیکربندی شده بار کاری با توزیع پواسن فرستاده می‌شود (شکل ۱) [۱].

در پیکربندی VMTAها، هر لایه روی یک ماشین مجازی که دارای یک هسته پردازشی است، پیکربندی شده و دیسک هم بین آنان اشتراک گذاشته شده است. پیکربندی سخت‌افزاری و نرم‌افزاری به ترتیب در جدول‌های (۳) و (۴) نشان داده شده است.



شکل ۱: پیکربندی ارتباطی بستر آزمایشگاهی

جدول ۳: مشخصات سخت‌افزاری

Server machines	HP ProLiant DL580
Processor	2.67GHz, 24MB cache
# of Cores	8
Main Memory	64GB (8x8GB) DDR3
Disk	2.4TB, SAS, 7200rpm, 6Gbps
NIC	1 Gbps
Switch	Gigabyte Cisco (SRW2024P)
Client machines	Intel i7, 2.2 GHz, 6MHz cache

جدول ۴: مشخصات نرم‌افزاری

OS	Fedora Core 19
Xen hypervisor	ver. 4.2.2
Apache httpd	ver. 2.4.6
PHP	ver. 5.4.0
Apache Tomcat	ver. 7.0.56
Java Servlet	ver. 3.1
MySQL	ver. 5.6
Httpperf	ver. 0.9.0

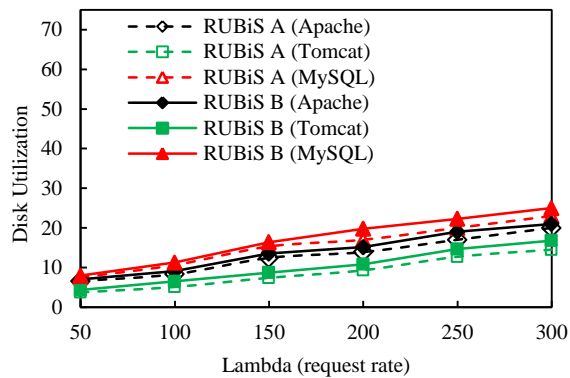
پیکربندی ماشین‌های مجازی مربوط به لایه‌های دو نمونه RUBiS VA در جدول (۵) نشان داده شده است. میزبان A دارای پنج ماشین مجازی و میزبان B دارای سه ماشین مجازی است. ماشین‌های مجازی چهارم و پنجم میزبان A با توجه به سناریوی مورد بررسی روشن می‌شوند و ماشین مجازی متناظر (یک و سه) در میزبان B خاموش می‌شود. برای نمونه در سناریوی نخست که به بررسی اثر هم‌نشینی لایه‌های وب (سرورهای Apache) می‌پردازیم، ماشین مجازی میزبان وب سرور روی میزبان B (VM 1) خاموش و ماشین مجازی وب سرور روی میزبان A (VM 4) روشن می‌شود و به درخواست‌های وب مربوط به کلاینت B پاسخ می‌دهد. در سناریوی دوم نیز که به بررسی هم‌نشینی لایه‌های پایگاه داده (سرورهای MySQL) می‌پردازیم، ماشین مجازی میزبان پایگاه داده روی میزبان B (VM 3) خاموش شده و ماشین

<sup>1</sup> Figure

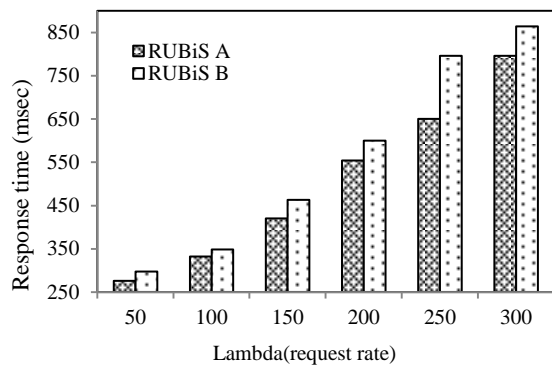
مجازی میزبان پایگاه داده روی میزبان A، (VM 5) روشن می‌شود.

جدول ۵: سناریوهای آزمایشگاهی

Scen.	Host A					Host B		
	VM 1	VM 2	VM 3	VM 4	VM 5	VM 1	VM 2	VM 3
1	Apache	Tomcat	MySQL	Apache	OFF	OFF	Tomcat	MySQL
2	Apache	Tomcat	MySQL	OFF	MySQL	Apache	Tomcat	OFF



شکل ۳: سناریوی (۱) بهره‌وری دیسک



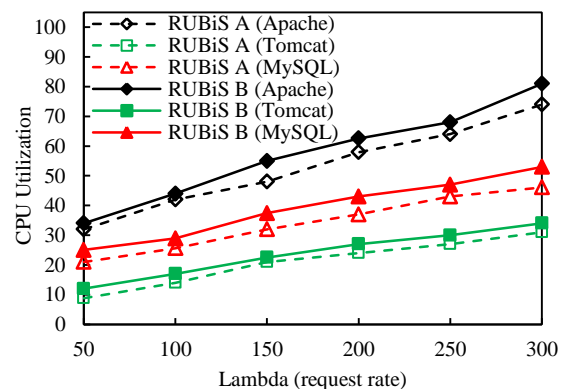
شکل ۴: سناریوی (۱) زمان پاسخ درخواست

از آنجا که هر ماشین مجازی یک سرویس مشخص را میزبانی می‌کند، اجرای سرویس مورد نظر روی ماشین فیزیکی نیازمند بهره‌وری منابع آن است. چنانچه دو ماشین مجازی که هر دو به زمان پردازش زیادی روی پردازشگر یا دیسک نیاز دارند روی یک میزبان فیزیکی هم‌نشین شوند، میزان زمان پاسخ سرویس‌های متناظر نسبت به حالتی که ترکیبی از سرورهای پردازش-محور و دیسک-محور داشته باشیم، بیشتر خواهد بود. بنابراین تا حد امکان می‌بایست سرویس‌هایی که منابع یکسانی را بیشتر بهره‌ور می‌سازند، روی میزبانان فیزیکی متفاوتی باشند تا تداخل کارکردی آنان کمتر و در نهایت زمان پاسخ کمتر شود.

با توجه به این دست‌آوردها، در بخش بعدی الگوریتم مکان‌دهی لایه‌های VMTAهایی که در مجموعه ماشین‌های مهاجر هستند را با در نظر گرفتن کارکرد لایه‌ها، منابع درخواستی و میزان بهره‌وری منابع ماشین‌های میزبان در راستای کاهش تداخل کارکرد ماشین‌های مجازی و ترازمندی بار در مرکز داده ارائه می‌دهیم.

