



دانشگاه کاشان
University of Kashan

مجله محاسبات نرم

SOFT COMPUTING JOURNAL

تارنمای مجله: scj.kashanu.ac.ir



زمانبندی مبتنی بر هزینه جریان‌های کاری با استفاده از ساختار جبری[✦]

محمد جواد نجفی آرائی^۱، استادیار، سعید دوست علی^{۲*}، دانشجوی دکتری

^۱ گروه علوم کامپیوتر، دانشکده علوم، مرکز آموزش عالی محلات، محلات، ایران.

^۲ گروه مهندسی کامپیوتر، دانشکده برق و کامپیوتر، دانشگاه کاشان، کاشان، ایران.

اطلاعات مقاله

چکیده

تاریخچه مقاله:

دریافت ۱۲ شهریور ماه ۱۳۹۹

پذیرش ۲۳ شهریور ماه ۱۴۰۰

کلمات کلیدی:

زمانبندی جریان کاری

رایانش ابری

مسیر بحرانی

مجموعه مرتب جزئی

مشبکه

جریان‌های کاری یک مدل عمومی برای توصیف دامنه وسیعی از برنامه‌های کاربردی در سیستم‌های توزیع شده هستند. با توجه به قدرت محاسباتی رایانش ابری، از آن به طور گسترده برای حل جریان‌های کاری بزرگ استفاده می‌شود. زمانبندی جریان کاری در ابر در واقع یافتن منبع مناسب برای هر کار در جریان کاری به منظور ارضای برخی معیارهای کارایی مانند زمان اجرا و هزینه است. از آنجایی که زمانبندی یک مسئله زمان چندجمله‌ای غیرقطعی سخت (NP-complete) است، بسیاری از روش‌های ابتکاری برای سیستم‌های توزیع شده همگن و ناهمگن ارائه شده‌اند. مسیر بحرانی طولانی‌ترین مسیر یک جریان کاری است و زمان اجرای کلی جریان کاری به آن وابسته است. در واقع تاخیر در کارهای مسیر بحرانی می‌تواند زمان خاتمه جریان کاری را با تاخیر مواجه کرده و زمان انقضای جریان کاری را نقض کند. بر همین اساس در این مقاله، ما یک الگوریتم ابتکاری موازی برای زمانبندی جریان کاری مبتنی بر کیفیت سرویس ارائه می‌کنیم. تابع هدف این الگوریتم یک زمانبندی ایجاد می‌کند که هزینه اجرای یک جریان کاری را کمینه کرده، در حالی که زمان انقضای جریان کاری را نیز ارضا می‌کند. با اختصاص یک شبه مشبکه به هر زیرجریان کاری، زمان آغاز و پایان هر وظیفه و همچنین منبع مناسب برای آن مشخص می‌شود. نتایج حاصل از شبیه‌سازی بر روی جریان‌های کاری واقعی Montage و LIGO نشان می‌دهد که روش پیشنهادی در مقایسه با الگوریتم IC-PCP به میزان ۵/۵ درصد و نسبت به IC-PCPD2 به میزان ۱۱ درصد هزینه را کاهش داده است.

© ۱۴۰۰ - مجله محاسبات نرم، کلیه حقوق محفوظ است.

۱. مقدمه

و بیوانفورماتیک شامل می‌شوند [۱]. به صورت کلی، یک جریان کاری توسط یک گراف جهتدار غیرمدور (DAG)^۱ نشان داده می‌شود که در آن هر کار محاسباتی توسط یک گره و وابستگی‌های داده و کنترل توسط یال‌های جهتدار بین گره‌های مربوطه مدل می‌شوند.

جریان‌های کاری مهمترین و رایج‌ترین مدل برای توصیف دامنه وسیعی از برنامه‌های کاربردی در ابر هستند که طیف گسترده‌ای از کاربردهای علمی را در زمینه‌های مختلف مانند فیزیک، نجوم

✦ نوع مقاله: پژوهشی

* نویسنده مسئول

پست(های) الکترونیک: mjnajarani@gmail.com (نجفی آرائی)

doostali.s@gmail.com (دوست علی)

¹ Directed Acyclic Graph

سرویس‌های مناسب مواجهه است، که این امر در واقع به عنوان یک مسئله بهینه‌سازی چند هدفه می‌تواند شناخته شود [۷].

روش‌های بسیاری برای زمانبندی چند هدفه وجود دارد:

۱. یافتن جواب‌های بهینه با استفاده از جبهه پرتو^۷ و انتخاب بهترین زمانبندی بر اساس نیازمندی‌های کاربر،
۲. اختصاص وزن به هر معیار زمانبندی و بهینه کردن مجموع وزنی این معیارها،
۳. زمانبندی دو معیاری^۸ که در آن کاربر یک محدودیت برای یک معیار (مانند زمان انقضا یا بودجه) مشخص می‌کند و الگوریتم تلاش می‌کند سایر معیارها را با توجه به قید بیان شده بهینه کند.

روش آخر، یک راه مناسب برای کاربران به منظور بیان نیازهای آنها و یک روش مفید برای محققان به منظور ساده‌سازی مسئله و پیشنهاد راه‌حل‌های سریع با کارایی بالا است. بسیاری از محققان روش بیان شده را برای ارائه الگوریتم زمانبندی چندهدفه دنبال کرده‌اند اما با توجه به پیچیدگی این روش بسیاری از آنها از روش‌های فرااکتشافی زمان بر مانند الگوریتم ژنتیک استفاده کرده‌اند. بنابراین تنها تعداد کمی روش ابتکاری و سریع برای غلبه بر این مشکل وجود دارد [۷،۸،۹].

در این مقاله ما یک الگوریتم ابتکاری موازی برای زمانبندی جریان کاری مبتنی بر کیفیت سرویس با نام مسیرهای بحرانی مبتنی بر شبکه (CPL) ارائه می‌کنیم. تابع هدف این الگوریتم یک زمان بندی ایجاد می‌کند که هزینه اجرای یک جریان کاری را کمینه کرده، در حالی که زمان انقضای تعریف شده برای زمان اجرای نهایی را ارضا می‌کند. روش پیشنهادی با ارائه مسیر بحرانی به صورت خوشتعریف و سپس اختصاص شبه شبکه به هر زیر جریان کاری تلاش می‌کند تا به صورت موازی زمان شروع و خاتمه هر وظیفه در مسیر بحرانی زیرجریان‌های کاری مجزا را مشخص کند. این زمان‌ها به گونه‌ای اختصاص می‌یابند که هزینه اجرای کلی کمینه شود. به منظور ارزیابی کارایی روش پیشنهادی، آن را با دو روش IC-PCP و IC-PCPD2 [۸] مورد

زیرساخت‌هایی برای اجرای جریان‌های کاری بزرگ که به منابع و خدمات زیادی نیاز دارند، به خود جلب کرده است [۳،۲]. زیرساخت به عنوان خدمت (IaaS)^۱ به عنوان متداول‌ترین مدل سرویس ابری، منابع را به صورت مجازی برای استفاده کارآمد، فراهم می‌کند. در این مدل، کاربران می‌توانند این سرویس‌ها را بر اساس یک توافقنامه سطح سرویس (SLA)^۲ که پارامترهای کیفیت خدمات (QoS)^۳ مورد نیاز را تعریف می‌کند، استفاده نمایند [۴].

