

## بهینه سازی هزینه پولی زمانبندی جریان کاری علمی با محدودیت مهلت معین با استفاده از یک الگوریتم ابتکاری در ابرهای عمومی

صدیقه السادات سیدخرازی<sup>۱</sup>، دکتر محمدجعفر تارخ<sup>۲\*</sup>

۱- دانشجوی رشته فناوری اطلاعات گرایش تجارت الکترونیک دانشگاه آزاد قزوین

Zohre.kharrazi@gmail.com

۲- دانشیار، دانشگاه خواجه نصیرالدین طوسی، ایران

mjtarokh@kntu.ac.ir

### چکیده

با پیشرفت شبکه‌های سرعت بالا در سطح کشورها و خطوط پر سرعت اینترنت، امکان استفاده کاربران از منابعی که از لحاظ جغرافیایی در مکان‌های دیگری قرار گرفته اند فراهم گردیده است. محاسبات ابری یک مدل محاسباتی در هرمنطقه جغرافیایی در زیرساخت اینترنت فراهم می‌سازد. یکی از مسائل مهم در ابر زمانبندی و اجرای جریان‌های کاری بار عایت برخی از معیارهای کیفیت سرویس می‌باشد. محاسبات ابری با انواع متفاوتی از منابع مجازی کار می‌کند در نتیجه الگوریتم‌های زمانبندی جریان کاری نقش مهمی را در ابر دارند. در محیط ابر برای اجرای سرویس‌های مختلف از زمانبند استفاده می‌شود. در این مقاله الگوریتمی ارائه شده است که یک جریان کاری را دریافت می‌کند، سپس بر روی ابر عمومی زمانبندی می‌کند. هدف این الگوریتم زمانبندی جریان کاری بر روی منابع ناهمگون ابر و اجرای کلیه وظایف جریان کاری در زیرمهلت تعیین شده با کمترین هزینه ممکن می‌باشد. این الگوریتم وظایف را با رتبه بندی روبه بالا رتبه بندی می‌کند، سپس وظایف را برحسب رتبه پایین انتخاب و براساس والد بحرانی خوشه بندی می‌کند. در انتها وظایف خوشه‌ها به ترتیب شماره خوشه، به بهترین منابع ممکن تخصیص می‌یابند. روش پیشنهادی با الگوریتم ترکیب خوشه CCA (cluster combining algorithm) مورد ارزیابی قرار می‌گیرد. آزمایش‌ها بر روی جریان‌های کای Montage و Epigenomics با دو ضریب مهلت زمانی  $\alpha=1.35$  و  $\alpha=1.6$  انجام می‌شود. نتایج شبیه سازی نشان می‌دهد، هزینه اجرای همه جریان‌های کاری در روش پیشنهادی نسبت به روش CCA ۱۵ درصد کاهش یافته است.

کلمات کلیدی: زمانبندی وظایف، رایانش ابری، قیمت گذاری، مهلت زمانی، زیرساخت به عنوان سرویس

### ۱-مقدمه

توسعه پویای محاسبات ابری و توزیع شده منجر به راه حل‌های باثباتی برای پردازش داده ارائه شده است. امروزه تمرکز زیادی بر مدل‌های جدید برای انواع خاصی از پردازش‌ها به وجود آمده است. علاوه بر سیستم‌های ذخیره سازی توزیع شده، راه حل‌های بسیاری از وظایف خاص برای مثال دسترسی تصادفی سریع به داده، پردازش خط لوله، محاسبات گراف و غیره به دست آمده است [1,2]. برنامه‌های کاربردی جریان کاری با برنامه‌های موازی، همراه هستند که شامل مجموعه ای از وظایف محاسباتی مرتبط با داده‌ها و وابستگی‌های کاری است [3]. به عنوان مثال از این برنامه‌ها می‌توانیم به جریان‌های پیچیده علمی [4] و جریان‌های کاری چند سرویسه وب [5] اشاره کنیم یا برنامه‌هایی که داده‌های بزرگ را پردازش می‌کنند. چنین برنامه‌هایی به طور معمول نیاز به توان محاسباتی بیش از توانایی یک دستگاه دارند [6]. معمولاً یک جریان

کاری توسط یک DAG<sup>1</sup> توصیف می شود که یک گراف غیر چرخه ای است که در آن هر یک از کارهای محاسباتی توسط یک گره نشان داده شده است و هر داده یا وابستگی بین وظایف توسط یک لبه بین گره های مربوطه نشان داده شده است. برای اجرای جریانهای کاری در زمان معقول، ما نیاز به محیط محاسباتی با کارایی بالا مانند شبکه، خوشه یا ابر دارد. اخیراً برخی محققان ایده استفاده از ابرهای IaaS<sup>2</sup> [7] را برای اجرای جریانهای علمی [9,8,3] در نظر گرفته اند.