نتایج هم‌نشینی دو VMTA در دو سناریوی ۱ و سناریوی ۲ در شکل‌های (۲) تا (۷) نشان داده شده است. همان‌گونه که می‌بینیم، هنگامی که لایه‌های وب در سناریوی نخست در کنار هم روی ماشین فیزیکی A می‌باشند، میزان مصرف منابع به ویژه پردازشگر بسیار زیاد می‌شود. هر چند که ماشین‌های مجازی آنان جدا از هم می‌باشد و دارای هسته‌های پردازشی جداگانه‌ای می‌باشند، این دو ماشین مجازی در بدست‌گیری پردازشگر و حافظه نهان با هم رقابت داشته که موجب افزایش بهره‌وری پردازشگر می‌شود. همچنین زمانی که لایه‌های پایگاه داده در کنار هم می‌باشند و روی ماشین فیزیکی A قرار دارند، میزان منابع مصرفی به ویژه دیسک ماشین فیزیکی A نسبت به سناریوی نخست افزایش زیادی یافته است. دلیل این افزایش رقابت در بدست‌گیری راه‌اندازهای دیسک و دسترسی فیزیکی به دیسک توسط ماشین‌های مجازی است. در حقیقت، لایه‌هایی که از نظر کارکرد همانند می‌باشند، روی منابع یکسان رقابت دارند. در نرخ بارهای کاری بیشتر رقابت بیشتر شده و بهره‌وری منابع رقابتی بیشتر و در نهایت زمان پاسخ افزایش می‌یابد.

همان‌گونه که می‌بینیم زمان پاسخ در سناریوی دوم نسبت به سناریوی نخست بیشتر است. این نتایج نشان می‌دهد که هم‌نشینی ماشین‌های مجازی که روی منابع یکسان رقابت دارند، موجب افزایش زمان انتظار در بدست‌گیری آن منبع شده و در نتیجه زمان پاسخ درخواست‌ها زیاد می‌شود. همچنین هم‌نشینی ماشین‌های مجازی که میزبان پایگاه داده می‌باشند با افزایش بیشتر بهره‌وری دیسک و در نتیجه زمان پاسخ همراه است.

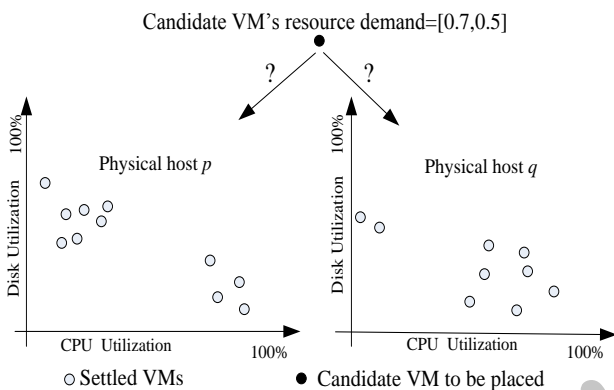


شکل ۲: سناریوی (۱) بهره‌وری پردازشگر

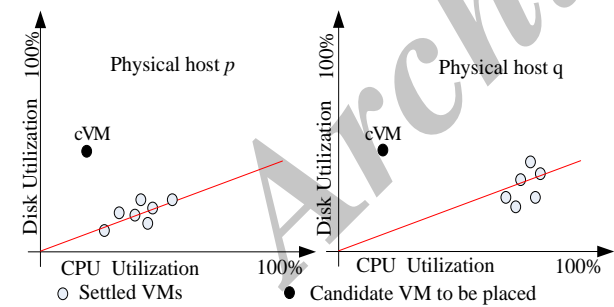


ماشین فیزیکی مقصد وجود دارند، تداخل کارکردی ماشین مجازی کاندید (cVM) را با ماشین‌های مجازی ( $VM_{j,k}$ ) که اکنون روی میزبانان فیزیکی مقصد (مجموعه  $\Lambda$ ) در حال اجرا می‌باشند را محاسبه خواهیم نمود. هر چه این زاویه بیشتر باشد، میزان فیزیکی برای جا دادن این ماشین مجازی مناسب‌تر است.

برای روشن شدن این ایده، نمونه‌ای از امکان جایابی بردار ماشین‌های مجازی روی دو ماشین فیزیکی در شکل (۸) نشان داده شده است. برای سادگی نمایش، این شکل دارای دو بعد شامل بهره‌وری پردازشگر روی محور افقی و بهره‌وری دیسک روی محور عمودی است. راهبرد مکان‌یابی ماشین‌های مجازی میزان لایه‌های VMTAها را با هدف کاهش تداخل کارکرد و ترازمندی بار در مرکز داده جایابی می‌کند.

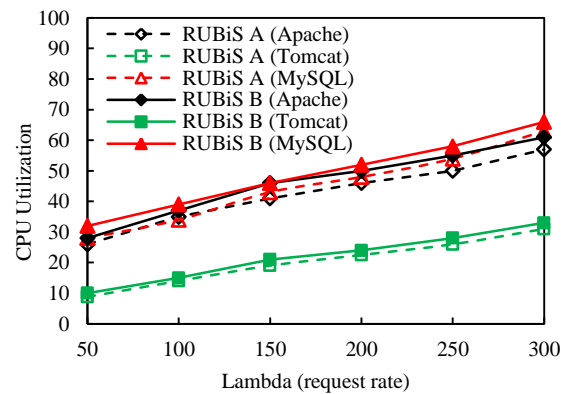


شکل ۸: نمونه‌ای از امکان جایابی ماشین مجازی روی دو میزبان فیزیکی

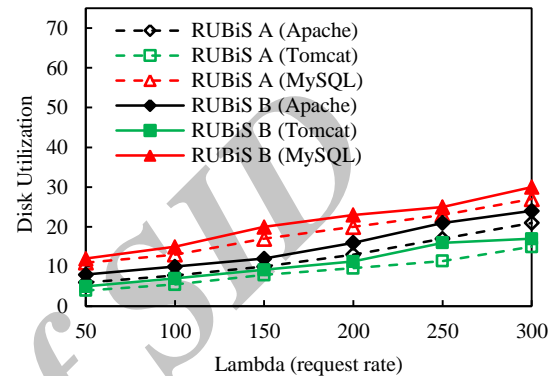


شکل ۹: سناریوهای مکان‌یابی

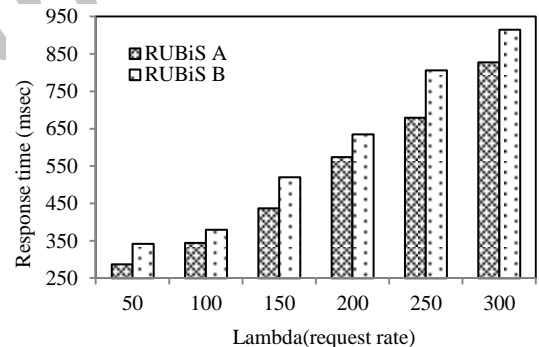
براین اساس، الگوریتم (۳) مکان‌یابی ماشین مجازی را روی مجموعه ماشین‌های فیزیکی که عضو  $\Lambda$  می‌باشند را انجام می‌دهد. نخست، این الگوریتم با توجه به میزان تقاضای ماشین مجازی کاندید (cVM)، همه ماشین‌های فیزیکی که عضو  $\Lambda$  می‌باشند را از نظر داشتن منابع کافی بررسی می‌کند (خط ۷). سپس، میانگین اختلاف زاویه با ماشین‌های مجازی و همچنین میانگین مجموع فاصله اقلیدیس با ماشین‌های مجازی که روی ماشین فیزیکی  $\Lambda$



شکل ۵: سناریوی (۲) بهره‌وری پردازشگر



شکل ۶: سناریوی (۲) بهره‌وری دیسک



شکل ۷: سناریوی (۲) زمان پاسخ درخواست

#### ۵-۵- الگوریتم مکان‌یابی آگاه به بار کاری VMTAها

الگوریتم‌هایی که برای مکان‌یابی ماشین‌های مجازی ارائه شده‌اند، بیشتر مکان‌یابی را همانند مساله bin-packing یا روش‌هایی اکتشافی مانند First Fit، Best Fit و... انجام می‌دهند [۲۴، ۲۹]. یادآور می‌شویم که نگاشت مجموعه ماشین‌های مجازی به مجموعه‌ای از ماشین‌های فیزیکی میزبان مساله‌ای NP-complete است. ما در پژوهش برای ارزیابی و بررسی تداخل کارکرد ماشین‌های مجازی به صورت زیر عمل می‌کنیم.

