7th International Conference on Web Research
May 19-20, 2021; Tehran, IRAN ICWR2021

Multi-Constraint Dynamic Scheduling of Scientific Workflows in Cloud with Multi-Resource Packing

Ehsan Saeedizade[1], Mehrdad Ashtiani[2]*

[1] Master Student of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
e24saeedi@gmail.com

[2] Assistant Professor, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
m_ashtiani@iust.ac.ir

Abstract

Scientific workflows are large compute-intensive applications that consist of different smaller parts called tasks, connected with specific structures to each other, to process data. Recently, numerous researches have been focused on solving the workflow scheduling problem in the cloud. Since the cloud providers offer scalable and cost-effective computational resources to their users and charge them based on a pay-as-you-go model, user's requests to run workflows in such an environment increase. Consequently, this brings the need for environments like Workflow as a Service platform. In these environments, users can submit workflows dynamically with their quality of service requirements. Usually, users specify these requirements in terms of time and cost. However, considering all these factors the scheduling problem is challenging.

In this paper, we propose a multi-constraint dynamic workflow scheduling algorithm with multi-resource packing. Unlike existing algorithms, we consider both deadline and budget as the quality of service requirements simultaneously. Commonly the researchers consider CPU demand in their proposed approach for simplicity. However, in addition to CPU demand, we consider memory demand in this paper. The proposed algorithm leverages containers to shares VMs among different users and reduces the overall execution cost. In the proposed algorithm, we also introduce a bi-factor to tradeoff between cost and resource utilization while mapping tasks to resources. The evaluation results show that the proposed algorithm achieves at least 96% Planning Success Rate under different workloads. Also, the experiments show that the proposed algorithm outperforms the EPSM and MW-HBDCS workflow scheduling algorithms in terms of overall execution cost and total leased VM number. It decreased the overall execution cost by at least 13% and on average 33.2% under different workloads.

Keywords: Workflow as a Service, Workflow Scheduling, Multi-Resource Packing, Cloud Computing, Quality of Service, Dynamic Scheduling.

7th International Conference on Web Research
May 19-20, 2021; Tehran, IRAN ICWR2021

زمان‌بندی چند محدودیته و پویا جریان‌های‌کار علمی در ابر به کمک بسته‌بندی چند منبع

احسان سعیدی‌زاده[۱]، مهرداد آشتیانی[۲*]

[۱] کارشناسی ارشد مهندسی نرم افزار، دانشگاه علم و صنعت ایران، تهران،
e24saeedi@gmail.com

[۲] استادیار، دانشکده کامپیوتر، دانشگاه علم و صنعت ایران ، تهران،
m_ashtiani@iust.ac.ir

چکیده

جریان‌های‌کار علمی، دنباله‌ای از محاسبات هستند که پردازش داده‌های حجیم را در یک فرم ساختاریافته را ممکن می‌سازند. برای پردازش این جریان‌های‌کار به منابع زیادی نیاز است. هریک از آن‌ها ازنظر منابع مورد نیاز ویژگی خاصی دارد. پژوهش‌های زیادی در حوزه زمان‌بندی جریان‌کار در محیط‌های مختلف از جمله ابر انجام شده است. ابر یک محیط مقیاس‌پذیر و اقتصادی است که به کاربرها امکان دسترسی به منابع محاسباتی نامحدود با مدل هزینه، پرداخت به‌میزان استفاده را می‌دهد. افزایش انگیزه کاربرها در اجرای جریان‌های‌کار در محیط ابری باعث توسعه بسترهای چندمستأجره‌ای مانند بستر جریان‌کار به‌عنوان سرویس شده است. این بستر محیطی را ارائه می‌دهد که کاربرها به‌راحتی می‌توانند جریان‌های‌کار خود را با تعیین کیفیت سرویس موردنظر خود برای اجرا ثبت کنند. زمان و هزینه معروف‌ترین انواع کیفیت سرویس هستند. در این مقاله ما الگوریتمِ زمان‌بندیِ چند محدودیتی و پویا را برای محیط جریان‌کار به‌عنوان سرویس، معرفی می‌کنیم . برخلاف الگوریتم‌های معرفی‌شده در این حوزه، در الگوریتم پیشنهادی دو کیفیت سرویس فرجه و بودجه را به طور همزمان در نظر گرفته شده است. معمولاً پژوهشگرها با هدف ساده‌کردن مسئله، تنها نیازمندی توان پردازشی را برای محاسبات یک جریان‌کار در نظر گرفته‌اند اما در روش پیشنهادی علاوه بر نیازمندی توان پردازشی، نیازمندی حافظه نیز در نظر گرفته می‌شود. همچنین ، در الگوریتم پیشنهادی با بهره‌گیری از کانتینر، امکان اشتراک منبع بین کاربرها و اجرای چند وظیفه به طور همزمان روی آن و در نهایت کاهش هزینه وجود دارد. این الگوریتم از یک معیار دو-فاکتور برای کنترل و سبک-وسنگین کردن بین هزینه و بهره‌وری منابع در حین نگاشت وظیفه‌ها به منابع استفاده می‌کند. نتایج حاصل از آزمایش‌ها نشان می‌دهدکه نرخ موفقیت الگوریتم در بارهای کاری مختلف بالای ۹۶٪ است. همچنین این الگوریتم در بارهای کاری متفاوت در مقایسه با دو الگوریتم زمان‌بندی EPSM و MW-HBDCS هزینه کل را حداقل ۱۳.۲٪ و به طور میانگین ۳۳.۲٪ کاهش داده و از نظر تعداد ماشین‌های مجازی اجاره‌شده نیز بهتر عمل می‌کند.

کلمات کلیدی

جریان‌کار به‌عنوان سرویس، زمان‌بندی جریان‌کار، بسته‌بندی چند منبع، پردازش ابری، کیفیت سرویس، زمان‌بندی پویا

۱- مقدمه

برنامه‌های علمی معمولاً از مجموعه‌ای از وظیفه‌ها با وابستگی داده‌ای بین آن‌ها تشکیل شده‌اند و با یک گراف جهت‌دار بدون دور مدل می‌شوند. جریان‌های‌کار علمی پردازش و تحلیل داده‌های حجیم در فرم ساختاریافته و توزیع‌شده را ممکن می‌سازند. اهمیت پردازش جریان‌های‌کار در حوزه پردازش داده‌های حجیم و استخراج اطلاعات مفید از میان داده‌ها در دوره‌ای که ابزار-های مختلف به‌طور پیوسته در حال تولید این داده‌ها هستند، بیش از پیش قابل‌ملاحظه است. مسئله زمان‌بندی این جریان‌ها با توجه به ماهیت بزرگ بودن ابعاد محاسبات در جریان‌های‌کار علمی و ویژگی خاص هرکدام از آن‌ها، از جهت تمرکز بر نوع منابع و کیفیت سرویس مورد نیاز برای پردازش، یک مسئله مهم است. همچنین جریان‌های‌کار علمی ذاتاً دارای مهلت زمانی برای اجرا هستند. در زمان‌بندی جریان‌کار، به دنبال پیداکردن یک نگاشت از این وظیفه‌ها به منابع محاسباتی هستیم، به‌طوری‌که نیازمندی‌ها و کیفیت سرویس مورد نیاز آن‌ها رعایت شود. به دلیل کیفیت سرویس متفاوت و ناهمگونی منابع در محیط اجرا، مسئله زمان‌بندی جریان‌کار در کلاس NP-complete قرار دارد و پیداکردن یک جواب بهینه برای این مسئله در زمان چندجمله‌ای امکان‌پذیر نیست [1]. ازاین‌رو، ارائه یک الگوریتم زمان‌بندی مناسب، یک چالش محسوب می‌شود.