رایانش ابری اخیراً به عنوان یکی از جایگاههای مناسب برای اجرای این برنامه های علمی پیچیده ارائه شده است. بسیاری از محققین به بررسی استفاده از رایانش ابری برای برنامه های علمی پرداخته اند و به این نتیجه رسیده اند که این سیستم های ابری می توانند از نظر عملکرد و هزینه، شرایط مناسبی را ایجاد کنند. اما با ظهور این الگوی محاسباتی جدید، الگوریتم های زمانبندی جدیدی که بتوانند از فواید سیستم های ابری استفاده کنند و چالش های خاص محیط های ابری را هم رفع کنند، در این قسمت مورد نیاز می باشد. ابرها می توانند مدل های محاسباتی را ارائه کنند که این امکان را فراهم می کند تا منابع پردازشی بر روی اینترنت تحویل داده شود و سپس از یک مدل پرداخت در زمان مصرف استفاده می کند که در این مدل، کاربران هزینه را بر اساس میزان مصرف از منابع پرداخت می کنند [10]. سیستم های ابری می توانند تعداد نامحدودی از منابع پردازشی را ارائه کنند و این کار با استفاده از مفاهیم مجازی سازی محقق می شود که می توان از آن ها در بازه زمانی معقول استفاده کرد و هزینه استفاده از آن نیز بر اساس میزان مصرف از منابع مشخص می شود. پلتفرم های ابری از این رو می توانند یک روش جایگزین برای اجرای برنامه های علمی باشند که در این برنامه ها، منابع در سازمان های تحقیقاتی قرار ندارد بلکه از یک مرکز داده ی بزرگ در زمان نیاز اجاره می شود. این برون سپاری کردن باعث کاهش بار مالی سازمان ها می شود [1].

در این مقاله، یک الگوریتم زمانبندی جریان کاری پیشنهاد شده است که جریان کاری را بر روی زیر ساخت های ابر با منابع ناهمگون زمانبندی کند. بطوریکه جریان کاری را در مهلت تعیین شده کاربر، با کمترین هزینه ممکن اجرامی کند. برای بهبود هزینه از یک روش خوشه بندی با رتبه روبه بالا مبتنی بر HEFT [11] استفاده می شود. سپس با استفاده از مفهوم والد بحرانی<sup>3</sup> عمل خوشه بندی انجام می شود. سپس خوشه ها به ترتیب انتخاب می شوند و منابع ارزانی که بتوانند وظایف هر خوشه را در زیرمهلت تعیین شده اجرا کنند تخصیص صورت می گیرد.

ادامه مقاله به شرح زیر تدوین شده است. بخش ۲ بررسی کار مرتبط، بخش 3 مدل سیستم ما شامل مدل نرم افزار، مدل ابر، مدل هزینه و تعریف مساله است. الگوریتم زمانبندی پیشنهادی در بخش ۴ توضیح داده شده است. ارزیابی عملکرد در بخش ۵ ارائه شده است بخش ۶ نتیجه گیری مقاله است.

## ۲- کار مرتبط

برخی از مهمترین و جدیدترین الگوریتم هایی که برای زمانبندی جریان کاری ارائه شده اند که زمان اجرا و هزینه را به عنوان پارامتر کیفیت سرویس در نظر گرفته اند می توان در ادامه بیان کرد.

Adhikari و همکاران [12] یک الگوریتم زمانبندی با کارایی بالا برای محیط محاسبات ابری پیشنهاد دادند. این الگوریتم ماشین های مجازی مناسب را انتخاب می کند تا در مهلت مشخص کارها را اجرا کند و بهره وره منابع افزایش یابد. این الگوریتم در مهلت معین هزینه را کاهش می دهد. توسط Reddy و همکاران [13] الگوریتم زمانبندی جریان کاری با زمان و هزینه آگاه پیشنهاد شده است که وظایف یک جریان کاری را برای تکمیل اجرا در کوتاه ترین زمان ممکن زمانبندی می کند تا قیمت خدمات ارائه شده به مشتریان به حداقل برسد. برای کاهش زمان اجرا ابتدا وظایف بحرانی زمانبندی می شوند سپس برای کاهش هزینه از منابع ارزانی که وظایف را در بازه زمانی تعیین شده اجرا می کنند استفاده می شود. Deldari و همکاران [14] یک الگوریتم زمانبندی ایستای جریان کاری در ابر برای منابع پردازشی چند هسته ای ارائه داده اند که می تواند هزینه