بردار میزان تقاضای ماشین مجازی  $\Lambda$  را با بردار  $\overline{Demand}_i = (D_{CPU,i}, D_{disk,i}, D_{memory,i})$  نشان می‌دهیم. با استفاده از زاویه بین بردار این ماشین مجازی و ماشین‌های مجازی که روی

می‌باشند، محاسبه می‌شود (خطوط ۸ تا ۱۵).

در صورتی که میانگین اختلاف زاویه‌ای از مقدار پیشین بیشتر باشد (خطوط ۱۶ تا ۱۹)، ماشین فیزیکی مورد بررسی به عنوان میزبان برگزیده می‌شود (خطوط ۱۶ تا ۱۹). اگر اختلاف زاویه‌ای با مقدار پیشین تقریباً برابر باشد، میانگین فاصله اقلیدسی cVM با ماشین فیزیکی کنونی از مقدار ماشین فیزیکی برگزیده پیشین مقایسه می‌شود (خطوط ۲۰ تا ۲۶). اگر میانگین فاصله اقلیدسی بیشتر از مقدار پیشین باشد، ماشین فیزیکی کنونی به عنوان میزبان برگزیده می‌شود. این فرایند برای همه ماشین‌های فیزیکی در مجموعه M انجام می‌شود. نمونه‌ای از این سناریو در شکل (۹) نشان داده شده‌است. با توجه به روش گزینش ماشین فیزیکی، الگوریتم پیشنهادی ماشین فیزیکی  $q$  را برای ماشین مجازی کاندید برمی‌گزیند.

هرچند که چنین شرایطی با توجه به نحوه گزینش ماشین فیزیکی مقصد با احتمال کم پیش می‌آید، ولی در این الگوریتم این سناریو نیز در نظر گرفته شده است تا مناسب‌ترین میزبان از نظر کم کردن تداخل کارکردی برگزیده شود.

پیچیدگی زمانی اجرای الگوریتم (۳) به شمار ماشین‌های فیزیکی میزبان مناسب ( $\Lambda$ ) و شمار ماشین‌های مجازی قابل پشتیبانی توسط فناوری مجازی‌سازی ( $V$ ) وابسته است. برای نمونه بیشینه ماشین‌های مجازی ویندوزی و لینوکسی قابل پشتیبانی توسط Xenserver به ترتیب ۵۰۰ و ۶۵۰ است [۲۵]. با توجه به اینکه بیشینه مقدار  $|VM|$  در خط دهم الگوریتم  $V$  است، بنابراین، پیچیدگی این الگوریتم  $O(\Lambda \times V)$  است.

## ۶- ارزیابی کارایی

در این بخش به ارزیابی کارایی راهبرد پیشنهادی در مکان‌یابی VMTAها می‌پردازیم. نخست، بستر و پیش‌نیازهای ارزیابی و پارامترهای کارایی گفته خواهد شد، سپس دست‌آوردهای کمی و کیفی داوری می‌شوند. در این راه، از CloudsimEx [۱۸] که نمونه توسعه یافته‌ای از شبیه‌ساز CloudSim است برای ارزیابی مدل استفاده می‌شود.

CloudsimEx توانمندی‌های چارچوب CloudSim را داشته و مدل‌سازی و همانندسازی زیرساخت محاسبات ابر و خدمات وابسته را فراهم می‌کند. همچنین، عملیات ورودی-خروجی را برای زمان‌بندی دیسک ماشین‌های میزبان بین ماشین‌های مجازی همانندسازی می‌کند. سیستم کامپیوتری با ویژگی‌های بخش پنجم برای اجرای همانندسازی سناریوهای روش مکان‌یابی

پیشنهادی استفاده می‌شود. از نمونه‌های ردیابی بار کاری<sup>۱</sup> Wikipedia [۲۷] و RUBiS VA که نمونه‌ای از کاربردهای دو لایه‌ای و سه لایه‌ای می‌باشند در آزمایش‌ها بهره برده می‌شود. مرکز داده مجازی شده دارای ۱۲۰۰ ماشین فیزیکی است که هر یک ویژگی‌های سرور HP ProLiant DL980 G7 با ۸ پردازنده ۶۴ هسته‌ای، دیسک SSD ۷,۲ گیگا بایتی، حافظه اصلی ۲۵۶ گیگابایتی را دارا بوده و در شبکه با پهنای باند ۱۰ گیگا بایت بر ثانیه همانندسازی می‌شود. میزبانان فیزیکی دارای معماری x86 که پیشگر ماشین مجازی Xen و لینوکس برای سیستم عامل آنان پیکربندی شده است. هر میزبان فیزیکی در وضعیت مانای کاری<sup>۲</sup> ۱۲۰۰ وات و در اوج مصرف ۱۴۴۰ وات انرژی مصرفی پیش‌بینی شده‌است.

ماشین‌های مجازی براساس نمونه‌های C3 از Amazon EC2 [۲۸] پیکربندی شده و می‌توانند سرعت پردازشی (برحسب MIPS) که یک مقدار تصادفی از بردار [1000:200:3600]، سرعت عملیات ورودی-خروجی روی دیسک (برحسب IOMIPS) یک مقدار تصادفی از بردار [30000:1000:48000] و مقدار فضای حافظه اصلی که می‌تواند یک مقدار تصادفی از بردار [1.7,2.0,2.5,3.0,3.5,3.75] (برحسب گیگابایت) ماشین فیزیکی را دارا باشند. مقدار IOMIPS براساس IOPS(SSD) volume است که در Amazon EBS [۲۹] برای بارهای کاری پایگاه داده‌ای دارای کارایی ورودی-خروجی بسیار بالا در نظر گرفته شده است. هر ماشین مجازی تنها یک لایه را میزبانی کرده و وابستگی بین لایه‌ها (ماشین‌های مجازی) هر VMTA با استفاده از ساختار داده لیست پیوندی بیان می‌شود.