پژوهش‌های زیادی در حوزه زمان‌بندی جریان‌کار برای خودکارسازی اجرای آن‌ها در محیط‌های توزیع‌شده و رسیدن به یک جواب نزدیک به بهینه وجود دارد. در بین همه‌ی این پژوهش‌ها، بخش بزرگی از آن‌ها زمان و هزینه را به‌عنوان دو هدف اصلی در نظر گرفته‌اند. برخی از پژوهش‌ها یکی از این دو هدف را به‌عنوان محدودیت در مسئله در نظر می‌گیرند و به دنبال بهینه‌سازی هدف دوم هستند [2, 3, 4]. برخی دیگر نیز مسئله زمان‌بندی جریان‌کار را به عنوان یک مسئله ارضا محدودیت در نظر می‌گیرند. آن‌ها هر دو هدف زمان و هزینه را به عنوان محدودیت در مسئله در نظر می‌گیرند [5, 6, 7]. بعضی از الگوریتم‌های موجود ایستا هستند و برای زمان‌بندی یک جریان‌کار طراحی شده‌اند [8, 9, 2]. همچنین در سال‌های اخیر بعضی از پژوهش‌ها زمان‌بندی پویا چند جریان‌کار و یا دسته‌های جریان‌کار را در محیط ابر تحقیق کرده‌اند [10, 11, 12]. با توجه به رفتار الگوریتم می‌توان گفت زمان‌بندی ایستا، یک برنامه زمان‌بندی در ابتدای کار ایجاد می‌کند و با توجه به برنامه تولیدشده وظایف به منابع محاسباتی نگاشت خواهند شد. در روش‌های پویا الگوریتم زمان‌بندی وضعیت سیستم و اجرای وظایف را دیده‌بانی می‌کنند و هر زمان که نیاز باشد، یک برنامه زمان‌بندی جدید را تولید کرده و یا برنامه تولیدشده قبلی را اصلاح می‌کند.

پردازش ابری یک مدل جدید است که منابع نامحدود و مقرون‌به‌صرفه را با مدل پرداخت به‌میزان استفاده به کاربرها ارائه می‌دهد. در این محیط پردازشی، کاربرها می‌توانند از بین انواع منابع با توجه به نیاز خود انتخاب کنند. به همین دلیل اخیراً برای اجرای جریان‌کار مورد توجه کاربرها قرار گرفته است و در نتیجه نیاز به یک بستر مناسب برای این منظور افزایش یافته است. به همین دلیل در سال‌های اخیر بعضی از پژوهش‌ها روی زمان‌بندی جریان‌کار در محیط جریان‌کار به عنوان سرویس تمرکز کرده‌اند. آن‌ها جریان‌کار به‌عنوان سرویس را به عنوان یک بستر در نظر گرفته‌اند که در آن کاربرها به صورت پویا جریان‌های‌کار را با نیازمندی کیفیت سرویس مورد نظر برای اجرا در این بستر ثبت می‌کنند. بیشتر الگوریتم‌های پیشنهادی برای محیط جریان‌کار به-عنوان سرویس، دو کیفیت سرویس بودجه و فرجه را به طور همزمان در نظر نگرفته‌اند. همچنین یک فرض مشترک بین روش‌های زمان‌بندی مشابه این است در هر لحظه از زمان تنها یک وظیفه روی منبع محاسباتی در حال اجرا است. همچنین از جهت مدل منابع اغلب در الگوریتم‌های پیشین، ماشین مجازی به‌عنوان منبع محاسباتی در نظر گرفته‌شده است. تعدادی از روش‌های جدیدتر نیز از کانتینر به‌عنوان منبع محاسباتی استفاده می‌کنند. همچنین هیچ‌کدام به طور همزمان نیازمندی منابع پردازشی و حافظه وظایف یک جریان‌کار را در نظر نمی‌گیرند. در نتیجه در این مقاله یک الگوریتم زمان‌بندی پویا به نام DDBSW2 با درنظرگرفتن چند محدودیت برای محیط جریان‌کار به‌عنوان سرویس طراحی شده است. به طور کلی برای مسئله زمان‌بندی با درنظرگرفتن چند محدودیت، روش‌های فرااکتشافی استفاده می‌شوند. اما در حالت زمان‌بندی پویا چند جریان‌کار ، به دلیل زمان‌بر بودن و هزینه روش‌های فرااکتشافی ، استفاده از این روش‌ها معقول نیست. به همین دلیل الگوریتم پیشنهادی یک الگوریتم اکتشافی است. این الگوریتم با هدف کاهش هزینه نهایی و تعداد ماشین‌های مجازی اجاره شده، مسئله زمان‌بندی را به کمک بسته‌بندی چند منبع و اجرای چند وظیفه روی یک ماشین‌مجازی حل می‌کند.

در الگوریتم DDBSW2 فرض شده است که کاربرها در زمان متفاوت جریان‌کار خود را با محدودیت فرجه و بودجه در محیط ثبت می‌کنند. جریان-کار پس از فاز پیش‌پردازش برای اجرا به استخر جریان‌کار منتقل می‌شود. در هر چرخه زمان‌بندی و اجرای الگوریتم ، از هر جریان‌کار یک وظیفه آماده اجرا با بالاترین اولویت انتخاب می‌شود. همه وظیفه‌های انتخاب شده در صف وظایف آماده اجرا قرار می‌گیرند. پس از مرتب کردن این صف، الگوریتم وظیفه‌ها را با توجه به نیازمندی آن‌ها، روی ماشین‌های مجازی زمان‌بندی می‌کند. در ادامه در این مقاله تعدادی از الگوریتم‌های مرتبط با این حوزه، در بخش دوم بررسی می‌شوند. در بخش سوم برخی از مفاهیم اولیه مورد نیاز مرور می‌شوند. در بخش چهارم چارچوب مسئله زمان‌بندی معرفی می‌شود. جزئیات الگوریتم زمان‌بندی پیشنهادی در بخش پنجم ارائه شده است. در بخش ششم کارایی الگوریتم زمان‌بندی در مقایسه با دو الگوریتم دیگر ارزیابی شده است. در انتها نتیجه و کارهای آینده برای توسعه الگوریتم پیشنهادی نیز در بخش هفتم این مقاله ارائه شده است.