<sup>1</sup> Directed Acyclic Graph

<sup>2</sup> Infrastructure as a Service

<sup>3</sup> Critical Parent

## سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

اجاره منابع را در مهلت معین کمینه کند. الگوریتم پیشنهادی از یک روش خوشه بندی برای تقسیم جریان کاری به تعدادی خوشه استفاده می کند و تلاش می کند تا خوشه ها را به گونه ای ترکیب کند تا اهداف اصلی الگوریتم را بدست آورند. در هر گام از روش رتبه بندی با بهترین ترکیب موجود انتخاب می کند. روش خوشه بندی مبتنی بر الگوریتم PCH [15] می باشد. Viriyapant و همکاران [16] یک الگوریتم زمانبندی جریان کاری با هدف کاهش هزینه در مهلت معین با سه الگوریتم ابتکاری ارائه داده اند. همچنین تعداد نامحدودی ماشین مجازی در ابر عمومی وجود دارد. الگوریتم زمانبندی کارها را طوری روی ماشین های مجازی نگاهش می کند که جریان های کاری در مهلت معین اجرا می شوند و هزینه اجرای جریان های کاری کمینه می شود. Arabnejad و همکاران [17] یک الگوریتم زمانبندی ابتکاری برای اجرای جریان کاری در محدودیت مهلت معین براساس مسیر بحرانی جهت کاهش هزینه اجرا ارائه داده اند. در این الگوریتم ابتدا یک مسیر بحرانی انتخاب می کند و کارهای روی مسیر بحرانی را اولویت بندی می کند و در خوشه قرار می دهد. در مرحله دوم هر خوشه زمانبندی می شود. با اجرا شدن هر خوشه در یک نمونه منبع زمان و هزینه انتقال داده کاهش می یابد. عمل رتبه بندی و اولویت بندی هر وظیفه مبتنی بر رتبه بندی رو به بالا [11] استفاده می کند. در واقعیت رتبه روبه بالای یک گره، برابر با طول بلندترین مسیر از آن وظیفه، به وظیفه پایانی گراف است. Topcuoglu و همکاران [18] یک الگوریتم زمانبندی مبتنی بر خوشه بندی کارهای مرتبط به هم برای کاهش هزینه و بهبود کارایی منابع ارائه داده اند. با توجه به تاخیر در انجام وظایف مرتبط با برنامه ریزی محاسبات ابری، یک مدل کارهای سلسله مراتبی مبتنی بر ساختار مورد بحث قرار گرفت و الگوریتم های مربوط به برنامه ریزی کار بر اساس محدودیت تاخیر پیشنهاد شده است. عمل خوشه بندی بصورت سلسله مراتبی در هر گراف انجام می شود. به اینصورت که هر سطح از گراف در یک خوشه قرار می گیرد. این کارها بصورت موازی و همروند بایکدیگر اجرا می شوند. Kanagaraj و همکاران [19] یک رویکرد واقع بینانه برای نمایش و زمانبندی جریان کاری در محیط محاسبات ابری ارائه داده اند. در این تحقیق دو الگوریتم DCWS<sup>1</sup> و CCWS<sup>2</sup> ارائه شده است. الگوریتم اول سعی می کند جریان کاری را در محدودیت مهلت زمانی مشخصی اجرا کند و مدت زمان اجرای جریان کاری را تا حد ممکن کاهش دهد. در الگوریتم دوم هدف کاهش هزینه اجرای جریان کاری می باشد. به این منظور برای اجرای کارها از روش خوشه بندی مبتنی بر مسیر بحرانی استفاده می شود. Sun و همکاران [20] یک الگوریتم زمانبندی برای اجرای جریان کاری جهت افزایش درصد زمانبندی موفق در محدودیت مهلت و محدودیت بودجه ارائه داده اند. معیار میزان زمانبندی موفق برای نشان دادن اثربخشی الگوریتم پیشنهاد شده مورد استفاده قرار گرفته است. برای هر یک از وظایف گراف یک زیرمهلت و یک بودجه معین توزیع می کند. سپس منبعی انتخاب می شود که بر رضایت از محدودیت های هزینه و زمان تمرکز کند. این الگوریتم پیچیدگی زمانی را نیز کاهش می دهد. Dedari و همکاران [21] الگوریتم زمانبندی ترکیبی خوشه ای در محیط ابر پیشنهاد داده اند. الگوریتم زمانبندی پیشنهادی عمل نگاهش وظایف به منابع چند هسته ای در ابر را در محدودیت زمانی معین انجام می دهد به شرط اینکه هزینه اجرای کارها بهبود یابد. روش کار به اینصورت است که هر جریان کاری را به چند خوشه تقسیم می کند. سپس با استفاده از روش رتبه بندی روبه بالا بهترین خوشه ها بایکدیگر ترکیب می شوند و عمل تخصیص صورت می گیرد. سپس عمل زمانبندی از وظیفه ای که بالاترین رتبه را دارد آغاز می شود.

### ۳- مدل زمانبندی سیستم

در این بخش، مفاهیم اصلی از مدل برنامه، مدل ابر زیرساخت، مدل هزینه و همچنین مساله اصلی تعریف می شود.

### ۳-۱- مدل جریان کاری

<sup>1</sup> Deadline Constrained Workflow Scheduling

<sup>2</sup> Cost Constrained Workflow Scheduling

## سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

مدل جریان کاری یکی از موفقترین الگوها برای برنامه‌های علمی روی زیرساخت‌های توزیع شده ابر می‌باشد. از آنجا که وظایف جریان کاری به صورت اختیاری بهم متصل می‌باشند، مدل برنامه مورد استفاده برای یک جریان کاری علمی، گراف غیرممدور جهت دار  $G=(V,E)$  می‌باشد. که در آن  $V=\{t_1,t_2,\dots,t_n\}$  به سری وظایف جریان کاری اختصاص دارد.  $E_{ij}=(t_i,t_j)$  لبه‌های میان راس‌ها را نشان می‌دهد. همچنین هر لبه وزنی دارد که به محدودیت اولویت و مقدار ارتباطات داده‌ای بین وظایف  $t_i$  و  $t_j$  اختصاص دارد. از آنجا که گراف مستلزم ورودی و گره خروجی می‌باشد،  $t_{entry}$  و  $t_{exit}$  با زمان پردازش صفر و ارتباطات صفر به ابتدا و انتهای گراف اضافه شده است. با ارسال هر جریان کاری، یک زیرمهلت زمانی توسط کاربر ارسال می‌شوند. از این رو، زیرمهلت برای هر وظیفه محاسبه می‌گردد که باید طوری باشد که جریان کاری زیرمهلت را برآورده سازد. تمرکز پژوهش فعلی، زمان بندی یک جریان کاری واحد روی ابر و اجرای آن در زیرمهلت تعریف شده توسط کاربر می‌باشد بطوریکه هزینه اجرای جریان کاری کاهش یابد.