زمانبندی جریان کاری در واقع نداشت هر کار به ماشین مجازی مناسب برای ارضای برخی معیارهای کارایی است. از آنجایی که زمانبندی یک مسئله NP-complete است [۵]، بسیاری از روش‌های ابتکاری^۴ برای سیستم‌های توزیع‌شده همگن و ناهمگن ارائه شده‌اند. تلاش این روش‌ها در جهت کمینه کردن زمان اجرا^۵ جریان کاری بوده و بسیاری از سیستم‌های مدیریت جریان کاری فعلی از چنین روش‌های زمانبندی استفاده می‌کنند. در برنامه‌های حساس به زمان که باید در یک زمان مشخص یا به عبارتی در یک ضرب‌الاجل اجرا شوند، از دست دادن زمان انقضاً ممکن است نتایج فاجعه باری در زندگی یا محیط زیست ایجاد کند. یک جریان کاری معمولاً چندین مسیر مختلف از گره ابتدایی تا گره انتهایی دارد. طولانی‌ترین مسیر، مسیر بحرانی و وظایف موجود در این مسیر، وظایف بحرانی نامیده می‌شوند. زمان اتمام کل جریان کار به مسیر بحرانی بستگی دارد. تاخیر در یک وظیفه بحرانی باعث تأخیر در زمان اجرای کل جریان کاری می‌شود [۶].

با این همه، در رایانش ابری پتانسیل‌های بیشتری برای کیفیت سرویس علاوه بر زمان اجرا مانند اطمینان‌پذیری، امنیت و دسترس‌پذیری وجود دارد. در ضمن، کیفیت سرویس سریع‌تر به معنای قیمت بالاتر برای خدمات است. بنابراین، زمانبندی با یک مصالحه بین کیفیت سرویس و هزینه برای انتخاب

¹ Infrastructure as a Service

² Service Level Agreement

³ Quality of Services

⁴ Heuristic approaches

⁵ Makespan

⁶ Deadline

⁷ Pareto

⁸ Bicriteria

پویا، راه‌حلی با هزینه بهینه و محدود به زمان انقضا را بدست آوردند، در حالی که برای وظایف غیربحرانی، آنها یک روش تکراری را برای به حداقل رساندن هزینه ارائه دادند.

در [۱۱]، نویسندگان یک الگوریتم زمانبندی برای جریان‌های کاری مبتنی بر مسیر بحرانی را پیشنهاد کردند. آنها با در نظر گرفتن زمان انقضای تعریف شده توسط کاربر برای یک جریان کاری، وظایف موجود در مسیر بحرانی را یافته و هزینه اجرای کلی جریان کاری را کاهش داده‌اند. بر اساس الگوریتم مسیره‌های بحرانی جزئی ارائه شده در [۷]، ابریشمی و همکاران دو الگوریتم جدید با نام‌های IC-PCP و IC-PCPD2 برای زمانبندی جریان‌های کاری در ابر ارائه کرده‌اند [۸]. در الگوریتم IC-PCP، نویسندگان به جای تعیین زیر زمان‌های انقضا برای هر وظیفه در یک جریان کاری، هر یک از آنها را به صورت مجزا زمانبندی می‌کنند، در حالی که در IC-PCPD2، آنها مراحل توزیع مهلت و برنامه‌ریزی موجود در الگوریتم مسیره‌های بحرانی جزئی را برای سازگاری با ابر اصلاح می‌کنند. اگرچه این دو روش دارای پیچیدگی زمان کمی هستند، اما برای زمان انقضای سخت، آنها قادر به ارضای مهلت نخواهند بود. علاوه بر این تعریف مسیر بحرانی در آنها خوشتعریف نیست، به عبارت دقیق‌تر، به ازای انتخاب هر ماشین مجازی دلخواه به عنوان مرجع یافتن مسیر بحرانی جریان کاری، مسیر بحرانی متفاوتی را در شبکه نمایش می‌دهد، در حالی که بنابر این الگوریتم مسیر بحرانی بایستی مستقل از انتخاب ماشین مجازی باشد.

در [۱۲]، نویسندگان برای بهبود روش IC-PCP، از یک الگوریتم تکرار شونده استفاده کرده‌اند. این الگوریتم برای افزایش احتمال ارضای زمان انقضا، وظایف یک جریان کاری را تکرار می‌کند.

کای و همکاران یک روش ابتکار مبتنی بر مسیر بحرانی برای ایجاد مسیره‌های بحرانی کامل به جای مسیره‌های جزئی تولید شده توسط الگوریتم مسیره‌های بحرانی جزئی ارائه کرده‌اند [۱۳]. با توجه به اختصاص مسیر به برنامه‌نویسی پویا برای تعیین جبهه پرتو، استفاده از این روش هنگامی که جریان کاری

مقایسه قرار داده‌ایم. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی می‌تواند به صورت موثرتری منجر به کاهش هزینه‌های اجرای جریان‌های کاری شود.

این مقاله در هفت بخش سازمان‌دهی شده است. بخش دوم به مرور پیشینه زمانبندی دارای محدودیت در ابر مبتنی بر مسیر بحرانی می‌پردازد. سپس در بخش سوم مدل زمانبندی سیستم ارائه خواهد شد. در بخش چهارم تناظر دوسویی بین یک جریان کاری با یک ساختار جبری به نام شبکه بیان می‌شود. سپس در بخش پنجم، راه‌کار پیشنهادی مبتنی بر شبه شبکه ذکر شده و این روش از طریق شبیه‌سازی در بخش ششم مورد ارزیابی قرار می‌گیرد. در نهایت در بخش هفتم نتیجه‌گیری و کارهای آتی بیان می‌شود.

۲. کارهای گذشته

کارهای زیادی در مورد زمانبندی جریان‌های کاری وجود دارد که سعی می‌کنند چندین هدف را به طور همزمان بهینه کنند. با توجه به اهمیت مسیر بحرانی، در این بخش، برخی از کارهایی که مرتبط‌تر و قابل توجه‌تر در این زمینه هستند، مورد بررسی قرار خواهد گرفت.