۲- کارهای مرتبط

ماریا و راجکومار یک الگوریتم زمان‌بندی اکتشافی برخط و پویا به نام EPSM برای زمان‌بندی در بستر جریان‌کار به‌عنوان سرویس معرفی کرده‌اند.

به گفته آن‌ها این اولین الگوریتم معرفی‌شده به‌منظور استفاده در این بستر و اشتراک منابع است. هدف اصلی این الگوریتم افزایش بهره‌وری و کاهش هزینه ضمن وجود فرجه به‌عنوان محدودیت زمان اجرا است.در این الگوریتم برای اشتراک ماشین‌های مجازی از کانتینر استفاده شده است. همچنین فرض کرده‌اند که در هر زمان تنها یک کانتینر روی یک منبع اجرا می‌شود[13]. در مقایسه با الگوریتم پیشنهادی دو محدودیت فرجه و بودجه نیز در نظر گرفته شده است، این الگوریتم تنها فرجه را برای زمان‌بندی در نظر گرفته است. در این الگوریتم برای هر وظیفه تنها نیازمندی پردازنده در نظر گرفته شده است. همچنین در هر لحظه تنها یک وظیفه روی منبع محاسباتی در حال اجرا است و همه توان محاسباتی یک منبع در اختیار یک وظیفه قرار می‌گیرد. درحالی که در الگوریتم پیشنهادی برای هر وظیفه دو نیازمندی پردازنده و حافظه در نظر گرفته می‌شود. همچنین به کمک استفاده از کانتینر، امکان اجرای چند وظیفه روی یک منبع محاسباتی به طور هم‌زمان بسته‌بندی وظیفه‌ها روی منابع وجود دارد. اما در الگوریتم پیشنهادی با هدف استفاده حداکثری از فرجه تعیین‌شده و کاهش هزینه، کم‌ترین میزان توان محاسباتی در اختیار وظایف قرار می‌گیرد به طوری که در فرجه خودشان اجرا شوند.

الگوریتم MW-HBDCS یک الگوریتم اکتشافی و پویا برای زمان بندی جریان‌های‌کار به صورت هم‌زمان و با محدودیت فرجه و بودجه است. وظایف از جریان‌های‌کار مختلف با معیارهای جدیدِ معرفی‌شده در مرجع [14] اولویت‌بندی می‌شوند. برای محاسبه اولویت و انتخاب منابع از یک رویکرد ترکیبی بر اساس میزان فرجه و بودجه باقی‌مانده استفاده می‌کند. به کمک این رویکرد در زمان‌های مختلف به‌راحتی بین این دو نیازمندی سبک‌وسنگین شود. در این الگوریتم نیز در هر لحظه تنها یک وظیفه روی منبع محاسباتی اجرا می‌شود و منابع به طور مشترک برای اجرای هم‌زمان وظیفه‌ها استفاده نمی‌شوند. درحالی که الگوریتم پیشنهادی با استفاده کانتینر، منابع را بین چند وظیفه به اشتراک می‌گذارد که این موضوع خود باعث کاهش هزینه می‌شود. این الگوریتم به دلیل این که در حین زمان‌بندی فرجه و بودجه را به صورت پویا برای وظایف به‌روزرسانی نمی‌کند، نسبت به بار بالا روی سیستم و تغییرات کارایی و کاهش توان پردازشی منبع محاسباتی حساس است. در حالی که به دلیل رفتار پویا در الگوریتم پیشنهادی، این حساسیت به بار بالا کم‌تر است.

الگوریتم NOSF یک الگوریتم زمان‌بندی آگاه از فرجه و پویا برای زمان‌بندی چند جریان‌کار به طور هم‌زمان با زمان اجرا غیرقطعی است. هدف این الگوریتم افزایش بهره‌وری ماشین‌های مجازی و کاهش هزینه نهایی اجرای جریان‌کار است [15]. این الگوریتم برای استفاده در بستر جریان‌کار به‌عنوان سرویس مناسب است اما تفاوت اصلی آن با الگوریتم پیشنهادی درنیازمندی کیفیت سرویس، مدل منابع و نحوه اجرای وظایف روی ماشین مجازی است. این الگوریتم تنها فرجه را در نظر می‌گیرد اما الگوریتم پیشنهادی دو محدودیته است و فرجه و بودجه را در نظر می‌گیرد. در الگوریتم پیشنهادی با هدف کاهش هزینه وظایف روی کانتینرها اجرا و روی منابع محاسباتی بسته‌بندی می‌شوند اما این الگوریتم در هر لحظه تنها یک وظیفه روی هر ماشین مجازی در حال اجرا است.

۳- مفاهیم اولیه

در این بخش برای فهم بهتر مطالب در ادامه این مقاله برخی مفاهیم اولیه توضیح داده شده اند.

۱. رتبه رو به بالا: رتبه روبه‌بالا برای یک وظیفه در گراف جریان‌کار، اندازه طولانی‌ترین مسیر از آن وظیفه تا گره خروجی است.
۲. عدم قطعیت: به‌طورکلی رفتارها و اتفاقات غیرقابل‌پیش‌بینی مانند تغییرات کارایی پردازنده یا تغییرات پهنای باند و یا تأخیر در گرفتن منابع پردازشی که مستقیم یا غیر مستقیم بر روی کارایی الگوریتم زمان‌بندی تأثیرگذار هستند، می‌تواند مثالی برای عدم قطعیت باشد.
۳. زمان اجرا: زمان اجرای هر جریان‌کار برابر با اختلاف بین بزرگ‌ترین زمان پایان از بین وظیفه‌های خروجی و کوچک‌ترین زمان شروع از بین وظیفه‌های ورودی در گراف آن جریان‌کار است.
۴. فرجه: حداکثر مدتی که اجرای جریان‌کار در آن زمان باید انجام شود.
۵. بودجه: حداکثر هزینه قابل پرداخت برای اجرای جریان کار.

٤- معرفی چارچوب زمان‌بندی

در این بخش به طرح مسأله و توضیح چارچوب زمان‌بندی، مدل جریان‌کار و منابع، نحوه نگاشت وظیفه‌ها به منابع، اهداف و محدودیت‌ها می‌پردازیم.