### ۳-۲- مدل ابر

مدل ابر شامل یک ارائه دهنده زیرساخت سخت افزاری به عنوان سرویس است که منابع مجازی را به مشتریان خود ارائه می‌دهد. به طور معمول، ارائه دهندگان ابر زیرساخت وظیفه ارائه انواع ماشین‌های مجازی مختلف  $VM=\{vm_1,vm_2,\dots,vm_n\}$  را دارند. هر نوع VM با دو ویژگی توان و هزینه نشان می‌دهند، به طوری که  $p$  نشان دهنده توان پردازش کلی ماشین مجازی است، از جمله سرعت پردازشگر، اندازه حافظه، سرعت I/O و پهنای باند شبکه و  $C$  نشان دهنده هزینه کلی پولی از نوع ماشین مجازی است [6].

### ۳-۳- مدل هزینه

دو دسته هزینه‌های مختلف در نظر گرفته شده است: هزینه منابع و هزینه انتقال داده. هر نوع ماشین مجازی دارای هزینه تعریف شده برای هر فاصله زمانی است [21]. از فواصل زمانی که منابع مورد استفاده قرار گرفته اند (آخرین فاصله زمانی باید کاملاً شارژ شود، حتی اگر آن را به طور کامل استفاده نکنید). هزینه انتقال یا هزینه پهنای باند شامل انتقال داده‌های ورودی و خروجی وظایف از یک منبع به یک منبع دیگر می‌باشد [22].

### ۳-۴- تعریف مساله

در این مقاله، مسئله زمان بندی جریان کاری برای انتخاب یک ماشین مجازی مناسب برای هر وظیفه جریان کاری است بطوریکه: زیرمهلت زمانی رعایت شود و هزینه اقتصادی اجرای کل جریان کاری تا حد زیادی کاهش یابد.

### ۴- الگوریتم‌های پیشنهادی

در این بخش، ابتدا مفاهیم مورد استفاده در الگوریتم پیشنهادی تعریف شده، سپس ایده اصلی الگوریتم پیشنهاد شده و در نهایت الگوریتم پیشنهادی در بخش‌های زیر توضیح داده شده است.

### ۴-۱- تعریف‌های اولیه

در این بخش ابتدا به تعریف مفاهیم و متغیرهای تحقیق می‌پردازیم و در ادامه روش پیشنهادی را تشریح می‌کنیم.

## سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

در رابطه ذیل مدت زمان اجرای وظیفه  $i$  بر روی ماشین مجازی  $j$  محاسبه می شود. در این رابطه  $MI$  تعداد دستورالعمل های وظیفه  $i$  می باشد. هر چقدر این مقدار بیشتر شود مدت زمان اجرا افزایش می یابد.

$$Et_i = \frac{MI_i}{s_j} \quad (1)$$

زمان انتقال داده ها در ابر با رعایت وابستگی بین کارها از رابطه ذیل استفاده می شود:

$$TT(e_{ij}) = \frac{Data(e_{ij})}{Bandwith(VM_i, VM_j)} \quad (2)$$

در این رابطه منظور از  $Data(e_{ij})$  میزان انتقال داده از وظیفه  $t_i$  به وظیفه  $t_j$  می باشد. همچنین  $Bandwith(VM_i, VM_j)$  بیانگر پهنای باند داخلی بین ماشین های مجازی  $VM_i$  به  $VM_j$  در ابر می باشد.

در رابطه زیر زودترین زمان آغاز محاسبه می شود و زودترین زمانی است که  $t_i$  می تواند پردازش خود را آغاز کند:

$$EST(t_{entry}) = 0 \quad (3)$$

$$EST(t_i) = \max_{tp \in t_i \text{ parents}} \{EST(tp) + ET(tp) + TT(ep, i)\}$$

که در آن  $ET(t_i)$  برابر زمان اجرای وظیفه  $t_i$  بر روی منبعی مانند  $s_j$  که کمترین مقدار زمان اجرا را در بین کلیه منابع موجود دارد و مقدار  $TT$  زمان انتقال داده ای باشد و  $EST$  گره آغازین صفر می باشد.

به همین ترتیب زودترین زمان اتمام یک وظیفه زمان بندی نشده زودترین زمانی است که  $t_i$  می تواند محاسبات خود را به پایان برساند و به صورت زیر محاسبه می گردد:

$$EFT(t_i, p_j) = EST(t_i, p_j) + ET(t_i, p_j) \quad (4)$$

$LFT$  یک وظیفه دیرترین زمانی است که، وظیفه می تواند زمان بندی شود با این شرط که همه والدهای آن نیز در دیرترین زمان ممکن قبل از  $D$  زیرمهلت زمانی اجرایی شوند و به صورت ذیل محاسبه می شود:

$$LFT(t_{exit}) = D \quad (5)$$

$$LFT(t_i) = \min_{tc \in t_i \text{ children}} \{LFT(tc) - ET(tc) - TT(ei, c)\}$$

دیرترین زمان پایان گره خروجی برابر زیر مهلت کلی می باشد. در رابطه فوق  $t_c$  نشان دهنده وظیفه فرزند می باشد.