یوان و همکاران یک رویکرد اکتشافی موثر و کارآمد مبتنی بر درخت زودرس^۱ برای زمانبندی جریان‌های کاری را پیشنهاد کردند [۱۰]. در روش آنها، درخت زودرس، درخت پوشای مرتبط با یک زمانبندی شدنی سریع است. بنابراین هر گره در این درخت (که معادل یک وظیفه در جریان کاری است)، شامل زودترین زمان شروع مربوط به آن گره بر اساس سرویس‌های در دسترس است. در ضمن برای انتخاب یال متصل به گره‌ای مانند i در این درخت، بایستی اختلاف زمان‌های شروع گره‌های دو سر این یال برابر با کمینه زمان اجرای وظیفه t_i باشد. به هر حال، روش آنها به عنوان یک روش اکتشافی محدود شده به زمان انقضا، وظایف را به دو گروه تقسیم می‌کند: بحرانی و غیربحرانی. برای وظایف بحرانی، آنها با استفاده از برنامه‌نویسی

¹ Early Tree

متغیر تصادفی نرمال مشخص می‌کند. دقت کنید که، آنها الگوریتم سنتی مسیر بحرانی را برای رتبه‌بندی دقیق‌تر وظایف در یک جریان کاری گسترش داده‌اند. نتایج شبیه‌سازی انجام شده توسط آنها نشان می‌دهد که به لطف اطلاعات دقیق رتبه‌بندی، تصمیمات بهتری در مورد اختصاص وظایف به منابع محاسباتی گرفته می‌شود.

در [۱۹]، نویسندگان رویکردی برای زمانبندی جریان‌های کاری در محیط ابر پویا ارائه کرده‌اند. آنها برای غلبه بر مشکل زمانبندی جریان کاری، از یک مدل پیش‌بینی مبتنی بر زمان و یک استراتژی انتخاب ماشین مجازی بر اساس تخمین مدت زمان مسیر بحرانی استفاده کرده‌اند. روش پیشنهادی آنها قادر به پشتیبانی از تغییرات زمان واقعی عملکرد زیرساخت‌های ابر و همچنین پشتیبانی از زمانبندی جریان‌های کاری پویا است.

۳. مدل زمانبندی سیستم

مدل زمانبندی سیستم ارائه شده شامل یک مدل جریان کاری، یک مدل منابع ابر و یک معیار کارایی برای زمانبندی است. جریان کاری مقید به زمان انقضا توسط گراف جهت‌دار بدون دور $G(T, E)$ مدل می‌شود که در آن مجموعه n وظیفه محاسباتی $\{t_1, t_2, \dots, t_n\}$ و E مجموعه وابستگی‌های داده و کنترل بین وظایف است که با یال‌های جهت‌دار نشان داده می‌شوند. هر یال $e_{i,j} = (t_i, t_j)$ معرف یک محدودیت تقدمی است به این معنی که وظیفه t_i باید پیش از شروع وظیفه t_j تمام شود. در واقع وظیفه t_i والد وظیفه t_j است که با $t_i \in \text{Parent}(t_j)$ نشان داده می‌شود و به تبع وظیفه t_j فرزند وظیفه t_i است ($t_j \in \text{Child}(t_i)$). در گراف، وظیفه‌ای که دارای پدر نیست به عنوان وظیفه ورودی^۵ و وظیفه بدون فرزند به عنوان وظیفه خروجی^۶ شناخته می‌شوند. دقت داشته باشید که، در صورت عدم وجود این دو وظیفه در گراف، به سادگی می‌توان دو وظیفه ساختگی t_{entry} و t_{exit} را به ابتدا و انتهای جریان کاری اضافه کرد. این وظیفه‌ها دارای زمان

دارای زمان انقضای سخت است، ناکارآمد است.

پولا و همکاران [۱۴] سه الگوریتم برای زمانبندی جریان‌های کارهای بر روی منابع ناهمگن در ابر پیشنهاد داده‌اند. این الگوریتم‌ها بر اساس مسیرهای بحرانی جزئی ضمن به حداقل رساندن هزینه و زمان اجرای جریان کاری، یک زمانبندی قوی^۱ و مقاوم در برابر خطا^۲ را ارائه می‌دهند. بر اساس سیاست هر یک از الگوریتم‌ها، اهداف ذکر شده دارای اولویت‌های مختلفی در آنها می‌باشند.

در [۹]، نویسندگان یک الگوریتم ابتکاری به نام PCP-B2 را برای بهینه‌سازی زمان اجرا با بودجه محدود ارائه دادند. آنها بودجه را در میان زیرمسیرهای بحرانی با توجه به ساختار متوالی یا موازی آنها موازنه می‌کنند.

در زمینه بارهای کاری، عرب‌نژاد و همکاران دو الگوریتم PDC و DCCP را برای زمانبندی بارهای کاری علمی دارای زمان انقضا ارائه کرده‌اند [۱۵، ۱۶].

غفوری و همکاران [۱۷] الگوریتم برنامه‌ریزی با نام CB-DT برای کاهش زمان اجرای جریان‌های کاری مقید به بودجه محدود را پیشنهاد دادند. به منظور بدست آوردن زمان اجرای کوچک‌تر، الگوریتم پیشنهادی آنها با استفاده از روش بازگشت به گذشته^۳، سعی می‌کند تا حد ممکن منابع سریع‌تر و گران‌تر را برای وظایف بحرانی انتخاب کند.

در [۱۸]، نویسندگان جریان‌های کاری را به صورت تصادفی در نظر می‌گیرند و پارامترهای وظایف را به عنوان متغیرهای تصادفی نرمال مدل می‌کنند. اما به منظور ساده‌سازی فرآیند محاسبه، آنها یک جریان کاری تصادفی^۴ را با چندین جریان کار با فاصله تقریبی تخمین می‌زنند. یک جریان کار تصادفی، سه‌تایی (T, D, N) است که در آن مجموعه وظایف جریان کاری، $D \subseteq T \times T$ مجموعه وابستگی بین وظایف و $N: T \rightarrow R$ تابعی است که اندازه هر وظیفه در جریان کاری را به وسیله یک

¹ Robustness scheduling

² Fault tolerant

³ Back tracking

⁴ Stochastic workflow

⁵ Entry task

⁶ Exit task

کاربران اختصاص می‌دهد. بنابراین هزینه اجرای وظیفه‌ای مانند t_i می‌تواند به صورت زیر محاسبه شود:

$$C_{i,j} = C_j \left(\frac{T_{i,j}}{u} \right) \quad (2)$$

که در این رابطه C_j هزینه استفاده از بازه‌های زمانی ماشین مجازی vm_j و u طول بازه زمانی بر اساس استراتژی قیمت‌گذاری است.

آخرین عنصر در مدل ما معیار کارایی است. در حالت کلی کاربران ترجیح می‌دهند تا زمان پایان کار خود را کمینه کنند. با این حال، هزینه اجرا نیز یک عامل مهم برای آنها است، بنابراین کاربران ترجیح می‌دهند از سرویس‌های ارزان‌تر با QoS کمتر که نیازها و انتظارات آنها را برآورده می‌کند، استفاده کنند. در جریان‌های کاری مقید به زمان انقضا، مشتری یک زمان انقضا تعریف می‌کند که انتظار دارد جریان کاری تا آن زمان به پایان برسد. بنابراین در این مقاله، معیار کارایی کمینه کردن هزینه اجرا جریان کاری با در نظر گرفتن تمام شدن جریان کاری پیش از زمان انقضای تعریف شده توسط کاربر در نظر گرفته می‌شود.