شمار ماشین‌های مجازی هر ماشین فیزیکی در سناریوها تفاوت دارد. برای نمونه، شمار ماشین‌های مجازی برای هر ماشین فیزیکی از ۴ تا ۶۴ است. پس، بیشینه ۴۸۰۰ ماشین مجازی برای زمانی که ۴ ماشین مجازی روی هر ماشین فیزیکی می‌تواند باشد و بیشینه ۷۶۸۰۰ ماشین مجازی زمانی که ۶۴ ماشین مجازی برای هر ماشین فیزیکی در مرکز داده وجود دارند. همچنین، فرض کرده‌ایم که ۲۰ درصد گنجایش مرکز داده را ماشین‌های مجازی تازه تدارک داده شده می‌باشند، که براساس فرآیند پواسن به مرکز داده وارد می‌شوند [۵، ۲۹، ۳۰]. شمار ماشین‌های مجازی به نسبت بین کاربردهای دو لایه‌ای و سه لایه‌ای بخش‌بندی می‌شود. برای نمونه در سناریوی که ۴ ماشین مجازی می‌تواند روی ماشین فیزیکی پیکربندی شود،  $(80\%) \times (2/5) \times 4800 = 1536$  ماشین

<sup>1</sup> Workload trace

<sup>2</sup> Steady state

کار می‌بریم. الگوریتم Best Fit از الگوریتم‌های تخصیص شناخته شده است که یک ماشین مجازی را روی میزبان مقصدی با کمترین مقدار باقیمانده منابع جا می‌دهد. پارامترهای ارزیابی عبارتند از:

- **زمان پاسخ:** زمان پاسخ VMTA به عنوان یک پارامتر کیفیت سرویس برای ارزیابی کارایی کاربردهاست که آشکارا وابسته به میزان تقاضای منابع و نرخ ورود درخواست‌ها است.
- **تختی از سطح سرویس توافقی:** این پارامتر توافقی بین کاربر نهایی و فراهم کننده سرویس است و بیشتر وابسته به پارامترهایی مانند زمان پاسخ و در دسترس بودن سرویس است. ما این پارامتر را به عنوان میزان انحراف از زمان پاسخ توافقی در نظر می‌گیریم.
- **انرژی مصرفی مرکز داده:** از آنجا که یکی از هدف‌های فراهم کنندگان سرویس، کاهش مصرف انرژی میزبانان فیزیکی مرکز داده است، ما از این پارامتر برای آزمودن اینکه راهبرد پیشنهادی تا چه اندازه شمار میزبانان فیزیکی را در راستای کاهش مصرف انرژی و دستیابی به کیفیت سرویس مورد توافق مفید است، استفاده می‌کنیم.
- **بهره‌وری منابع میزبان فیزیکی:** از این پارامتر برای آزمون چگونگی پخش عادلانه بار کاری بین ماشین‌های فیزیکی مرکز داده استفاده می‌شود.
- **نرخ مصرف انرژی میزبان فیزیکی:** اگر ماشین‌های مجازی بین همه ماشین‌های فیزیکی عادلانه پخش شده باشد، ما انتظار داریم که مصرف انرژی میزبانان فیزیکی نزدیک به هم باشد.

راهبرد مکان‌یابی پیشنهادی که مبتنی بر رتبه‌بندی و آگاه به بار کاری است را  $RWP^1$  می‌نامیم. کارایی راهبرد پیشنهادی را هنگامی که یکی از یکی از فازهای آن در نظر گرفته نشده باشد در دو سناریوی جداگانه ارزیابی می‌کنیم. این سناریوها مکان‌یابی بدون رتبه‌بندی ( $nRP^2$ )، مکان‌یابی ناآگاه از بار کاری ( $nWP^3$ ) نام دارند. الگوریتم تخصیص Best Fit را نیز BF می‌نامیم.

## ۶-۲- ارزیابی دست‌آوردهای راهبردهای مکان‌یابی

مقیاس‌پذیری یکی از هدف‌های اساسی در ارزیابی کارایی روش

مجازی به صورت تصادفی به کاربردهای دو لایه‌ای و  $4800 \times (3/5) \times (80\%) = 2304$  ماشین مجازی به کاربردهای سه لایه‌ای تخصیص داده می‌شود. شمار هسته‌هایی از میزبان فیزیکی که باید فعال باشد با توجه به سناریو است. برای نمونه، در سناریویی که ۱۶ ماشین مجازی می‌تواند روی ماشین فیزیکی پیکربندی شود، تنها ۱۶ هسته فعال بوده و ۴۸ هسته دیگر غیرفعال هستند.

### الگوریتم ۳: مکان‌یابی آگاه به بار کاری لایه‌های VMTAها

```

1. Input: cVM, set  $\Lambda$ , resources demand vectors of VMs in  $\Lambda$  and cVM
2. Output: Allocation of candidate VM (cVM) to a PM in  $\Lambda$  (cVM  $\rightarrow \Lambda$ )
3. max AngDev  $\leftarrow -\infty$ 
4. max EuclDist  $\leftarrow -\infty$ 
5. desHost  $\leftarrow \{\emptyset\}$ 
6. for ( $j \leftarrow 1$  to  $|\Lambda|$ )
7.   if ( $\overline{\text{demand}}_{cVM} \leq (1 - \eta - \text{residual}_{PM_j})$ )
8.     sumAngDev  $\leftarrow 0$ 
9.     sumEuclDev  $\leftarrow 0$ 
10.    for ( $k \leftarrow 1$  to  $|VM_j|$ )
11.      sumAngDev  $\leftarrow \text{sumAngDev} + \arccos\left(\frac{cVM^T \cdot VM_{j,k}}{|cVM| |VM_{j,k}|}\right)$ 
12.      sumEuclDev  $\leftarrow \text{sumEuclDev} + \left(\sum_{i=1}^{|\text{res}|} (\text{demand}_{\text{res}, cVM} - \text{demand}_{\text{res}, VM_{j,k}})^2\right)^{0.5}$ 
13.    end for
14.    avgAngDev = sumDist /  $|VM_j|$ 
15.    avgEuclDev = sumEuclDev /  $|VM_j|$ 
16.    if (avgAngDev > max AngDev)
17.      desHost  $\leftarrow \{PM_j\}$ 
18.      max AngDev  $\leftarrow$  avgAngDev
19.      max EuclDev  $\leftarrow$  sumEuclDev
20.    else if (avgAngDev  $\approx$  max AngDev)
21.      if (avgEuclDev > max EuclDev)
22.        desHost  $\leftarrow \{PM_j\}$ 
23.        max AngDev  $\leftarrow$  avgAngDev
24.        max EuclDev  $\leftarrow$  avgEuclDevDev
25.      end if
26.    end if
27.  end for
28. end for
29. if (desHost  $\neq \{\emptyset\}$ )
30.   allocate(desHost,  $VM_j$ )
31.   update(residual resources vector of  $PM_j$  in  $\Lambda$ )
32. end if
33. return
    
```

## ۶-۱- پارامترهای کارایی و سناریوهای ارزیابی

برای ارزیابی راهبرد مکان‌یابی پنج پارامتر کارایی را برای ارزیابی الگوریتم‌های پیشنهادی در مقایسه با روش تخصیص Best Fit به

<sup>1</sup> Ranking and Workload-aware Placement (RWP)

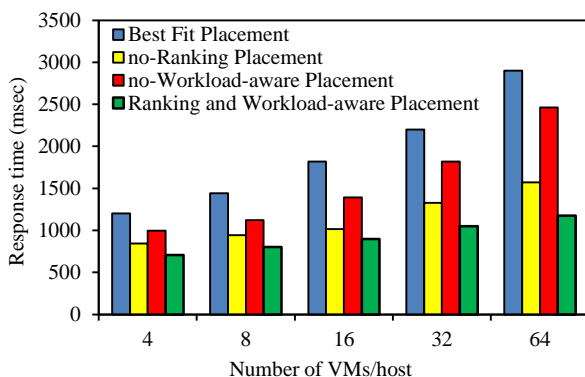
<sup>2</sup> no-Ranking Placement

<sup>3</sup> no-Workload-aware Placement

کاربردهای دو لایه‌ای و سه لایه‌ای می‌بینیم.