هزینه اجرای وظیفه  $t_i$  بر روی پردازنده  $vm_j$  با  $EC(t_i, vm_j)$  نشان داده می شود و از رابطه ذیل بدست می آید.

$$EC(t_i, vm_j) = ET(t_i, vm_j) \times Price(vm_j) \quad (6)$$

که در آن  $price$  برابر است با هزینه هر واحد زمان استفاده از پردازنده می باشد.

در رابطه ذیل هزینه ارسال داده از یک وظیفه به وظیفه دیگر محاسبه می شود:

$$\begin{cases} price_{B_{intra}(s_{pi}, s_{pj})} = 0 & \text{if } i = j \\ TC(t_{i,j}) = Data(e_{ij}) \times price_{B_{intra}(s_{pi}, s_{pj})} \end{cases} \quad (7)$$

در رابطه فوق  $Tc$  هزینه ارسال داده های مورد نیاز  $t_i$  می باشد.

در رابطه ذیل زمان پایان واقعی وظیفه  $i$  محاسبه می شود:

$$FT(t_i, p) = \max_{tp \in pred(t_i)} \{FT(tp) + et(t_i) + TT(tp, t_i)\} \quad (8)$$

رتبه بندی روبه بالا از رابطه زیر بدست می آید:

$$P_i = ET_i \quad \text{if } succ_i = 0 \quad (9)$$

$$P_i = \max_{j \in succ_i} (P_j + ET_i + TT_{j,i}) \quad \text{else}$$

برای محاسبه والد بحرانی همه وظایف یک گراف از رابطه ذیل استفاده می شود:

$$CP_i = \max_{tp \in t_i \text{ parent}} \{ET(tp) + TT(tp, t_i)\} \quad (10)$$

## ۴- الگوریتم پیشنهادی

## سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

در این بخش الگوریتم پیشنهادی تشریح می شود. ابتدا مفهوم خوشه بندی بیان می شود. یک گراف خوشه بندی شده، گرافی است که وظایف آن در قالب خوشه هایی تقسیم بندی شده و زمان تبادل اطلاعات بین وظایف موجود در یک خوشه برابر صفر قرار داده شده است. در این الگوریتم برای اولویت دهی به وظایف از شاخصی به نام رتبه رو به بالا استفاده می شود که برابر با مسیر بحرانی است.

تعریف مسیر بحرانی: طول بلندترین مسیر از یک وظیفه، به وظیفه پایانی گراف است. ابتدا همه گره های جریان کاری از گره خروجی تا گره ورودی (پایین به بالا) رتبه بندی می شوند. در مرحله ی خوشه بندی وظایف، ابتدا وظایف، بر اساس اولویت و به صورت صعودی در لیست اولویت قرار می گیرند. سپس، به منظور رسیدن به اهداف ذکر شده در الگوریتم زمان بندی پیشنهادی، از والد بحرانی<sup>۱</sup> استفاده می کنیم. بدین صورت که، ابتدا گراف به طور کامل پیمایش شده و والد بحرانی هر وظیفه، محاسبه می شود. به عبارت دیگر، والد بحرانی یک وظیفه، والدی است که بیشترین زمان اجرا و تبادل داده را با آن دارد. خوشه بندی وظایف بدین صورت است که در هر مرحله، یک وظیفه بررسی نشده که کمترین اولویت را دارد به عنوان وظیفه جاری انتخاب می گردد. از آنجاییکه، والد بحرانی هر وظیفه قبلاً مشخص شده است، ابتدا بررسی می شود که آیا آن وظیفه والد بحرانی یکی از وظایف خوشه بندی شده است یا خیر، در صورتی که آن وظیفه، والد بحرانی یکی از وظایف خوشه بندی شده، باشد، به خوشه مربوطه اضافه می شود. در غیر این صورت، وظیفه به خوشه جدید تخصیص داده خواهد شد. این مرحله، آنقدر تکرار می شود تا تمامی وظایف خوشه بندی شوند. سپس هر خوشه به منبع ارزانی که در مهلت معین کارها را اجرا کند تخصیص می یابد.

### ۴-۱- شبه کد الگوریتم زمان بندی پیشنهادی

در شکل ۱ شبه کد الگوریتم پیشنهادی نشان داده شده است. در سطر ۱ جریان کاری دریافت می شود. در سطر ۲ وظایف ورودی و خروجی گراف ایجاد می شوند. سپس در سطر ۳ منابع به همراه مشخصات آنها تعریف می شود. در سطر ۴ تا ۱۱ مقادیری اولیه متغیرها انجام می شود. در سطر ۱۲ عمل رتبه بندی در گراف انجام می شود سپس در سطر ۱۳ تا ۱۸ کارها بر اساس رتبه مرتب می شوند و والد بحرانی هر گره مشخص و خوشه بندی می شود و کارها به بهترین منبعی که بتواند وظایف را در مهلت تعیین شده با کمترین هزینه اجرا کند تخصیص می یابند. و در انتهای زمان بندی هزینه و زمان اجرا کلی محاسبه می شوند.