۴. تناظر دوسویی بین جریان کاری و شبکه

در این بخش ابتدا به معرفی ساختار جبری یک شبکه می‌پردازیم و سپس تناظر دوسویی مابین یک جریان کاری و یک شبکه را نشان می‌دهیم. اگرچه در این بخش سعی شده است تمامی مطالب کلیدی مربوط به یک ساختار جبری بالاخص شبکه به صورت جامع و مانع توضیح داده شود، با این وجود لازم به ذکر است نویسندگان از مفاهیم و نمادهای کتاب [۲۰] به عنوان مرجع استفاده کرده‌اند.

یک ساختار جبری، عبارت است از یک مجموعه غیرتهی از اشیا، که یک یا چند عمل روی آن تعریف شده باشد. در نظر داشته باشید که جبر مجرد به مطالعه ساختارهای جبری می‌پردازد. به عنوان مثال مجموعه اعداد حقیقی به همراه عمل جمع یک ساختار جبری است. گاهی ممکن است ساختمان جبری همراه بیش از یک عمل دوتایی بیان شود به عنوان مثال

اجرای صفر بوده و با یال‌هایی با وزن صفر به ورودی واقعی و خروجی واقعی متصل می‌شوند.

بستر ابر، منابع محاسباتی را در قالب ماشین مجازی برای اجرای جریان‌های کاری در IaaS فراهم می‌کند. در ابر دو فرض اساسی در مورد منابع وجود دارد:

۱. منابع در ابر کشسان^۱ هستند، بدان معنی که استخراج منابع در ابر تقریباً نامحدود است و کاربران می‌توانند منابع را در هر زمان و به هر اندازه که بخواهند در اختیار بگیرند.
۲. منابع ناهمگن هستند، یعنی انواع متفاوتی از منابع در ابر وجود دارند. فرض کنید مجموعه $H = \{h_1, h_2, \dots, h_m\}$ نشان‌دهنده m نوع ماشین مجازی در ابر باشد. هر نوع می‌تواند شامل ویژگی‌های QoS متنوعی مانند قدرت پردازش، میزان حافظه، پهنای باند و هزینه‌های استفاده از آنها باشد. در این مقاله دو مورد مهم یعنی زمان اجرا و هزینه اجرا در نظر گرفته شده است. دقت کنید که هزینه یک سرویس معمولاً به زمان اجرا آن وابسته است، یعنی زمان اجرای کوتاه‌تر گران‌تر است.

با این مفروضات، زمان محاسبه وظیفه t_i می‌تواند به شکل زیر محاسبه شود:

$$T_{i,j} = \frac{L(t_i)}{MIPS_j} \quad (1)$$

که در آن $L(t_i)$ نشان دهنده طول وظیفه t_i بر حسب میلیون دستور $MIPS_j$ نشان دهنده قدرت پردازش vm_j است. دقت داشته باشید که طول یک وظیفه در جریان کاری، در واقع تعداد دستورالعمل‌های آن وظیفه است که بایستی توسط یک ماشین مجازی اجرا شود و قدرت پردازش یک ماشین مجازی برابر با تعداد دستورالعمل‌های است که می‌تواند در واحد زمان پردازش کند. بنابراین با تقسیم تعداد دستورالعمل‌های وظیفه t_i بر قدرت پردازش ماشین vm_j زمان محاسبه این وظیفه به دست می‌آید. در ابر استراتژی‌های قیمت‌گذاری متفاوتی وجود دارد. به عنوان مثال آمازون EC2 بازه‌های یک ساعتی و ماکروسافت آژور بازه‌های یک دقیقه‌ای را برای اجرای وظایف

¹ Elastic

بالا نداشته باشد. یک کران بالا u از S را هنگامی کوچکترین کران بالا یا سوپریموم می‌نامیم که به ازای هر کران بالا x داشته باشیم $x \leq u$. یک مجموعه لزوماً کوچکترین کران بالا ندارد ولی نمی‌تواند بیش از یکی داشته باشد. از طرفی، $l \in L$ یک کران پایین برای S است اگر به ازای هر $s \in S$ داشته باشیم $l \leq s$. یک کران پایین مانند l از S را هنگامی بزرگترین کران پایین یا اینفیموم می‌نامیم هرگاه به ازای هر کران پایین x داشته باشیم $x \leq l$. یک مجموعه ممکن است چند کران پایین داشته باشد، یا اصلاً کران پایین نداشته باشد ولی می‌تواند حداکثر یک بزرگترین کران پایین داشته باشد.

مشبکه (L, \leq) مجموعه‌ای با ترتیب جزئی است که در آن هر دو عنصر دارای سوپریموم¹ و اینفیموم² منحصر بفردی هستند. اعداد طبیعی یک مثال از مشبکه‌ها هستند که می‌توان با استفاده از رابطه مقسوم‌علیهی، یک رابطه ترتیب جزئی بر روی آنها تعریف کرد. سوپریموم منحصر به فرد این رابطه کوچکترین مضرب مشترک یا ک.م.م. و اینفیموم منحصر به فرد آن بزرگترین مقسوم علیه مشترک یا ب.م.م. است. برای هر دو عنصر دلخواه $a, b \in L$ سوپریموم و اینفیموم a و b را به ترتیب با $a \vee b$ و $a \wedge b$ نشان می‌دهیم که به آنها به ترتیب ترکیب فصلی و عطفی مجموعه $\{a, b\}$ گویند. یک مشبکه را کراندار گوئیم، هرگاه دارای عناصر بیشینه و کمینه باشد. هر مشبکه را می‌توان در یک مشبکه کراندار با اضافه کردن عناصر جدیدی به عنوان بیشینه و کمینه نشان داد و هر مشبکه متناهی ناتهی کراندار است. برای این منظور یک مشبکه متناهی را در نظر بگیرید. برای این مشبکه متناهی، می‌توان ترکیب عطفی و فصلی تمام عناصر را به ترتیب با $a_1 \vee \dots \vee a_n$ و $a_1 \wedge \dots \wedge a_n$ نشان داد، که در این دو رابطه $L = \{a_1, a_2, \dots, a_n\}$ است. دقت کنید که، ترکیب عطفی و فصلی تمام عناصر به ترتیب بیان کننده عناصر بیشینه و کمینه مشبکه متناهی هستند. توجه داشته باشید که یک مجموعه مرتب جزئی مشبکه کراندار است اگر و تنها اگر هر

می‌توان به مجموعه اعداد حقیقی به همراه دو عمل جمع و ضرب اشاره کرد. این ساختارها معمولاً دارای اصول موضوع تعریف شده‌ای هستند. از جمله مهم‌ترین ساختارهای جبری مجرد می‌توان تکواره‌ها، گروه‌ها، حلقه‌ها و مشبکه‌ها را نام برد. در ادامه به معرفی ساختار جبری یک مشبکه و رابطه آن با یک جریان کاری می‌پردازیم.