هر چند هدف فناوری مجازی‌سازی جداسازی کارایی عملیاتی ماشین‌های مجازی هم‌نشین است، ماشین‌های مجازی برای دستیابی به منابع با هم در رقابت هستند. درخواست‌ها نه تنها برای دستیابی به منابع قابل دیدن مانند پردازشگر، دیسک و حافظه اصلی رقابت دارند، بلکه برای گرفتن منابع نرم افزاری غیرقابل دیدن مانند نخ‌ها، قفل‌های پایگاه داده و پهنای باند حافظه در رقابت می‌باشند. هرگاه بار زیادی (در سناریوی نرخ ۱۲۰ درصد) به سیستم فرستاده می‌شود، رقابت بین درخواست‌ها منجر به زمان انتظار بیشتر برای دستیابی به منابع می‌شود. نخست، درخواست‌ها باید برای دستیابی به منابع نرم افزاری رقابت کنند و سپس در بدست‌گیری منابع سخت افزاری رقابت کنند.

برای نمونه، هنگامی که درخواست‌ها از سوی کاربران فرستاده می‌شود، سیستم عامل لایه نخست (لایه وب) برای پاسخ به درخواست‌های HTTP به آنان پورت‌ها و نخ‌ها را تخصیص می‌دهد. هرگاه درخواستی این منابع را دریافت نمود، برای اجرا روی منابع سخت افزاری می‌بایست منتظر مانده و رقابت نمایند. سپس، این درخواست به لایه دیگر (لایه کاربرد) فرستاده می‌شود تا برای بدست‌گیری منابع این لایه رقابت کنند. برای انجام و اجرای نهایی، درخواست باید برای دستیابی به منابع پایگاه داده همانند قفل پایگاه داده و سپس سخت‌افزاری میزبان پایگاه داده رقابت کند. هرگاه شمار درخواست‌ها زیاد می‌شود، چون درخواست باید لایه‌های رقابتی را بگذرانند، با دیرکرد اجرایی زیادی روبرو می‌شوند. بنابراین، زمان پاسخ زیاد شده و سرانجام شمار تخطی از سطح سرویس توافقی افزایش می‌یابد.



شکل ۱۰: مقایسه زمان پاسخ VMها

پیشنهادی برای مکان‌یابی VMهاست. هنگامی که شمار ماشین‌های مجازی (لایه) روی یک میزبان کم است، ماشین‌ها شمار کمتری مهاجرت پی‌درپی را تجربه می‌کنند. هرچند، با افزایش شمار ماشین‌های مجازی شمار مهاجرت‌های پی‌درپی نیز زیاد می‌شود. در این وضعیت، شمار بیشتری از ماشین‌های مجازی در بدست‌گیری منابع با هم رقابت می‌کنند. این پدیده موجب می‌شود ماشین‌های مجازی برای دستیابی به منابع بیشتر منتظر باشند. بیشتر اینکه، بهره‌وری منابع افزایش یافته و سرانجام احتمال رفتن ماشین فیزیکی به آستانه بهره‌وری بالا افزایش یابد. بنابراین، زمان پاسخ درخواست‌های کاربردها افزایش می‌یابد، شکل (۱۰). در این شرایط، همان‌گونه که در شکل (۱۱) نشان داده شده است، کاربردهای میهمان با تخطی از سرویس بیشتری روبرو می‌شوند.

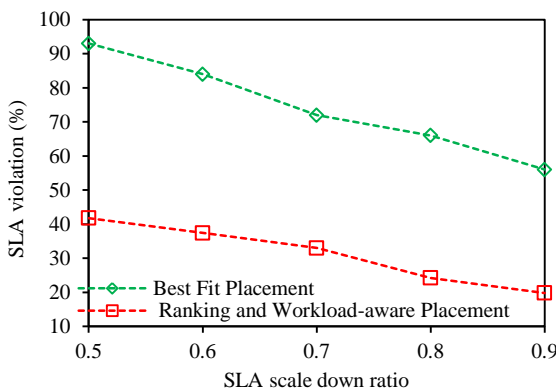
همان‌گونه که نتایج نشان می‌دهد، الگوریتم تخصیص BF دارای زمان پاسخ، تخطی از سرویس و انرژی مصرفی را نسبت به دیگر سناریوها داشته است (شکل (۱۲)). این کاهش کارایی برای این است که این الگوریتم تنها کمترین منابع باقی‌مانده میزبانان را برای جایابی ماشین‌های مجازی در نظر می‌گیرد. از سوی دیگر، RWP ماشین میزبان را با آگاهی از ویژگی‌های VMها و ویژگی کاربردهای اجرایی روی میزبانان هدف گزینش می‌کند. راهبرد پیشنهادی با آگاهی از وابستگی بین لایه‌های VMها، VMها را که با تخطی زیادی در دوره اجرایشان روبرو بوده‌اند را زودتر مکان‌دهی می‌کند. همچنین نتایج نشان می‌دهد هرگاه RWP بدون فاز آخر (مکان‌یابی بدون آگاهی از بارکاری (nWP)) اجرا شود، این روش مکان‌یابی در سنجش با nRP که مکان‌یابی را بدون رتبه‌بندی و با در نظر گرفتن نوع بار کاری انجام می‌دهد با میزان بیشتری از زمان پاسخ، تخطی از سرویس و مصرف انرژی مرکز داده همراه است. مکان‌یابی VMها با اجرای nWP منجر به کنار هم قرارگیری لایه‌هایی می‌شود که روی منابع مشترک رقابت دارند. افزایش رقابت موجب طولانی شدن طول صف انتظار منابع شده و در نهایت منجر به کاهش پارامترهای کیفیت سرویس VMها می‌شود.

ما همچنین اثر نرخ ارسال بارهای VMها را بر کارایی VMها بررسی نمودیم. شکل (۱۳) نشان دهنده نتایج سنجش کاربردهای دو لایه‌ای و سه لایه‌ای برای الگوریتم مکان‌یابی BF و RWP هنگامی که نرخ بارهای کاری از ۴۰ درصد تا ۱۲۰ درصد در نوسان است.