Procedure schedule() 1. Read workflow s G(T,E) 2. add $t_{entry}, t_{exit}$ into DAG 3. Create resource 4. compute $ET(t_i, s_i)$ 5. calculate user Deadline 6. compute $EST(t_i)$ 7. compute $LFT(t_i)$ 8. compute $EFT(t_i)$ 9. compute $TT(t_i, t_j)$ 10. set $EST(t_{entry}) \leftarrow 0$ 11. set $LFT(t_{exit}) \leftarrow Deadline$ 12. for each task calculate $rank_u$ 13. sort tasks with ascending $rank_u$ 14. while(schedule all tasks) 15. begin 16. select task from start list then calculate critical parent for clustering 17. Select resource for Schedule cluster with Deadline constraint 18. End while 19. calculate cost 20. calculate makespan endp
--

شکل (۱): الگوریتم زمان بندی پیشنهادی

<sup>1</sup> Critical parent

## ۵- شبیه سازی و ارزیابی کارایی

برای ارزیابی عملکرد الگوریتم‌های پیشنهادی در پاسخ به مساله زمانبندی جریان کاری در ابرها، الگوریتم زمانبندی CCA<sup>1</sup> [21] به عنوان الگوریتم پایه انتخاب شده است. برای ارزیابی عملکرد الگوریتم‌ها از گراف‌های واقعی استفاده شده است [23]. که دو مورد از آنها گراف Montage با ۲۵، ۵۰ و ۱۰۰ وظیفه گراف Epigenomics با ۲۴، ۴۷ و ۱۰۰ وظیفه استفاده شده است. پهنای باند میانگین بین منابع محاسباتی تا حد دقیقی، برابر ۲۰ مگابیت بر ثانیه می‌باشد. توان پردازشی منابع بین ۵۰۰ تا ۷۰۰ MIPS یا میلیون دستورات عمل در ثانیه در نظر گرفته شده است. هزینه هر ساعت استفاده از منابع بر مبنای یک ساعت و مطابق ابر آمازون در نظر گرفته شده است.

### ۵-۱- نتایج آزمایش‌ها

در این بخش نتایج شبیه سازی الگوریتم‌های مورد نظر بررسی می‌شود. در جدول ۱ مقایسه زمان اجرا و هزینه اجرای الگوریتم CCA و پیشنهادی با مقدار آلفای ۱٫۶ محاسبه شده است. در این آزمایش با افزایش ضریب، زیرمهلته زمانی برای اجرای گراف‌ها افزایش می‌یابند. با افزایش زیرمهلته شش گراف جریان کاری توسط الگوریتم پیشنهادی در زیرمهلته اجرای می‌شوند. همچنین زمانیکه زیرمهلته افزایش می‌یابد از طرف دیگر هزینه اجرا کاهش می‌یابد. چون امکان زمانبندی بر روی منابع ارزان تر فراهم می‌شود. اما روش CCA جریان کاری montage با وظایف ۲۵ و ۵۰ در مهلت معین اجرا نمی‌کند و تنها چهار جریان کاری در زیرمهلته تعیین شده اجرا می‌شوند. نتایج نشان می‌دهند که هزینه اجرای روش پیشنهادی کمتر از روش CCA می‌باشد.

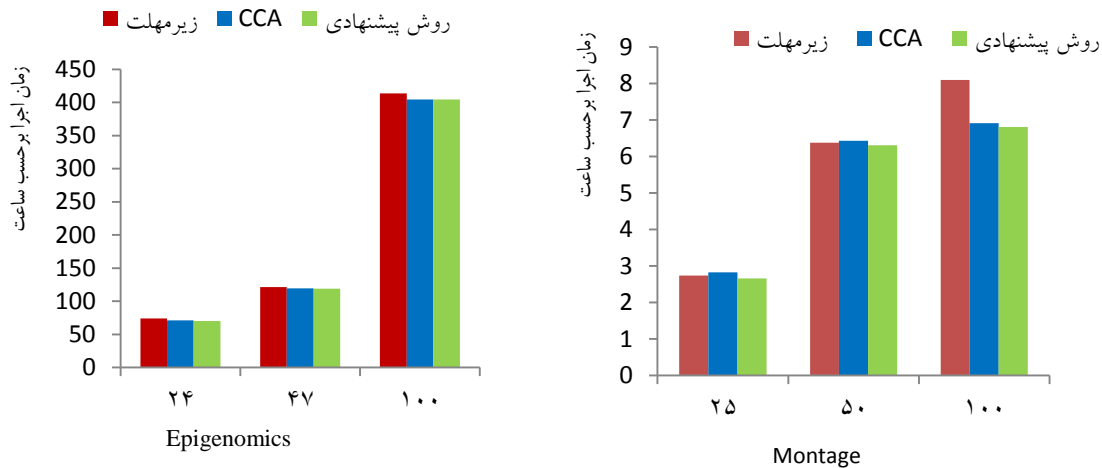
جدول (۱): مدت زمان اجرا و هزینه الگوریتم پیشنهادی و الگوریتم CCA با ضریب مهلت ۱٫۶