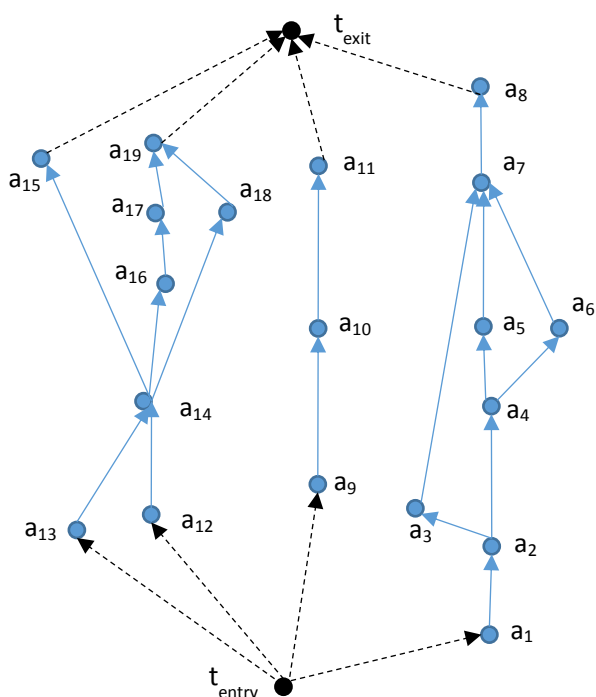
رابطه R در مجموعه Q یعنی $R \subseteq Q \times Q$ را یک رابطه ترتیب جزئی (که گاه به اختصار رابطه ترتیبی) می‌نامیم هرگاه انعکاسی، پادتقارنی و تعدی باشد. ریاضیدانان با الگو قرار دادن رابطه ترتیبی معمولی اعداد، یعنی \leq ، این نماد را برای هر رابطه ترتیبی دلخواه R در مجموعه دلخواه Q به کار می‌برند. در این حالت گوئیم (Q, \leq) یک مجموعه مرتب جزئی است. به سادگی می‌توان نشان داد گره‌های یک جریان کاری تحت رابطه پردازش گره، یک مجموعه مرتب جزئی هستند. به عبارت دیگر، اگر u و v دو گره دلخواه از شبکه باشند عبارت $u \leq v$ بدین مفهوم است که گره u محدودیت تقدیمی برای پردازش گره v است. می‌گوئیم y تالی x است یا x مقدم بر y است و می‌نویسیم $y < x$ هرگاه هیچ عضوی چون z به طور سره بین x و y وجود نداشته باشد. به عبارت دیگر نتوان عضوی مانند $z \in Q$ یافت به طوری که $x < z < y$ برقرار باشد. عنصر $a \in Q$ را بیشین می‌گوئیم هرگاه از هیچ عضو دیگر Q کوچکتر نباشد و a را بیشینه گوئیم هرگاه از همه اعضای Q بزرگتر باشد. مشابه عنصر $b \in Q$ را کمین می‌نامیم هرگاه هیچ عضو دیگر Q از آن کوچکتر نباشد و b را کمینه گوئیم هرگاه از همه اعضای Q کوچکتر باشد. یک ترتیب کامل یا یک ترتیب خطی، یک ترتیب جزئی روی یک مجموعه است که همه اعضای آن با هم رابطه داشته باشند. یک مجموعه با ترتیب خطی را مجموعه مرتب کلی یا زنجیر می‌گویند.

اگر (L, \leq) یک مجموعه مرتب جزئی باشد و S یک زیر مجموعه یا مساوی دلخواه L باشد، آنگاه $u \in L$ یک کران بالا برای S است اگر به ازای هر $s \in S$ داشته باشیم $u \leq s$. یک مجموعه ممکن است چند کران بالا داشته باشد یا اصلاً کران

¹ Supremum

² Infimum

شده است. هر مسیر می‌تواند با مسیرهای دیگر دارای گره مشترک باشد.



شکل (۱). جریان کاری ساده با ۱۹ گره

به عنوان مثال شکل (۱) را در نظر بگیرید. با استفاده از گره‌های مجازی (یعنی گره‌های t_{entry} و t_{exit}) یک شبکه ایجاد می‌شود. دقت کنید که زمان پردازش عناصر پیشینه و کمینه در این شبکه را صفر در نظر می‌گیریم. جریان کاری ارائه شده در این مثال شامل سه زیرجریان کاری مجزا (یا شبه شبکه) است که در این زیرجریان‌های کاری عناصر ابتدایی جریان یعنی گره‌های a_1, a_9, a_{12} و a_{13} به عنوان عناصر کمینه شبه شبکه‌ها شناخته می‌شوند، در حالی که گره‌های a_8, a_{11}, a_{15} و a_{19} عناصر پیشین هستند. اما در شبکه مجازی شناخته شده عناصر t_{entry} و t_{exit} به ترتیب کمینه و پیشینه می‌باشند. بدون کاسته شدن از کلیت مسئله، ما از این واژه‌ها به صورت معادل در ادامه این مقاله استفاده خواهیم کرد. هر زیرجریان کاری مجزا مجهز به دو عمل دوتایی ۷ و ۸ است و یا جریان کاری (مشبکه) مجهز به دو عمل دوتایی ۷ و ۸ است. به عنوان مثال برای دو گره دلخواه a_5 و a_6 می‌توان نشان داد $a_5 \vee a_6 = a_7$ و $a_5 \wedge a_6 = a_4$ است. در مورد

زیرمجموعه متناهی از عناصر آن دارای سوپریمم و اینفیمم باشد.

حال یک گراف دلخواه مبتنی بر جریان‌های کاری که شامل تعداد متناهی گره است را در نظر بگیرید. این گراف دارای ساختاری به شکل شبکه است، که در آن تمامی یال‌ها، جریان‌هایی به صورت یک طرفه از چشمه‌ها به سوی چاه‌های شبکه هستند. این نوع از شبکه را می‌توان به صورت یک ساختار جبری با دو عملگر تعریف کرد. به عبارت دقیق‌تر، اگر همگی عناصر چاه را (که نشان‌دهنده عناصر پیشین در شبکه هستند)، به یک گره مجازی مانند t_{exit} و کلیه عناصر چشمه را (که نماد عناصر کمینه این مجموعه مرتب جزئی هستند)، به یک گره مجازی مانند t_{entry} متصل کنیم، آنگاه این شبکه جریان متناهی و کراندار الزاما تشکیل یک شبکه می‌دهد (شکل (۱) را ببینید). دقت کنید در این جریان کاری ترکیب فصلی دو عنصر دلخواه a و b کوچکترین عنصری از شبکه است که هر دو عنصر a و b را به عنوان محدودیت تقدیمی داراست. به طور مشابه $a \wedge b$ بیان‌کننده بزرگترین عنصری مانند c در ترتیب جزئی جریان کاری است به طوری که c محدودیت تقدیمی برای هر دو a و b است. از این پس به جهت ساده‌سازی به هر یک از مولفه‌های همبندی از نمودار هاسه مجموعه مرتب جزئی که با حذف t_{entry} و t_{exit} به وجود می‌آیند را یک شبه شبکه گوییم.