٤-١- مدل جریان‌کار

در این مقاله یک جریان‌کار به کمک یک گراف جهت‌دار بدون دور و بدون وابستگی شرطی $w = \{T, E\}$ توصیف می‌شود. در توصیف گراف جریان‌کار، $T = \{t_1, t_2, \ldots t_n\}$ مجموعه‌ای از وظیفه‌های محاسباتی است. هر وظیفه دارای یک اندازه مشخص و درخواست $D_{ti} = (d^1_{ti}, d^2_{ti})$ از نظر تعداد هسته‌های مورد نیاز و حافظه است. $E = \{e_1, e_2, \ldots e_m\} \subseteq T \times T$ یک مجموعه از لبه‌های جهت‌دار است. زمانی که یک لبه بین وظیفه t_i و وظیفه t_j وجود دارد، این لبه یک وابستگی داده‌ای را بین این دو وظیفه نشان می‌دهد. به‌این‌ترتیب $parents\ (t_i)$ و $children\ (t_i)$ به ترتیب مجموعه همه پدرها و فرزندهای وظیفه i را نشان می‌دهد. dt_{in} و dt_{out} نیز سایز اطلاعات ورودی مورد نیاز و اطلاعات خروجی یک هستند.

٤-٢- مدل منابع محاسباتی

در الگوریتم پیشنهادی از ماشین مجازی به‌عنوان منبع محاسباتی استفاده شده است. در این پژوهش در محاسبه ظرفیت پردازشی یک ماشین مجازی علاوه بر تعداد هسته‌ها، میزان حافظه آن ماشین نیز در نظر گرفته شده است. نوآوری مدل منابع در این پژوهش در این است که چند وظیفه به طور هم‌زمان روی

یک ماشین مجازی اجرا خواهند شد. ظرفیت یک ماشین مجازی با دوتایی $Cap(C_v,M_v)$ مشخص می‌شود که در آن C_v تعداد کل هسته‌ها و M_v اندازه حافظه ماشین مجازی است. هم‌چنین فرض شده است که وظایف در داخل کانتینرها اجرا خواهند شد. در الگوریتم پیشنهادی از مدل پرداخت به‌میزان استفاده برای محاسبه هزینه اجرا استفاده شده است. در این مدل، هزینه بر اساس دوره پرداخت ρ محاسبه می‌شود. هزینه اجرای وظیفه t روی کانتینر c و ماشین مجازی v با توجه به فرمول (۱) محاسبه می‌شود.

$$Cost(t,c,v) = \frac{\pi\ c}{\pi\ v} * Price_v * \left\lceil \frac{Proc_t}{\rho_v} \right\rceil \quad (۱)$$

در این فرمول π_c برابر با تعداد هسته‌های تخصیص‌داده‌شده به کانتینر و π_v برابر با تعداد کل هسته‌های ماشین‌مجازی، $Price_v$ هزینه ماشین v برای هر دوره پرداخت، $Proc_c$ زمان پردازش وظیفه t و $Cost(t, c, v)$ هزینه اجرای وظیفه t روی کانتینر c و ماشین v است.

زمان اجرا یک وظیفه Exe_t با اندازه I_t در قالب تعداد میلیون دستور روی یک کانتینر با توان پردازشی P_{ct} با توجه به فرمول (۲) محاسبه می‌شود.

$$Exe_t = \frac{I_t}{P_{ct}} \quad (۲)$$

در نهایت زمان پردازش یک وظیفه با توجه به فرمول (۳) برابر با حاصل جمع زمان اجرا و کل زمان مورد نیاز برای انتقال داده‌ها ورودی از روی منبع ذخیره مرکزی به ماشین مجازی و انتقال داده خروجی به روی آن منبع است.

$$Proc_t = Exe_t + \frac{\sum_{f\in In_t} Size\ f}{bwt_v} + \frac{\sum_{f'\in Out_t} Size\ f'}{bwt_v} \quad (۳)$$

در این فرمول In_t و Out_t به ترتیب مجموعه داده‌های ورودی و خروجی وظیفه t و bwt_v پهنای باند ماشین v است.

٤-٣- نگاشت چند وظیفه به منبع

به کمک استفاده از کانتینر، می‌توانیم برنامه‌های مختلفی را به طور همزمان روی یک میزبان اجرا کنیم. در این حالت می‌توانیم توان پردازشی میزبان را بین برنامه‌ها به اشتراک بگذاریم [16]. در یک ابر قیمت ماشین‌های مجازی ارائه‌شده توسط ارائه‌دهنده سرویس زیرساخت در مقایسه با توان پردازشی لزوماً به‌صورت خطی افزایش پیدا نمی‌کند. ممکن است استفاده از یک ماشین‌مجازی بزرگتر در مقایسه با اجاره چند ماشین‌مجازی کوچکتر، هزینه نهایی کمتری داشته باشد. همچنین تأخیر در ساختن ماشین مجازی می‌تواند زمان اجرا یک جریان‌کار را تحت تأثیر قرار دهد. با ساختن یک ماشین مجازی با توان بیشتر، می‌توانیم اثر این تأخیرها را در سیستم کاهش بدهیم. ممکن است که ارائه‌دهنده زیرساخت به‌عنوان سرویس، یک محدودیت به عنوان حداکثر تعداد ماشین‌های مجازی قابل ساخت برای کاربران خود تعیین کند. یکی دیگر از مزایای استفاده از این رویکرد این است که در چنین مواردی این امکان وجود دارد که با اجاره ماشین‌های مجازی با توان پردازشی بیش‌تر و به اشتراک‌گذاری آن‌ها بین وظایف، به هم‌روندی بالاتری در اجرای جریان‌های‌کار برسیم.

٤-٤- برنامه زمان‌بندی

در این مقاله یک برنامه زمان‌بندی معتبر از سه نگاشت زیر تشکیل شده است:
T2C: T → C.
C2V: C → V.
TYPE: V → PP.
در این نگاشت‌ها T مجموعه همه وظایف از همه جریان‌های‌کار است. C مجموعه همه کانتینرها است و $T2C$ یک نگاشت از وظایف به این کانتینرها است. V مجموعه همه ماشین‌های مجازی و $C2V$ یک نگاشت از کانتینرها به ماشین‌های مجازی است. $TYPE$ نیز یک نگاشت از هر ماشین مجازی به نوع PP است.

٤-٤-١- اهداف زمان‌بندی

در این مقاله هدف اصلی الگوریتم زمان‌بندی رعایت فرجه و بودجه تعیین‌شده است. همچنین هدف بعدی الگوریتم زمان‌بندی کاهش هزینه نهایی اجرا به کمک استفاده اشتراکی از منابع محاسباتی است. همچنین در نگاشت وظیفه به منبع نیز برای بالا نگه‌داشتن بهره‌وری منابع نیز تلاش می‌شود. منظور از هزینه نهایی اجرا، هزینه پرداخت شده توسط مشتری‌ها به ارائه‌دهنده سرویس در ازای اجرای جریان‌های‌کار است. این هزینه مطابق با فرمول (۴) محاسبه می‌شود.

$$\sum_{i=1}^{n} Cost\ W_i \quad (۴)$$

که در آن $Cost\ W_i$ هزینه اجرای جریان‌کار W_i است و هزینه اجرای هر جریان‌کار نیز برابر مجموع هزینه اجرای همه وظایف آن مطابق با فرمول (۱) محاسبه می‌شود. هزینه نهایی اجرا به معنی جمع هزینه اجرای همه جریان‌های‌کار ثبت‌شده در سیستم است.