جریان کاری	تعداد کارها	زیرمهلته	زمان اجرای الگوریتم پیشنهادی	زمان اجرای الگوریتم CCA	هزینه اجرا با الگوریتم پیشنهادی	هزینه اجرا با الگوریتم CCA
Epigenomics	۲۴	۷۴٫۰۸ h	۷۰٫۰۶ h	۷۱٫۳۱ h	\$ ۳۷٫۹۴	\$ ۴۹٫۲۳
	۴۷	۱۲۱٫۶۵ h	۱۱۹٫۱۳ h	۱۱۹٫۳۸ h	\$ ۶۲٫۹۵	\$ ۷۲٫۱۴
	۱۰۰	۴۱۳٫۷۵ h	۴۰۴٫۴۵ h	۴۰۴٫۵۶ h	\$ ۴۸۲٫۹۸	\$ ۵۰۴٫۹۷
Montage	۲۵	۲٫۷۴ h	۲٫۶۶ h	۲٫۸۳ h	\$ ۱۶٫۵۱	\$ ۲۳٫۹
	۵۰	۶٫۳۸ h	۶٫۳۱ h	۶٫۴۳ h	\$ ۲۹٫۲۱	\$ ۵۴٫۸۹
	۱۰۰	۸٫۱ h	۶٫۸۱ h	۶٫۹۱ h	\$ ۶۲٫۰۶	\$ ۱۰۱٫۹

در شکل ۲ مقایسه زمان اجرای الگوریتم پایه و پیشنهادی برای گراف‌های Epigenomics و Montage با ضریب  $\alpha=1.6$  نشان داده شده است. نتایج نشان می‌دهد که برای گراف Montage روش CCA موفق عمل نکرده است و دو مورد از جریان‌های کاری را در زیرمهلته زمانی تعیین شده اجرا نمی‌کند. اما با افزایش ضریب مهلت، روش پیشنهادی هر سه جریان کاری را در مهلت تعیین شده اجرا می‌کند.

<sup>1</sup> cluster combining algorithm

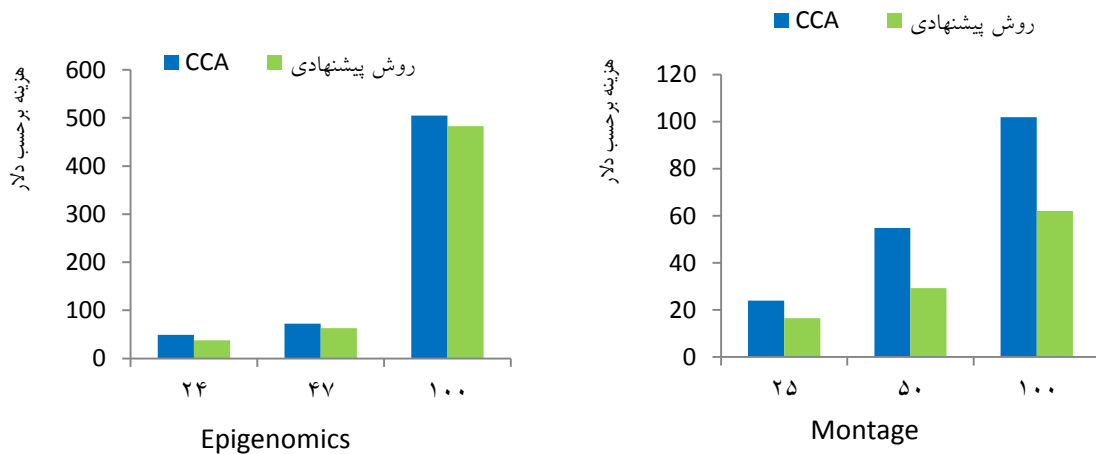
## سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم



شکل (۲): زمان اجرای الگوریتم پیشنهادی و CCA برای گراف Epigenomics و Montage با ضریب  $\alpha=1.6$

همچنین برای گراف Epigenomics نتایج نشان می‌دهد که هر دو روش پایه و پیشنهادی با افزایش ضریب مهلت، هر سه جریان کاری را در مهلت تعیین شده اجرا می‌کنند.

در شکل ۳ مقایسه هزینه اجرای الگوریتم پایه و پیشنهادی برای گراف‌های Epigenomics و Montage با ضریب  $\alpha=1.6$  نشان داده شده است. با افزایش زیرمهلت زمانی، هزینه اجرای جریان‌های کاری کاهش می‌یابد. نمودارها برای هر دو گراف نشان می‌دهد که روش پیشنهادی هزینه اجرای جریان‌های کاری را در زیر مهلت تعیین شده نسبت به روش CCA بیشتر کاهش داده است.



شکل (۳): هزینه اجرای الگوریتم پیشنهادی و CCA برای گراف Epigenomics و Montage با ضریب  $\alpha=1.6$

### ۸- نتیجه گیری

در این مقاله الگوریتمی ارائه شده است که یک جریان کاری را دریافت می‌کند، سپس بر روی ابرعمومی زمانبندی می‌کند. هدف این الگوریتم زمانبندی جریان کاری بر روی منابع ناهمگون ابر و اجرای کلیه وظایف در زیرمهلت تعیین شده با کمترین هزینه ممکن می‌باشد. این الگوریتم ابتدا وظایف را رتبه بندی می‌کند، سپس وظایف را برحسب رتبه پایین انتخاب و براساس والد بحرانی خوشه بندی می‌کند. در انتها وظایف خوشه‌ها به ترتیب شماره خوشه، به بهترین منابع ممکن تخصیص می‌یابند. آزمایش‌های انجام شده بر روی دو جریان کاری واقعی Epigenomic و Montage انجام شد. آزمایش‌ها با ضریب مهلت زمانی  $\alpha=1.6$  انجام می‌شود. با توجه به آزمایش‌های انجام شده الگوریتم پیشنهادی در زیرمهلت مشخص اجرایی شود



و هزینه اقتصادی پایین تری از الگوریتم CCA دارد. نتایج نشان می دهند که روش پیشنهادی برای اجرای جریان های کاری بطور متوسط هزینه کمتری را به کاربر تحمیل می کند. روش پیشنهادی ۱۵ درصد هزینه را نسبت به روش پایه کاهش داده است. در کار آتی چندین جریان کاری همزمان زمان بندی می شوند.