حال با استفاده از مفاهیم فوق به طراحی یک الگوریتم اختصاص منابع موازی برای زمانبندی و کمینه کردن هزینه پردازش گره‌ها می‌پردازیم.

۵. روش پیشنهادی

هر جریان کاری می‌تواند از یک یا چند زیرجریان کاری گره مجزا تشکیل شده باشد که هر یک از این زیرجریان‌های کاری معادل با یک شبه شبکه هستند. در نظر داشته باشید که این امر به این دلیل رخ می‌دهد که یک زیرجریان کاری می‌تواند بیش از یک گره شروع و یا بیش از یک گره پایانی داشته باشد. هر زیرجریان کاری به نوبه خود از تعدادی مسیر تشکیل

تعدادی زنجیر است که الزاما هر زنجیر دارای عناصر کمین و بیشین است. زمان t_{ij}^{st} را زمان شروع عنصر کمین زنجیر P_i دلخواه بعلاوه زمان پردازش عناصر این زنجیر بر روی ماشین vm_j در نظر بگیرید. دقت داشته باشید زمان شروع عنصر کمین برابر با زمان خاتمه عنصر مقدم بر کمین در شبکه (با عناصر بیشینه و کمینه مجازی) می‌باشد، یعنی اگر عنصر کمین، گره تالی عنصر کمینه شبکه باشد زمان شروع آن صفر است در غیر این صورت زمان شروع این عنصر برابر با بیشینه زمان خاتمه عناصر مقدم بر آن است (این حالت زمانی رخ می‌دهد که یک زیرجریان کاری اولیه خود به زیرجریان‌های کاری کوچکتر تقسیم گردد و عناصر کمین و بیشین جدید حاصل شود). این زمان‌ها تحت رابطه k یک مجموعه مرتب جزئی هستند. دقت کنید این مجموعه مرتب جزئی مستقل از ماشین مجازی vm_j است، زیرا ماشین‌های مجازی تحت یک ضریب ثابت دارای زمان‌های پردازش متفاوت هستند. حال می‌توان تحت این رابطه طولانی‌ترین مسیر را تعریف کرد.

تعریف ۱: مسیر متناظر با طولانی‌ترین زمان پردازش در رابطه ترتیب جزئی وابسته به یک جریان کاری، معرف طولانی‌ترین مسیر و در نتیجه معرف مسیر بحرانی است.

واضح است که تعریف طولانی‌ترین مسیر خوش‌تعریف است زیرا زمان پردازش مسیر مستقل از vm_j است، اما دقت داشته باشید که طولانی‌ترین مسیر الزاما یکتا نیست.

حال فرض کنید $P: a_1, a_2, \dots, a_k$ طولانی‌ترین زنجیر در زیرجریان کاری باشد. واضح است که a_1 عنصر کمین و a_k عنصر بیشین این زنجیر است. حال تمام عناصر این زنجیر را روی کندترین ماشین ممکن قرار می‌دهیم به نحوی که زمان انقضا را نقض نکند. با توجه به این امر زمان‌های شروع و خاتمه تمام عناصر این زنجیر مشخص خواهد شد. هر زنجیر دلخواهی مانند $P': a_i, b_1, \dots, b_z, a_j$ (که در آن $a_i, a_j \in P$) زمانی کمتر از اختلاف زمان شروع a_i تا خاتمه a_j در P دارد، زیرا در غیر این صورت مسیر a_1 زنجیر P تا

کران‌های پایین می‌توان به a_1, a_2 و a_4 به عنوان مثالی برای کران‌های پایین a_5 و a_6 اشاره کرد، در حالی که a_7 و a_8 کران‌های بالا برای این دو گره هستند. هر مسیر در زیرجریان کاری معرف یک زنجیر است به عنوان مثال $a_1, a_2, a_4, a_5, a_7, a_8$ یک زنجیر است که در آن a_5 تالی عنصر a_4 است و a_2 مقدم بر a_4 است.

توجه کنید که الگوریتم پیشنهادی می‌تواند به صورت موازی برای زیرجریان‌های کاری مجزا استفاده شود، لذا در ادامه الگوریتم پیشنهادی برای هر زیرجریان کاری مجزا بیان می‌شود. نکته حائز اهمیت این است که در ادامه روند الگوریتم پیشنهادی ممکن است هر زیرجریان کاری خود به زیرجریان‌های کاری جدیدی تقسیم شود که در این حالت الگوریتم پیشنهادی به صورت موازی برای آنها نیز قابل استفاده خواهد بود.

واضح است که هر وظیفه در یک زیرجریان کاری باید به ماشین مجازی با کمترین هزینه اختصاص داده شود به طوری که قید زمان اجرا در آن حفظ گردد. فرض کنید k ماشین مجازی به صورت vm_1, \dots, vm_k وجود دارد به طوری که سرعت پردازش اطلاعات این ماشین‌ها متناسب هستند. به بیان دقیق‌تر سرعت پردازش vm_i نسبت به vm_j عدد ثابت c_{ij} می‌باشد. توجه داشته باشید که همواره ماشین‌های سریع‌تر هزینه بیشتری نسبت به ماشین‌های کندتر دارند، اما نسبت هزینه‌ها همچنان ثابت است. برای کاهش هزینه کلی، نیاز است تا وظایف هر یک از زیرجریان‌های کاری مجزا روی کندترین ماشین ممکن قرار گیرند به نحوی که زمان انقضا نقض نشود. برای این منظور ما بایستی زمان‌های شروع و خاتمه را برای هر یک از وظایف یک زیرجریان کاری مشخص کنیم به گونه‌ای که اختصاص دادن کارها به ماشین کندتر امکان‌پذیر نباشد. دقت کنید که این امر به نوبه خود سبب کمینه شدن هزینه می‌شود.

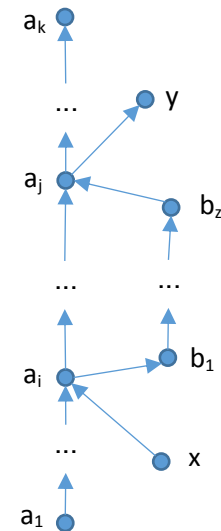
فرض کنید یک جریان کاری و زمان انقضای آن داده شده باشد. همانطور که قبلا نشان داد شد، یک زیرجریان کاری یک شبه شبکه است. هر شبه شبکه در یک جریان کاری دارای

باشد، آنگاه زمان شروع z_1 با زمان خاتمه a_t به روز می‌شود، این امر سبب می‌شود پردازش عناصر تمام زنجیره‌هایی با عنصر بیشین z_1 حتی الامکان بر روی S_i شدن باشد. دقت داشته باشید که تمام مسیرهایی که دارای کمین z_1 هستند نیز بر روی S_i قابل اجرا هستند زیرا در غیر این صورت با طرز انتخاب P در تناقض خواهند بود.