٤-٤-٢- محدودیت‌های زمان‌بندی

یک محدودیت در الگوریتم زمان‌بندی این است که در هر لحظه از زمان مجموع درخواست وظیفه‌های در حال اجرا روی یک ماشین مجازی از حداکثر ظرفیت آن ماشین بیشتر نباشد. محدودیت دیگری باید در حین زمان‌بندی به

آن توجه شود، وابستگی داده‌ای بین وظایف در یک جریان‌کار است. این به این معنی است که اجرای یک وظیفه قبل از اتمام تمام پدرهایش و انتقال اطلاعات مورد نیازش نمی‌تواند شروع شود.

۵- الگوریتم زمان‌بندی DDBSW2

در این بخش جزئیات الگوریتم زمان‌بندی پیشنهادی را شرح می‌دهیم.

۵-۱- پیش‌پردازش

در این فاز رتبه روبه‌بالا بر اساس فرمول (۵) برای هر وظیفه محاسبه می‌شود. که در آن $rank_t$ رتبه روبه‌بالا برای آن وظیفه، $Exe_{ave\,t}$ میانگین زمان اجرای وظیفه روی همه انواع منابع و $\overline{ct, t'}$ میانگین زمان جابه‌جایی داده بین دو وظیفه است. بر اساس این رتبه‌ها، فرجه تعیین‌شده D_w توسط کاربر مطابق فرمول (۶) بین وظیفه‌ها توزیع می‌شود. در نهایت بودجه نیز بین وظیفه‌ها توزیع می‌شود. برای این منظور با محاسبه هزینه مصرف شده کل بودجه باقی‌مانده محاسبه می‌شود. سپس این مقدار به نسبت میانگین هزینه هر وظیفه، بین آن‌ها توزیع می‌شود. در انتها جریان‌کار برای زمان‌بندی و اجرا به استخر جریان‌کار اضافه می‌شود. به منظور پویایی مراحل توزیع فرجه و بودجه با اتمام اجرای هر وظیفه تکرار می‌شوند و فرجه و بودجه برای وظیفه‌های باقی‌مانده در آن جریان‌کار به‌روزرسانی خواهد شد.

$$rank_t = \begin{cases} Exe_{ave\,t} & if\ \ children(t) = \emptyset \\ Exe_{ave\,t} + Max_{t' \in schildren(t)} rank_{t'} + \overline{ct, t'} \end{cases} \quad (۵)$$

$$\delta_t = \frac{\left(rank_{t\,entry} - rank_t + Exe_{ave\,t}\right)}{rank_{t\,entry}} * D_w \quad (۶)$$

۵-۲- زمان‌بندی

الگوریتم زمان‌بندی به صورت دوره‌ای اجرا می‌شود. در هر دوره سه مرحله زیر را انجام می‌دهد.

۵-۲-۱- جمع‌آوری وظایف

در این مرحله الگوریتم به دنبال آن است که وظایف آماده اجرا را از همه جریان‌های‌کار جمع‌آوری کند. به این منظور وظیفه‌های هر جریان کار بر اساس زیرفرجه به صورت صعودی مرتب و اولویت بندی می‌شوند. سپس الگوریتم برای رعایت انصاف از هر جریان‌کار یک وظیفه با بالاترین اولویت را انتخاب می‌کند. وظیفه‌های این صف بر اساس زیرفرجه مرتب می‌شوند.

۵-۲-۲- انتخاب منبع محاسباتی

در این مرحله الگوریتم حداقل تعداد هسته‌های پردازشی مورد هر وظیفه را طوری تخمین می‌زند که اجرای آن قبل از زیرفرجه آن به اتمام برسد. برای کاهش تأثیر ناشی از تأخیر گرفتن ماشین مجازی و کانتینر، ترجیح الگوریتم بر این است که وظیفه‌ها را روی کانتینرهای در حال اجرا زمان‌بندی کند.

اگر الگوریتم نتواند کانتینر مناسبی پیدا کند، در مرحله بعد به دنبال ساختن یک کانتینر جدید روی یک ماشین‌مجازی در حال اجرا است. بنابراین همه ماشین‌های مجازی مناسب برای اجرای وظیفه را جمع‌آوری می‌کند. سپس این مجموعه را با توجه به زیربودجه وظیفه به دو دسته قابل استطاعت و غیر قابل استطاعت تقسیم می‌کند. همچنین ماشین‌های مجازی در هر دو دسته با توجه به اینکه داده ورودی روی آن‌ها وجود دارد یا خیر، به دو زیر مجموعه جدید تقسیم می‌شوند. این کار برای کاهش تأثیر عدم قطعیت‌های ممکن در شبکه روی زمان جابه‌جایی داده است. الگوریتم در هر مجموعه جست‌وجو می‌کند تا با توجه به دو–فاکتور معرفی‌شده در فرمول‌های (۹–۷)، یک ماشین مجازی مناسب را انتخاب کند. الگوریتم، ماشین‌های قابل استطاعت را به غیر قابل استطاعت ترجیح می‌دهد. همچنین ماشین‌هایی که داده ورودی روی آن‌ها وجود دارد نیز به ماشین‌هایی داده‌ای روی آن‌ها نیست ترجیح داده می‌شوند. اگر الگوریتم نتواند یک ماشین‌مجازی مناسب پیدا کند و اجرای با تأخیر امکان‌پذیر نباشد، آن وظیفه به صف انتظار اضافه می‌شود تا در مرحله بسته‌بندی چند منبع روی یک ماشین‌مجازی جدید اجرا شود.

$$Cfactor(v,t) = \frac{\beta_t - cost(v,t)}{\beta_t - cost_{min}(v,t)} \quad (۷)$$

$$Ufactor(v,t) = \sqrt{\left(1 - \frac{core_{idle}\ v - Dem_{ct}}{core_{total}\ v}\right)^2 + \left(1 - \frac{mem_{idle} v - Dem_{mt}}{mem_{total} v}\right)^2} \quad (۸)$$

$$Bifactor(v,t) = Cfactor(v,t) + Ufactor(v,t) \quad (۹)$$

که در آن β_t زیربودجه وظیفه t است. $cost(v, t)$ و $cost_{min}(v, t)$ به ترتیب هزینه اجرای تخمینی وظیفه t روی ماشین v و کم‌ترین هزینه اجرا در بین همه ماشین‌های مجازی کاندید است. فاکتور $Cfactor(v, t)$ در واقع میزان نزدیکی هزینه اجرا به زیربودجه را اندازه می‌گیرد. $core_{idle}v$ و $mem_{idle}v$ به ترتیب تعداد هسته‌های بیکار و میزان حافظه آزاد ماشین v هستند. Dem_{ct} و Dem_{mt} نیز نشان‌دهنده درخواست وظیفه t برای تعداد هسته‌ها و میزان حافظه مورد نیازش است. $Ufactor(v, t)$ درواقع میزان نزدیکی بهره‌وری ماشین به حالت بهره‌وری کامل را بعد از نگاشت وظیفه t به آن ماشین، اندازه می‌گیرد.