### ۱۰-مراجع

- [1]. T. White, Hadoop: The Definitive Guide, O'Reilly, 2010.
- [2]. Tomasz Kajdanowicz , Przemyslaw Kazienko, Wojciech Indyk Parallel processing of large graphs Published by Elsevier B.V. All rights reserved. dx.doi.org/10.1016/08.007/j.future.2013.
- [3]. Juve, G., Deelman, E., Vahi, K., Mehta, G., Berriman, B., Berman, Scientific workflow applications on Amazon EC2. In 5th IEEE international conference on e-Science Maechling, P. (2010)..
- [4]. Juve, G., Chervenak, A., Deelman, E., Bharathi, S., Mehta, Vahi, K. Characterizing and profiling scientific workflows. Future Generation Computer Systems, 29(3), 682–692. 2012.
- [5]. Mao, M., & Humphrey, Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In Proceedings of 2011 international conference for high performance computing, networking, storage and analysis, Seattle, Washington (pp. 1–49). 2011
- [6]. Wu, F, Wu, Q., & Tan, Workflow scheduling in cloud: A survey. The Journal of Supercomputing, 71(9), 3373–3418. 2015
- [7]. Mell, P., & Grance, The NIST definition of cloud computing. Gaithersburg: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology. 2011
- [8]. Hoffa, C., Mehta, G., Freeman, T., Deelman, On the use of cloud computing for scientific workflows. In Proceedings of the Fourth IEEE international conference on eScience (pp).640–645. 2008
- [9]. Juve, G., & Deelman, Scientific workflows in the cloud. In M. Cafaro & G. Aloisio (Eds.), Grids, clouds and virtualization (pp. 71–91). New York: Springer 2011
- [10]. Jyoti Sahni, Deo Prakash Vidyarthi "A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment" 2168-7161 (c) 2015 IEEE.
- [11]. Topcuouglu, H., Hariri, S., & Wu, Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Transaction Parallel Distributed Systems, 13(3), 260–274. 2002
- [12]. Mainak Adhikari, Tarachand Amgoth " Efficient Algorithm for Workflow Scheduling in Cloud Computing Environment" 978-1-5090-3251-8/16/2016 IEEE
- [13]. Narendrababu Reddy, G, Dr. S. Phani Kumar " Time and Cost-Aware Method for Scheduling Workflows In Cloud Computing Systems" 978-1-5090-4715-4/17/2017 IEEE
- [14]. Arash Deldari, Mahmoud Naghibzadeh, Amin Rezaeian, Hamidreza Abrishami " A Clustering Approach to Schedule Workflows to Run on the Cloud" 978-1-5090-4335-4/16/2016 IEEE
- [15]. L. F. Bittencourt and E. R. M. Madeira, "Towards the scheduling of multiple workflows on computational grids," J. grid Comput., vol. 8, no. 3, pp. 419–441, 2010
- [16]. Kanchana Viriyapant, Sucha Smanchat " A Deadline-constrained Scheduling for Dynamic Multi-instances Parameter Sweep Workflow" 978-1-5090-0806-3/16/ copyright 2016 IEEE
- [17]. Vahid Arabnejad, Kris Bubendorfer, Bryan Ng and Kyle Chard "A Deadline Constrained Critical Path Heuristic for Cost-effectively Scheduling Workflows" 978-1-4503-3890-5/15 / 2015 IEEE
- [18]. Yingchi Mao, Haishi Zhong, Xiaofang Li " Hierarchical Model-based Associate Tasks Scheduling with the Deadline Constraints in the Cloud " 978-1-4673-9104-7/15/ 2015 IEEE
- [19]. K. Kanagaraj, S. Swamynathan " A Realistic Approach for Representing and Scheduling Workflows in Cloud Computing Environment " 978-1-5090-2029-4/16/ 2016 IEEE
- [20]. Ting Sun, Chuangbai Xiao, Xiujie Xu, " A scheduling algorithm using sub-deadline for workflow applications under budget and deadline constrained" Springer Science+Business Media, LLC, part of Springer Nature 2018
- [21]. Arash Deldari, Mahmoud Naghibzadeh, Saeid Abrishami "CCA: a deadline-constrained workflow scheduling algorithm for multicore resources on the cloud" © Springer Science+Business Media New York 2016
- [22]. Bryk, P., Malawski, M., Juve, G., & Deelman, Storage-aware algorithms for scheduling of workflow ensembles in clouds. Journal of Grid Computing, 14(2), 359–378. 2016
- [23]. Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M. H., & Vahi, Characterization of scientific workflows. In 2008 third workshop on workflows in support of large-scale science (pp. 1–10). 2008