۲. حالت دوم آنکه $a_t = z_1 \vee a_{t-1}$ (که در آن $1 < t \leq k$) باشد، بنابر آنچه در بالا بیان شد تمام مسیرهای منتهی به z_1 الزاما قابل اجرا بر روی S_i می‌باشند زیرا در غیر این صورت با طرز انتخاب P در تناقض است. حال باید زمان خاتمه z_1 با زمان شروع a_t به روز شود. دقت کنید که با توجه به طرز انتخاب طولانی‌ترین مسیر، z_1 تا بیشین (یعنی a_k) حتما دارای زمان طولانی‌تری نسبت به z_1 تا بیشین‌های دیگر است، بنابراین این مسیرها نیز بر روی S_i قابل اجرا هستند. در ادامه ی الگوریتم پیشنهادی، عنصر z_1 را به عنوان عنصر دارای محدودیت معرفی می‌کنیم.

در زیرجریان‌های کاری حاصل از حذف گره‌های مرحله قبل اگر گره‌ای دارای محدودیت زمان شروع باشد، بنابراین از هر گره تا رسیدن به این گره الزاما زمان پردازشی کمتر از محدودیت فعلی وجود خواهد داشت. دلیل این امر آن است که زمان پردازش بیشتر در این زیرمسیر با طرز انتخاب طولانی‌ترین مسیر در مرحله ماقبل در تناقض است. حال اگر مسیر دلخواه $P: a_1, \dots, a_k$ را داشته باشیم به طوری که a_t آخرین گره از این مسیر باشد که دارای محدودیت زمان شروع مانند t_{tj}^{st} است، آنگاه زمان پردازش مسیر P برابر با زمان پردازش گره‌های a_t, \dots, a_k بعلاوه t_{tj}^{st} است. توجه کنید که در این مسیر ممکن است گره‌هایی دارای محدودیت وجود داشته باشند، در این صورت همانطور که در بالا استدلال کردیم، هر یک از عناصر بین گره کمین تا اولین راس دارای محدودیت و همچنین بین هر دو راس دارای محدودیت و در نهایت بین آخرین راس دارای محدودیت و راس بیشین امکان اختصاص منبع دارند. لذا الگوریتم پیشنهادی را به صورت موازی برای

a_i به همراه P' و a_j تا بیشین P (یعنی مسیر $a_1, \dots, a_i, b_1, \dots, b_z, a_j, \dots, a_k$) دارای زمانی طولانی‌تر خواهد بود و این امر متناقض با طرز انتخاب زنجیر P است. مشابه می‌توان نشان داد اگر x عنصر کمین زیرجریان کاری باشد و زنجیری از x تا a_i موجود باشد، آنگاه زمان پردازش این زنجیر کمتر از زنجیر a_1 تا a_i خواهد بود. چنین شرایطی برای عنصر بیشینی مانند y در جریان کاری نیز وجود دارد به نحوی که زنجیر a_j تا y دارای زمانی کمتر از زنجیر a_j تا a_k است. برای نمونه شکل (۲) را که بخشی از یک زیرجریان کاری را نشان می‌دهد، مشاهده کنید. فرض کنید S_i کندترین سرویسی باشد که می‌تواند P را در زمان مدنظر اجرا کند. از آنجایی که مسیر P' زمان کوتاهتری را نسبت به مسیر a_i, \dots, a_j در P دارد، لذا الزاما سرویس متناظر با پردازش این مسیر کندتر یا مساوی با S_i می‌باشد. این امر مشابه برای قسمت‌های دیگر که در بالا به آنها اشاره شد، برقرار است.

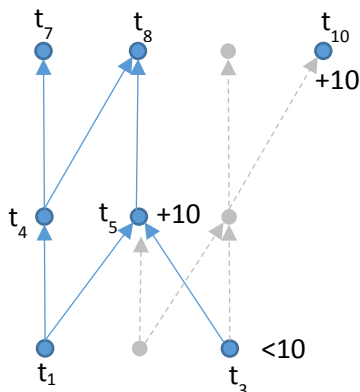


شکل (۲). طولانی‌ترین مسیر در یک زیرجریان کاری

حال گره‌های مسیر P را از زیرجریان کاری حذف می‌کنیم. در این صورت زیرجریان‌های کاری جدیدی ایجاد می‌شود. حال اگر راسی مانند z_1 به یکی از رئوس مسیر P مانند a_t متصل باشد برای این راس دو حالت رخ می‌دهد:

۱. حالت اول آنکه $a_t = z_1 \wedge a_{t+1}$ (که در آن $1 \leq t < k$) باشد و زمان شروع راس z_1 پیش از زمان خاتمه راس a_t

حذف این سه گره از شبکه، دو زیرجریان کاری حاصل می‌شود که الگوریتم پیشنهادی می‌تواند به صورت موازی روی هر کدام از آنها اجرا شود (شکل ۴). برای راحتی در بررسی مثال، ما این مراحل را به صورت پشت سر هم در این بخش شرح می‌دهیم. برای زمان شروع گره t_{10} پس از حذف وظایف بیان شده در بالا، یک قید به صورت $+10$ ایجاد می‌شود. این قید بیان می‌کند که زمان شروع این وظیفه می‌تواند پس از ۱۰ واحد زمانی باشد. با توجه به زمان شروع و پایان t_{10} ، این وظیفه بر روی vm_3 که کندترین ماشین ممکن است، می‌تواند اجرا شود.

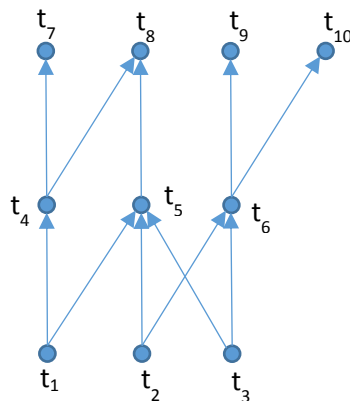


شکل (۴). جریان کاری پس از مرحله اول

برای زیرجریان کاری دوم، به گره‌های t_3 و t_5 به ترتیب قیدهای زمانی <10 و $+10$ اضافه می‌شود. قید <10 در t_3 بیان می‌کند که این وظیفه بایستی قبل از ۱۰ واحد زمانی خاتمه یابد. در ضمن قید زمان شروع برای وظیفه t_5 سبب به روز شدن زمان خاتمه این وظیفه و زمان‌های شروع و خاتمه فرزندان این وظیفه می‌شود (که در جدول (۲) با علامت * مشخص شده اند). در بین ۵ زنجیر این زیرجریان کاری، زنجیرهای t_1, t_5, t_8 ، t_3, t_5, t_8 ، t_5, t_8 دارای طولانی‌ترین زمان هستند. در این شرایط الگوریتم یکی از آنها را به صورت تصادفی انتخاب کرده و اقدام به اختصاص منبع به آن مسیر می‌نماید. فرض کنید مسیر انتخابی t_1, t_5, t_8 باشد. با توجه به داده‌های جدول (۱) این وظایف می‌توانند بر روی vm_3 اجرا شوند (مرحله ۲ از جدول (۲) را مشاهده کنید). با حذف این

هر بخش اجرا می‌کنیم.