۵-۲-۳- بسته‌بندی چند منبع

در این مرحله الگوریتم وظیفه‌های صف انتظار را به خوشه‌های متفاوت به طوری تقسیم می‌کند که وظایف هم عمق از یک جریان‌کار در یک خوشه قرار بگیرند. در این روش احتمال اینکه وظایف همنوع در یک خوشه قرار بگیرند، بیشتر می‌شود. سپس خوشه‌ها بر اساس زیرفرجه به صورت صعودی اولویت بندی می‌شوند. سپس الگوریتم با توجه به درخواست وظیفه‌های هر خوشه، تعداد مورد نیاز از هر نوع ماشین مجازی را طوری تخمین می‌زند که در نهایت با بسته‌بندی کردن وظیفه‌ها، تعداد کمتری ماشین اجاره بشوند.

۵-۳- دیدبانی

دیدبانی در واقع یک راهبرد آزادسازی منابع است و کمک می‌کند تا هزینه ماشین‌های مجازی اجاره‌شده کاهش یابد. این الگوریتم در هر دوره دیدبانی اجرا می‌شود. در هر بار اجرا، الگوریتم ماشین‌های مجازی بیکاری که در آخرین دقیقه‌های دوره پرداختشان قرار دارند را از استخر منابع حذف و آزاد می‌کند. مقدار دقیق دوره اجرا الگوریتم دیدبانی و آستانه تعیین‌کننده دقیقه-های پایانی در واقع یک سبک‌وسنگین بین کارایی الگوریتم زمان‌بندی و هزینه است.

٦- ارزیابی و تحلیل کارایی

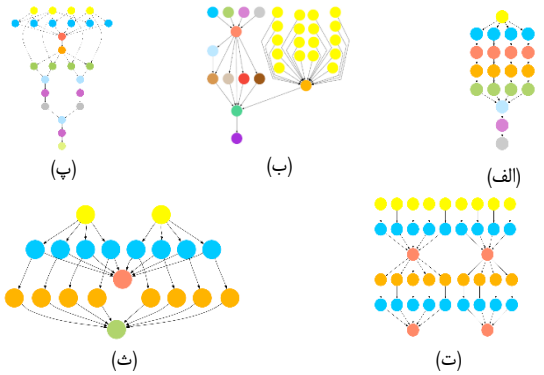
در این فصل در مورد محیط ارزیابی، مجموعه داده استفاده‌شده، خصوصیات ماشین‌های مجازی و معیارهای استفاده‌شده در ارزیابی و مقایسه کارایی الگوریتم زمان‌بندی شرح داده می‌شود. برای ارزیابی کارایی الگوریتم‌ها ابزار شبیه‌ساز CloudSim توسعه داده شده است تا از شبیه‌سازی یک محیط جریان‌کار به‌عنوان سرویس و امکان اجرای چند وظیفه روی یک ماشین مجازی و کانتینر پشتیبانی کند [17].

در ارزیابی‌ها انجام‌شده، فرض شده است که چهار مدل مختلف ماشین مجازی وجود دارد و همه‌ی این ماشین‌های مجازی در یک مرکز داده قرار دارند و توسط یک ارائه‌دهنده زیرساخت به‌عنوان سرویس ارائه می‌شوند. خصوصیات این ماشین‌های مجازی از نظر تعداد هسته، حافظه و قیمت آن‌ها به ازای هر دوره پرداخت از بین انواع ماشین‌های مجازی سرویس EC2 آمازون انتخاب شده‌اند [18]. این خصوصیات در جدول (۱) آورده شده است. در آزمایش‌ها عدم قطعیت در قالب تغییرات کارایی پردازنده و تغییرات پهنای باند در قالب توزیع نرمال، تأخیر در گرفتن و ساختن ماشین مجازی و کانتینر مطابق مرجع [13] در نظر گرفته شده است.

به‌منظور ارزیابی کارایی الگوریتم پیشنهادی از یک مجموعه داده شامل پنج نوع جریان‌های‌کار مصنوعی استفاده شده است. شکل (۱) ساختار کلی این پنج جریان کار را نشان می‌دهد. جزئیات بیشتر در مورد ساختار آن‌ها در مرجع [19] آورده شده است.

جدول (۱): خصوصیات ماشین‌های مجازی [18]

نوع ماشین	تعداد هسته	حافظه	هزینه/ساعت $
c4.large	۲	GiB ۳.۷۵	$ ۰.۱۰
c4.xlarge	۴	GiB ۷.۵	$ ۰.۱۹۹
c5.2xlarge	۸	GiB ۱۶	$ ۰.۳۴
c6g.4xlarge	۱۶	GiB ۳۲	$ ۰.۵۴۴



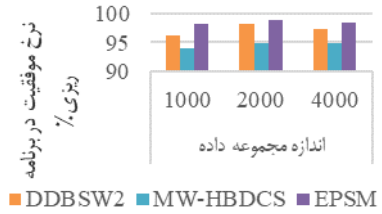
شکل (۱): ساختار کلی جریان‌های‌کار، (آ) اپیژنومیک، (ب) سیفت، (پ) مونتاژ، (ت) لیگو، (ث) سایبرشیک [19]

برای ساخت مجموعه داده از ابزار سازنده جریان‌کار مصنوعی پگاسوس استفاده شده است [20]. مجموعه داده از سه بارکاری مختلف با اندازه‌های ۱۰۰۰، ۲۰۰۰ و ۴۰۰۰ جریان‌کار تشکیل شده است. جریان‌های‌کار در هر بارکاری دارای یکی از اندازه‌های کوچک (۱۰۰ وظیفه)، متوسط (۲۰۰ وظیف)، بزرگ (۵۰۰ وظیفه) و خیلی بزرگ (۱۰۰۰ وظیفه) هستند. نوع و اندازه هر جریان‌کار برای تولید به صورت رندم انتخاب شده‌اند. زمان رسیدن هر درخواست در سیستم نیز بر اساس توزیع پواسون است.