هرگاه نرخ ارسال افزایش می‌یابد، نرخ مهاجرت‌های پی‌درپی افزایش می‌یابد. تفاوت زیادی بین نتایج BF در تخصیص

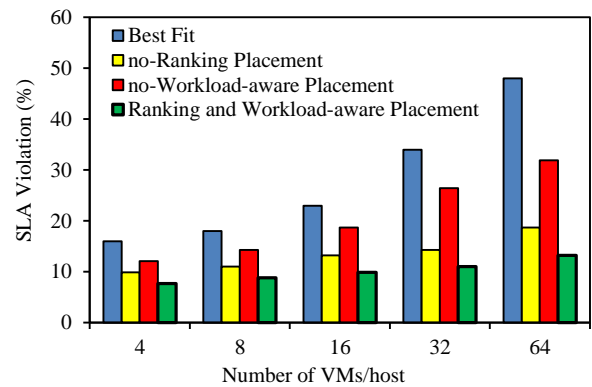
جولوگیری از مهاجرت پی‌درپی، آن را زودتر مکان‌دهی می‌کند. بیشتر اینکه، RWP برای کاهش تداخل کارایی بین یک لایه‌ها گزینش شده روی یک میزبان هدف و دیگر لایه‌های میهمان آن میزبان تلاش می‌کند. بنابراین همان‌طور که نتایج نشان می‌دهد، نرخ تخطی از سرویس توافقی و زمان پاسخ RWP بسیار کمتر از الگوریتم BF است. در واقع، الگوریتم BF الگوریتمی اکتشافی است که مکان‌یابی را بدون در نظر گرفتن کارکرد میزبان، ویژگی‌های کاربرد و ماشین‌های مجازی جا داده شده روی میزبان انجام می‌دهد.

شکل (۱۴) اثر بازه کاهش سطح توافق سرویس را روی میزان تخطی از سطح سرویس نشان می‌دهند. نتایج به روشنی نشان می‌دهند هرگاه اندازه توافق سطح سرویس کم می‌شود، نرخ تخطی از سطح سرویس زیاد می‌شود. هرچند این تاثیرگذاری روی RWP کمتر از الگوریتم BF است. RWP برای یافتن میزبانان مناسب برای لایه‌های مهاجر، تقاضای منابع و وابستگی بین لایه‌های VMTAها را در نظر گرفته و VMTAهایی که دارای میزان تخطی از سرویس بیشتری می‌باشند را زودتر جا می‌دهد. برای این کار، RWP ماشین‌های مجازی وابسته را روی میزبانانی با کمترین تداخل کاری با دیگر ماشین‌های مجازی جا می‌دهد.

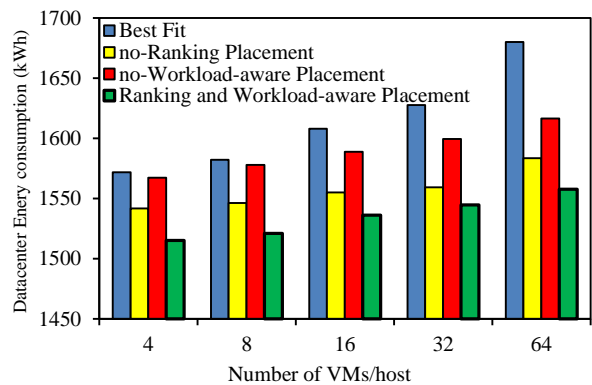


شکل ۱۴: مقایسه نرخ تخطی از سرویس

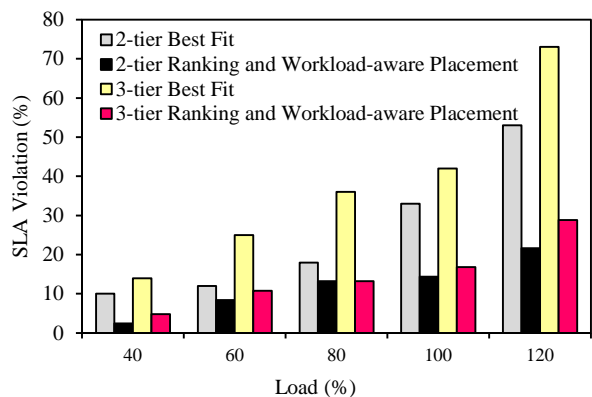
شکل‌های (۱۵) تا (۲۰) نتایج سنجش میانگین بهره‌وری منابع و مصرف انرژی میزبانان بین RWP و BF را نشان می‌دهند. نتایج نشان می‌دهند که انحراف داده‌های وابسته به بهره‌وری منابع و مصرف انرژی در الگوریتم BF خیلی بیشتر از RWP است. در واقع، فاز تخصیص با آگاهی از ویژگی‌های بار کاری موجب پخش مناسب لایه‌ها (بار کاری) بین میزبانان می‌شود تا نرخ بهره‌وری منابع میزبانان تقریباً هم اندازه شود. شکل‌های (۲۱) و (۲۲) نشان می‌دهند، هنگامی که شمار ماشین‌های مجازی روی یک میزبان فیزیکی افزایش می‌یابد، بهره‌وری منابع و مصرف انرژی الگوریتم BF انحراف زیادی از میانه داده‌های وابسته دارد.



شکل ۱۱: مقایسه نرخ تخطی از سرویس



شکل ۱۲: مقایسه انرژی مصرفی مرکز داده

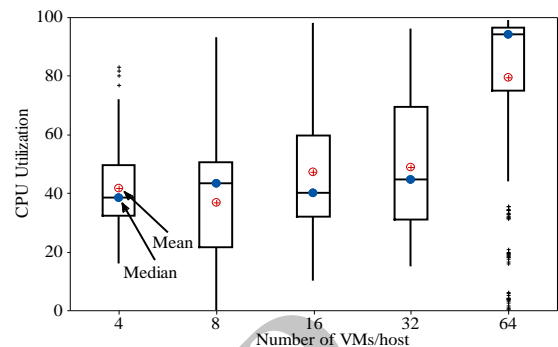


شکل ۱۳: مقایسه نرخ تخطی از سرویس

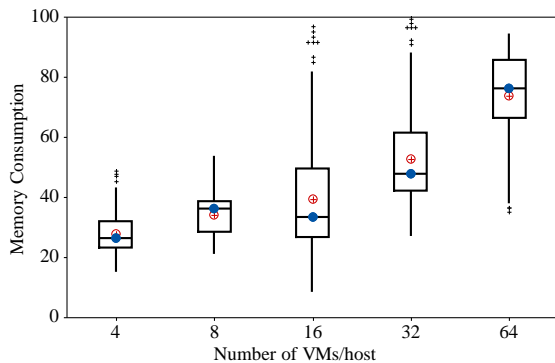
این فرآیند برای VMTAهایی که دارای لایه‌های بیشتری روی میزبانان با بهره‌وری بالا هستند، حادث‌تر است. لایه‌ای که روی میزبان با بهره‌وری بالاست، نه تنها می‌تواند روی کارایی لایه‌های دیگر VMTAها اثر گذارد، بلکه می‌تواند موجب کاهش کارایی لایه‌های وابسته شود. در این موارد، اگر لایه‌ای که تخطی از سرویس داشته روی میزبان نامناسبی جادهی شود، مهاجرت ماشین مجازی پیوسته روی می‌دهد. راهبرد اولویت‌دار پیشنهادی با در نظر گرفتن تخطی از سرویس و تقاضای منابع لایه مهاجر براساس رابطه (۱۵)، لایه‌ای دارای اولویت را برگزیده و برای

بنابراین، این الگوریتم تمایل به پرکردن گنجایش میزبانانی که بیشتر بهره‌ور شده‌اند دارد، در حالی که دیگر میزبانان بخش کمی از منابع‌شان بهره‌ور شده است و ماشین‌های مجازی کمی را میزبانی می‌کنند. این روش منجر به پراکندگی زیادی نسبت به میانه بهره‌وری منابع و انرژی مصرفی دارد و لایه‌ها به صورت ناعادلانه به میزبانان تخصیص داده می‌شوند. از سوی دیگر راهبرد مکان‌یابی پیشنهادی با بهره‌گیری از رتبه‌بندی براساس منابع و کارایی VMTA و با آگاهی از نوع لایه‌ها، آنان را روی میزبانان مناسب با کمترین تداخل کارکردی جا می‌دهد. در نهایت، شمار مهاجرت‌های پی‌درپی کاهش یافته و لایه‌ها زمان زیادی روی میزبانان مناسب اجرا می‌شوند.

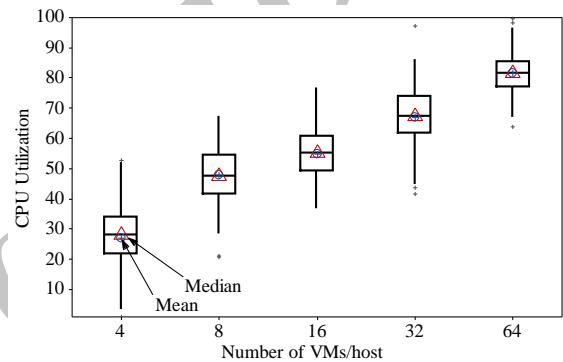
همان‌گونه که پیشتر گفته شد، این الگوریتم تخصیص روش اکتشافی را برای یافتن میزبانی با کمترین منابع مانده بدون در نظر گرفتن کارکرد میزبانان هدف و ویژگی بار کاری ماشین مجازی مهاجر اجرا می‌کند.



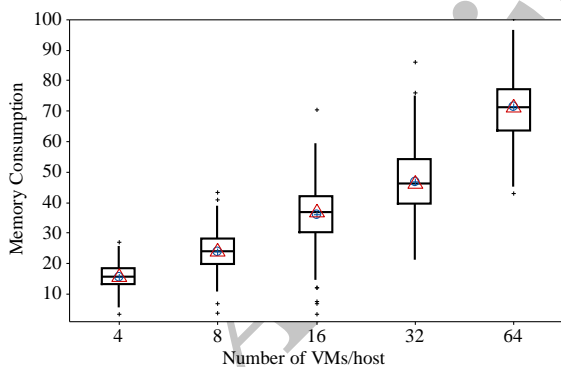
شکل ۱۵: بهره‌وری پردازشگر الگوریتم BF



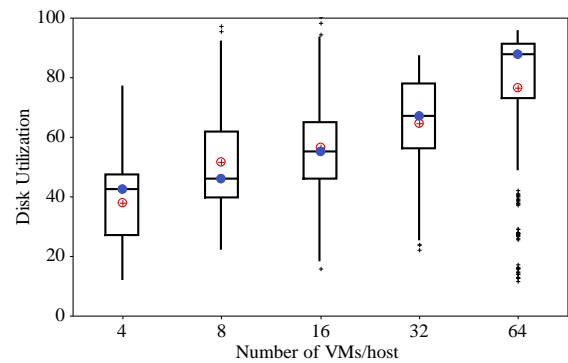
شکل ۱۹: مصرف حافظه اصلی الگوریتم BF



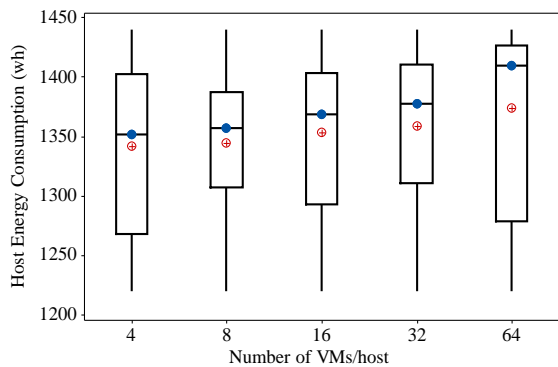
شکل ۱۶: بهره‌وری پردازشگر راهبرد مکان‌یابی RWP



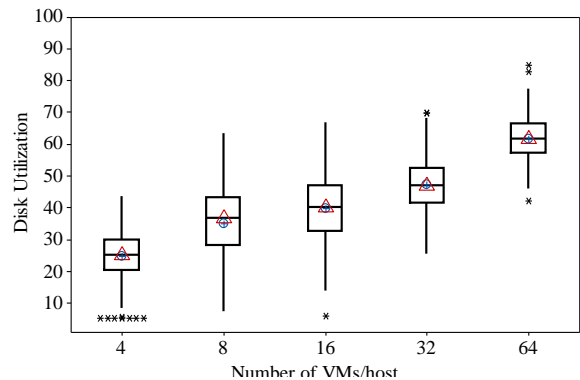
شکل ۲۰: مصرف حافظه اصلی راهبرد مکان‌یابی RWP



شکل ۱۷: بهره‌وری دیسک الگوریتم BF



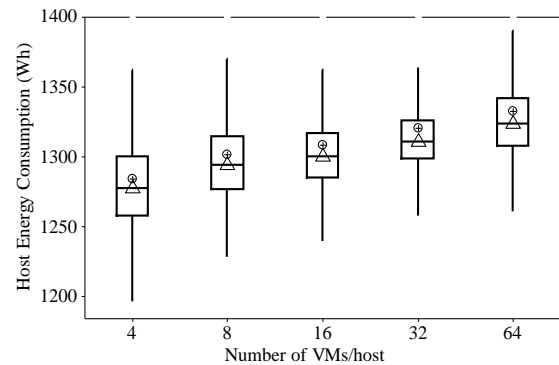
شکل ۲۱: مصرف انرژی میزبان فیزیکی الگوریتم BF