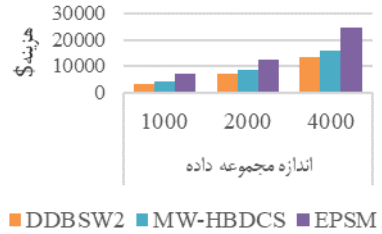
برای ارزیابی و مقایسه کارایی الگوریتم زمان‌بندی DDBSW2، دو الگوریتم MW-HBDCS [14] و EPSM [13] استفاده شده‌اند. برای ارزیابی کارایی الگوریتم پیشنهادی از معیار نرخ موفقیت در برنامه‌ریزی استفاده شده است. یک برنامه زمان‌بندی برای اجرای یک جریان‌کار زمانی موفق است که زمان و هزینه اجرای آن جریان‌کار از فرجه و بودجه تعیین‌شده توسط کاربر بیش‌تر نباشد. همچنین به‌منظور مقایسه کیفیت برنامه زمان‌بندی هزینه نهایی اجرا و تعداد ماشین‌های مجازی نیز با هم مقایسه شده‌اند.

همانطور که در شکل (۲) (آ) مشخص است نرخ موفقیت در برنامه‌ریزی در همه موارد برای این الگوریتم بالای ۹۶٪ است. تقریبا در همه موارد الگوریتم DDBSW2 از الگوریتم MW-HBDCS بهتر عمل کرده است. اگرچه در مقایسه با الگوریتم EPSM، در بعضی موارد نرخ موفقیت این الگوریتم مقدار مقدار ناچیزی کم‌تر است. این تفاوت می‌تواند به دلیل تأثیر تغییرات کارایی هسته پردازشی منابع باشد. با توجه به شکل (۲) (ب) همانطور که انتظار می‌رفت، الگوریتم پیشنهادی در همه موارد از دو الگوریتم دیگر بهتر عمل کرده‌است. به طور میانگین ۳۳.۲٪ و حداقل ۱۳.۲٪ منجر به هزینه کم-

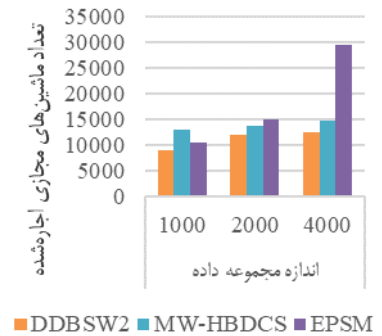
تری شده است. این نتایج تاثیر اشتراک منابع و اجرای همزمان چند وظیفه روی یک منبع را نشان می‌دهد. همچنین فرض ما که به کمک اشتراک منابع می‌توان هزینه‌های اجرا را کاهش داد، ثابت می‌کند. همچنین هزینه الگوریتم EPSM بسیار زیاد است. این می‌تواند به این دلیل باشد که نه‌تنها این الگوریتم بودجه را به طور مستقیم در نظر نمی‌گیرد، بلکه در حین زمان‌بندی، بسته‌بندی وظایف روی منابع را نیز در نظر نمی‌گیرد. شکل (۲) (پ) نشان می‌دهد که الگوریتم زمان‌بندی پیشنهادی در همه‌ی موارد، تعداد کم‌تری ماشین‌مجازی اجاره کرده است که به دلیل رفتار این الگوریتم و بسته‌بندی چند وظیفه روی یک ماشین مجازی است. زمانی که بار کلی افزایش می‌یابد، الگوریتم می‌تواند به جای اجاره‌کردن ماشین مجازی جدید، وظایف بیش‌تری روی ماشین‌های اجاره‌شده بزرگ‌تر نگاشت کند و بازه‌های زمانی خالی را پر کند. از طرفی الگورتیم EPSM برای رسیدن به نرخ موفقیت در زمان‌بندی بالاتر، تعداد بسیار زیادی ماشین مجازی اجاره کرده است.



(آ)



(ب)



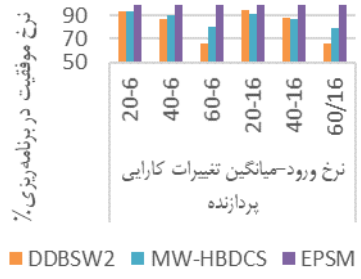
(پ)

شکل (۲): (آ) نرخ موفقیت در برنامه‌ریزی، (ب) هزینه نهایی اجرا، (پ) تعداد ماشین‌های مجازی اجاره‌شده در بارهای‌کار مختلف

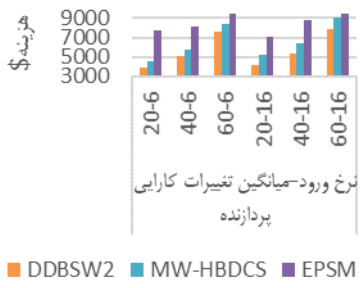
به منظور بررسی تاثیر نرخ ورود و عدم قطعیت، شکل‌های (۳) (آ) نرخ موفقیت در برنامه‌ریزی در دو نرخ ورود متفاوت و میانگین تغییرات کارایی پردازنده مختلف را نشان می‌دهد. به طور کلی با افزایش بار و میانگین تغییرات کارایی پردازنده، نرخ موفقیت در برنامه‌ریزی دو الگوریتم DDBSW2 و MW-HBDCS کاهش می‌یابد. به‌طور کلی الگوریتم DDBSW2 نسبت به تغییرات کارایی حساس‌تر است. دلیل این حساسیت این است که این الگوریتم کم‌ترین تعداد هسته‌های مورد نیاز را برای اجرای یک وظیفه در زیرفرجه تعیین‌شده در نظر می‌گیرد. در شرایطی که تغییرات کارایی پردازنده زیاد باشد، این حدس نادرست خواهد بود. یکی از مزیت‌های الگوریتم DDBSW2 که باعث می‌شود تا تأثیر تغییرات کارایی پردازنده روی کارایی الگوریتم کاهش یابد، رفتار پویا الگوریتم در به روز رسانی زیرفرجه و زیربودجه وظایف است. همچنین با توجه به شکل (۳) (ب) به دلیل بسته‌بندی وظایف روی منابع و اشتراک منابع، در همه موارد الگوریتم پیشنهادی کم‌ترین هزینه را دارد.

۷- نتیجه

در این مقاله الگوریتم زمان‌بندی پویا و چند محدودیته DDBSW2 برای زمان‌بندی جریان‌کار در محیط جریان‌کار به‌عنوان سرویس ارائه شد. این الگوریتم برای کاهش هزینه اجرا به کمک تکنولوژی کانتینر منابع محاسباتی



(آ)



(ب)

شکل (۳): (آ) نرخ موفقیت در برنامه‌ریزی، (ب) هزینه نهایی اجرا، در نرخ‌های ورود و میانگین تغییرات کارایی پردازنده مختلف

را بین چند وظیفه به اشتراک می‌گذارد. بـه طـور خلاصـه نـوآوری‌هـای ایـن الگوریتم معرفی شده در این مقاله مطابق زیر است:

۱. مسئله زمان‌بندی جریان‌های‌کار به کمک مسئله بسته‌بندی چند منبع و ارائه یک مدل جدید برای به اشتراک‌گذاری ماشین‌های مجازی بین چند وظیفه با استفاده از کانتینر حل شده است.
۲. یک مدل محاسبه هزینه جدید متناسب با مدل منابع استفاده‌شده و نحوه نگاشت وظیفه‌ها به منابع محاسباتی ارائه شده است.
۳. در هنگام نگاشت وظیفه‌ها و ایجاد برنامـه زمـان‌بنـدی، درخواسـت هـر وظیفه در قالب تعداد هسته‌ها و حافظه مورد نیاز آن در نظر گرفته شـده است.
۴. فرجه و بودجه به‌عنوان دو نیازمندی کیفیت سرویس تعیین‌شـده توسـط کاربر در هنگام زمان‌بندی و اجرای جریان‌کار فرض شده‌اند.