شکل ۱۸: بهره‌وری دیسک راهبرد مکان‌یابی RWP



- [6] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Adaptive resource provisioning for read intensive multi-tier applications in the cloud," *Futur. Gener. Comput. Syst.*, Vol. 27, No. 6, pp. 871–879, 2011.
- [7] C. W. Cobb and P. H. Douglas, "A Theory of Production," *Am. Econ. Rev.*, Vol. 18, No. 1, pp. 139–165, 1928.
- [8] M. A. Salehi, A. N. Toosi, and R. Buyya, "Contention management in federated virtualized distributed systems: Implementation and evaluation," *Softw. Pract. Exp.*, Vol. 44, pp. 353–368, 2014.
- [9] R. Krebs, C. Momm, and S. Kounev, "Metrics and techniques for quantifying performance isolation in cloud environments," *Sci. Comput. Program.*, Vol. 90, pp. 116–134, 2014.
- [10] T. Wood, L. Cherkasova, K. Ozonat, and P. Shenoy, "Profiling and Modeling Resource Usage of Virtualized Applications," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, 2008, New York, USA, pp. 366–387.
- [11] Y. Z. Y. Zhao and W. H. W. Huang, "Adaptive Distributed Load Balancing Algorithm Based on Live Migration of Virtual Machines in Cloud," *2009 Fifth Int. Jt. Conf. INC, IMS IDC 2009*, Beijing, China, pp. 170–175.
- [12] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and Gray-box Strategies for Virtual Machine Migration," in *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation*, 2007, Berkeley, USA, p. 17.
- [13] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos, "Cloud Computing: Survey on Energy Efficiency," *ACM Comput. Surv.*, Vol. 47, No. 2, pp. 33:1–33:36, Dec. 2014.
- [14] K. Lu, R. Yahyapour, P. Wieder, C. Kotsokalis, E. Yaqub, and A. I. Jehangiri, "QoS-aware VM placement in multi-domain service level agreements scenarios," in *IEEE International Conference on Cloud Computing, CLOUD*, 2013, Gottingen, Germany, pp. 661–668.
- [15] Y. Song, H. Wang, L. Yaqiong, B. Feng, and Y. Sun, "Multi-tiered on-demand resource scheduling for VM-based data center," in *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009*, 2009, Shanghai, China, pp. 148–155.
- [16] P. D. Kaur and I. Chana, "A resource elasticity framework for QoS-aware execution of cloud applications," *Futur. Gener. Comput. Syst.*, Vol. 37, pp. 14–25, 2014.
- [17] Q. Li, Q. Hao, L. Xiao, and Z. Li, "An Integrated Approach to Automatic Management of Virtualized Resources in Cloud Environments," *Comput. J.*, Vol. 54, No. 6, pp. 905–919, Jun. 2011.
- [18] N. Grozev and R. Buyya, "Performance Modelling and Simulation of Three-Tier Applications in Cloud and Multi-Cloud Environments," *Comput. J.*, Vol. 58, No. 1, pp. 1–22, 2013.
- [19] K. RahimiZadeh, R. Nasiri Gerde, M. AnaLoui, and P. Kabiri, "Performance evaluation of Web server workloads in Xen-based virtualized computer system: analytical modeling and experimental validation," *Concurr. Comput. Pract. Exp.*, Vol. 27, No. 17, pp. 4741–4762, 2015.
- [20] N. Mi, G. Casale, L. Cherkasova, and E. Smirni, "Sizing multi-tier systems with temporal dependence: Benchmarks and analytic models," *J. Internet Serv. Appl.*, Vol. 1, No. 2, pp. 117–134, 2010.
- [21] N. Megiddo, "Linear Programming in Linear Time When the Dimension Is Fixed," *Journal of the ACM*, Vol. 31, pp. 114–127, 1984.
- [22] RUBiS. [Online]. <http://forge.ow2.org/projects/rubis/>. [Accessed: 06-Jun-2015].
- [23] Httpperf. [Online]. <http://www.hpl.hp.com/research/linux/httpperf/>. [Accessed: 06-Jun-2015].
- [24] D. Borgetto, M. Maurer, G. Da-Costa, J.-M. Pierson, and I. Brandic, "Energy-efficient and SLA-aware management of IaaS clouds," in *2012 Third International Conference on Future Energy Systems Where Energy Computing and Communication Meet eEnergy*, 2012, New York, USA, pp. 1–10.
- [25] CITRIX. [Online]. <http://support.citrix.com/article/CTX137837>. [Accessed: 06-Jun-2015].
- [26] Mediawiki. [Online]. <https://www.mediawiki.org/wiki/MediaWiki>. [Accessed: 06-Jun-2015].



شکل ۲۲: مصرف انرژی میزبان فیزیکی راهبرد مکان‌یابی RWP

## ۷- نتیجه‌گیری

در این مقاله، راهبردی برای مکان‌یابی لایه‌های VMهای در یک مرکز داده پیشنهاد شد. راهبرد پیشنهادی، لایه‌ها را با هدف کاهش تداخل کارکردی با لایه‌های دیگر و کم کردن شمار مهاجرت‌های پی‌درپی جاده می‌کند. دست‌آوردها نشان از کارایی مناسب راهبرد پیشنهادی در توزیع بار (لایه‌ها) مناسب، مصرف انرژی و بهره‌وری عادلانه منابع میزبانان فیزیکی در مرکز داده دارد. همچنین، این راهبرد تداخل کارکرد و نرخ تخطی از سرویس توافقی VMهای را به خوبی کاهش می‌دهد.

در این پژوهش، از دیسک اشتراکی برای ماشین‌های مجازی روی یک ماشین فیزیکی استفاده شده است. راهبرد مکان‌یابی ارائه شده می‌تواند با در نظر گرفتن سکوهای ذخیره‌های اشتراکی همانند SAN و NAS و بکارگیری آن در جایابی ماشین‌های مجازی برای آینده پژوهشی پیشنهاد می‌شود. همچنین، تاثیر زیرساخت شبکه مورد بررسی قرار نگرفته است. در این راستا، یافتن مصالحه‌ای بین جایابی کارا، پهنای باند شبکه اشتراکی، هزینه جایابی و مهاجرت نیز می‌تواند مورد بررسی قرار گیرد.

## مراجع

- [1] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Trans. Parallel Distrib. Syst.*, Vol. 24, No. 7, pp. 1366–1379, 2013.
- [2] L. Cherkasova, K. Ozonat, N. Mi, J. Symons, and E. Smirni, "Automated anomaly detection and performance modeling of enterprise applications," *ACM Trans. Comput. Syst.*, Vol. 27, pp. 1–32, 2009.
- [3] Q. Zhu and T. Tung, "A performance interference model for managing consolidated workloads in QoS-aware clouds," in *Proceedings of IEEE 5th International Conference on Cloud Computing, CLOUD 2012*, 2012, San Jose, USA, pp. 170–179.
- [4] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, C. Pu, and Y. Cao, "Who is your neighbor: Net I/O performance interference in virtualized clouds," *IEEE Trans. Serv. Comput.*, Vol. 6, pp. 314–329, 2013.
- [5] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An analytical model for multi-tier internet services and its applications," *ACM SIGMETRICS Perform. Eval. Rev.*, Vol. 33, pp. 291–302, 2005.

## Multi-Tier Applications Placement in Virtualized Datacenter

Keyvan RahimiZadeh<sup>1\*</sup>, Morteza AnaLoui<sup>2</sup> and Peyman Kabiri<sup>3</sup>

1\*- Corresponding Author: Faculty of Engineering, Yasouj University, Daneshjoo Street, Yasouj, Iran.

2,3- School of Computer Engineering, Iran University of Science and Technology, Hengam Street, Tehran, Iran.

<sup>1\*</sup> RahimiZadeh@yu.ac.ir, <sup>2</sup> AnaLoui@iust.ac.ir, <sup>3</sup> Peyman.Kabiri@iust.ac.ir

**Abstract-** The increasing use of Multi-Tier Application (MTA) in virtualized environments necessitates performance evaluation of such systems to achieve scalable and flexible services. However, providing appropriate performance for Virtualized Multi-Tier Applications (VMTA) that have complex architecture is much more difficult than traditional application architecture. In this paper, we propose a placement strategy to settle virtual machines (VMs) of VMTAs in virtualized datacenters. First, VMs hosting tiers are ranked based on Cobb-Douglas production function. Then, VMTAs are prioritized based on resources utilization and performance metrics of hosts. Finally, VMTAs are placed regarding resource demands of tiers and their functional dependency. Results reveal that the proposed solution excels in terms of load balancing and energy consumption, while reduces Service Level Agreement (SLA) violation and VMs interference in the datacenter.

**Keywords-** Multi-tier application, virtualization, Cloud computing, performance evaluation, workload interference.

Archive of SID