نتایج حاصل از ارزیابی نشان می‌دهد که در این رویکرد نرخ موفقیـت در برنامه‌ریزی در این الگوریتم در بارهای کاری مختلـف بـالای ۹۶٪ اسـت. همچنین با اجرای همزمان چند وظیفه روی یک ماشین مجازی هزینه نهـایی اجرا حداقل ۱۳.۲٪ و به طور میانگین ۳۳.۲٪ کاهش یافته است؛ بنابراین می‌توان گفت که استفاده اشتراکی از یک منبع می‌تواند بـه کـاهش هزینـه اجـرا منجر شود. برای توسعه و بهبود الگوریتم ارائه‌شده جنبه‌هـای متفـاوتی وجـود دارد. به عنوان مثال برای بهبود بهره‌وری منابع می‌توان مرحله بسته‌بندی چند منبع را بهبود داد. همچنین مواردی مانند تحمل‌پذیری خطا، قابلیت اطمینـان، محدودیت در تعداد منابع می‌توانند زمینه‌های جـالبی بـرای توسـعه الگـوریتم زمان‌بندی باشند. همچنین استفاده از روش‌های یادگیری ماشین برای زمـان‌بندی در یک محیط پویا به منظور پیش‌بینی وضعیت سیستم در آینده نزدیک نیز می‌تواند کارایی یک الگوریتم زمان‌بندی را افزایش دهد.

مراجع

[1] Coffman, E.G., *Computer and Job-shop Scheduling Theory*, Wiley Publishing, 1976.
[2] Deldari A., Naghibzadeh M., Abrishami S., *"CCA: a deadline-constrained workflow scheduling algorithm for multicore resources on the cloud"*, The Journal of Supercomputing, Vol. 73, No.2, pp. 756-781, 2017.
[3] Sahni J., Vidyarthi D. P., *"A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment"*, IEEE Transactions on Cloud Computing, Vol. 6, No.2, pp. 2-18, 2015.
[4] Rodriguez M. A., Buyya R., *"Budget-Driven Scheduling of Scientific Workflows in IaaS Clouds with Fine-Grained Billing Periods"*, ACM Transactions on Autonomous and Adaptive Systems (TAAS), Vol. 12, No. 2, pp. 1-22, 2017.
[5] Ghasemzadeh M., Arabnejad H., Barbosa J. G., *"Deadline-Budget constrained Scheduling Algorithm for Scientific Workflows in a Cloud Environment"*, in Proceedings of the 20th International Conference on Principles of Distributed Systems (OPODIS 2016), April-2017, Dagstuhl, Germany, pp. 19:1-19.16.
[6] Arabnejad H., Barbosa J. G., Prodan R., *"Low-time complexity budget–deadline constrained workflow scheduling on heterogeneous resources"*, Future Generation Computer Systems, Vol. 55, pp. 29-40, 2016.
[7] Zheng W., Sakellariou R., *"Budget-Deadline Constrained Workflow Planning for Admission Control"*, Journal of grid computing, Vol. 11, No. 4, pp. 633-651, 2013.
[8] Zheng W., W. Yan, Bugingo E., Zhang D., *"Online Scheduling to Maximize Resource Utilization of Deadline-Constrained Workflows on the Cloud"*, in Proceedings of the 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD), May- 2018, Nanjing, China, pp. 98-103.
[9] Li Z., Ge J., Hu H., Song W., Hu H., Luo B., *"Cost and Energy Aware Scheduling Algorithm for Scientific Workflows with Deadline Constraint in Clouds"*, IEEE Transactions on Services Computing, Vol. 11, No. 4, pp. 713-726, 2018.
[10] Chen H., Zhu X., Liu G., Pedrycz Wi., *"Uncertainty-Aware Online Scheduling for Real-Time Workflows in Cloud Service Environment"*, IEEE Transactions on Services Computing (Early Access), pp. 1-13, 2018.
[11] Liu J., Ren J., Dai W., Zhang D., Zhou P., Zhang Y., Min G., Najjari N., *"Online Multi-Workflow Scheduling under Uncertain Task Execution Time in IaaS Clouds"*, IEEE Transactions on Cloud Computing (Early Access), pp. 1-15, 2019.
[12] Malawski M., Juve G., Deelman E., Nabrzyski J., *"Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds"*, Journal of Internet Services and Applications, Vol. 48, pp. 1-18, 2015.
[13] Rodriguez M. A., Buyya R., *"Scheduling Dynamic Workloads in Multi-tenant Scientific Workflow as a Service Platforms"*, Future Generation Computer Systems, Vol. 79, pp. 739-750, 2018.
[14] Zhou N., Li F., Xu K., Qi D., *"Concurrent workflow budget- and deadline-constrained scheduling in heterogeneous distributed environments"*, Soft Computing, Vol. 22, No.23, pp. 7705-7718, 2018.
[15] J. Liu, J. Ren, W. Dai, D. Zhang, P. Zhou, Y. Zhang, G. Min, N. Najjari, *"Online Multi-Workflow Scheduling under Uncertain Task Execution Time in IaaS Clouds,"* IEEE Transactions on Cloud Computing (Early Access), pp.1-15, 2019.
[16] Docker. *Empowering App Development for Developers*, Nov 2020, https://www.docker.com.
[17] Calheiros R. N., Ranjan R., Beloglazov A., De Rose C. A., Buyya R., *"Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms"*, Software: Practice and experience, Vol. 41, No.1, pp. 23-50. 2011.
[18] Amazon Web Services. *Amazon EC2 Instance Types*, Nov 2019, https://aws.amazon.com/ec2/instance-types/.

[19] Juve G., Chervenak A., Deelman E., Bharathi S., Mehta G., Vahi K., *"Characterizing and profiling scientific workflows"*, Future Generation Computer Systems, Vol. 29, No.3, pp. 682-692, 2013.

[20] da Silva R. F., Chen W., Juve G., Vahi K., Deelman E. *"Community Resources for Enabling Research in Distributed Scientific Workflows"*, in Proceedings of the 2014 IEEE 10th International Conference on e-Science, Oct 20-24, Sao Paulo, Brazil, pp. 177-184.