برای ایجاد شفایت در روش پیشنهادی، در ادامه آن را بر روی یک مثال بررسی می‌کنیم. شکل (۳) یک جریان کاری ساده با ۱۰ وظیفه t_1 تا t_{10} را نشان می‌دهد. فرض کنید برای زمانبندی جریان کاری ۳ ماشین مجازی vm_1 ، vm_2 و vm_3 در دسترس باشند که زمان‌های اجرای هر یک از وظایف جریان کاری بر روی آنها در جدول (۱) بیان شده است. در ضمن فرض کنید که هزینه اجرای هر واحد زمانی بر روی این سه ماشین مجازی به ترتیب برابر با ۵، ۳ و ۲ بوده و زمان انقضای جریان کاری برابر با ۳۰ واحد زمانی باشد.



شکل (۳). یک مثال از جریان کاری برای بررسی روش پیشنهادی

جدول (۱): زمان اجرای وظایف شکل (۳) روی سه ماشین مجازی

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
vm_1	۲	۵	۳	۴	۳	۴	۵	۳	۵	۴
vm_2	۴	۱۰	۶	۸	۶	۸	۱۰	۶	۱۰	۸
vm_3	۶	۱۵	۹	۱۲	۹	۱۲	۱۵	۹	۱۵	۱۲

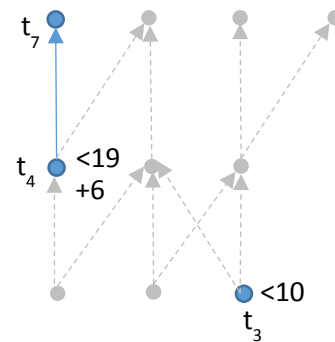
همانطور که در شکل (۳) مشاهده می‌شود، جریان کاری در ابتدا شامل تنها یک زیرجریان کاری (معادل یک شبه شبکه) است. در ضمن بنابر تعریف طولانی‌ترین مسیر (تعریف ۱)، زنجیر t_2, t_6, t_9 مسیر بحرانی زیرجریان کاری را تشکیل می‌دهد. بر اساس داده‌های جدول (۱) این زنجیر بر روی vm_2 قابل اجرا است. با اختصاص این ماشین مجازی به وظایف t_2 ، t_6 ، t_9 و پایان این سه وظیفه مشخص شده و آنها به عنوان پردازش شده علامت زده می‌شوند (مرحله ۱ از جدول (۲) را مشاهده کنید). حال با

بدون دور است. گویم گره a به گره b متصل است هرگاه یالی جهت‌دار از a به b در جریان کاری موجود باشد. برای محاسبه حداکثر تعداد یال‌های این گراف، در گام اول یک راس دلخواه مانند t_1 را در نظر بگیرید. واضح است که حداکثر تعداد راس‌هایی که t_1 می‌تواند به آنها متصل شود (اشاره کند) برابر با $n - 1$ است. حال فرض کنید که گره انتخابی دوم t_2 باشد. این گره نمی‌تواند به t_1 متصل شود، زیرا این امر سبب ایجاد دور می‌شود. لذا t_2 می‌تواند به $n - 2$ گره اشاره کند که سبب ایجاد $n - 2$ یال خواهد شد. در ضمن توجه داشته باشید که در یک گراف جریان کاری هر گره نمی‌تواند به خودش اشاره کند. در ادامه این روند، هر گره می‌تواند به تمام گره‌ها به جز گره‌های قبلی خود اشاره کند و در نهایت گره آخر به هیچ گره‌ای اشاره نخواهد کرد. بنابراین آنچه بیان شد مجموع یال‌ها در یک گراف جریان کاری برابر با $\sum_{i=0}^{n-1} i$ خواهد بود. این مجموع نشان‌دهنده حداکثر تعداد یال‌ها و برابر با $n(n - 1)/2$ است. لذا بزرگی زمانی یال‌ها که نشان‌دهنده وابستگی‌ها در جریان کاری هستند برابر با $O(n^2)$ است.

در گام اول الگوریتم پیشنهادی، مقادیر زمان شروع و خاتمه برای کلیه گره‌های جریان کاری بر حسب سریع‌ترین ماشین مجازی محاسبه می‌شود. این کار از گره ابتدایی جریان کاری (یعنی t_{entry}) آغاز شده و سپس زمان‌های شروع و خاتمه برای فرزندان این گره محاسبه می‌شوند و این روند تا گره انتهایی جریان کاری ادامه می‌یابد. از آنجایی که تعداد گره‌های جریان کاری برابر با n است و روش بیان شده هر گره را فقط یکبار بررسی می‌کند، لذا بزرگی زمانی این گام از الگوریتم برابر با $O(n)$ خواهد بود. در گام دوم الگوریتم پیشنهادی، یک مجموعه مرتب جزئی که شامل کلیه زنجیره‌های شبکه معادل با جریان کاری است، ساخته می‌شود. دقت کنید که هر زنجیره باید شامل عنصر جدیدی باشد. از آنجایی که زنجیره‌ها از طولانی‌ترین به کوتاه‌ترین ایجاد و مرتب می‌شوند و در ضمن الگوریتم پیشنهادی نیز عناصر زنجیره‌ها را فقط یکبار از طولانی‌ترین به کوتاه‌ترین بررسی کرده و ماشین مجازی

وظایف دو قید زمان $+6$ و <19 به گره t_4 اضافه خواهد شد (شکل ۵).

توجه داشته باشید که الگوریتم پیشنهادی مسیریایی که دارای قید انتهایی هستند را به صورت مجزا زمانبندی می‌کند. بنابراین در زنجیر t_4, t_7 ، وظیفه t_4 به صورت مستقل از t_7 زمانبندی می‌شود و می‌تواند روی یک پردازنده کندتر اجرا شود به شرطی که قید زمان انقضا نقض نشود. با توجه به زمان‌های اجرا، وظیفه t_4 بر روی vm_3 و وظیفه t_7 بر روی vm_2 اجرا می‌شوند. در ضمن وظیفه t_3 نیز به صورت موازی بررسی شده و بر روی vm_3 زمانبندی می‌شود (مرحله ۳ از جدول (۲) را مشاهده کنید).



شکل (۵). جریان کاری پس از مرحله دوم

جدول (۲): زمانبندی وظایف شکل (۳)

زمان	وظیفه									
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
t_1 شروع	0	0	0	2	5	5	6	8	9	9
t_1 خاتمه	2	5	3	6	8	9	11	11	14	13
t_2 شروع	0	0	0	2	+10	10	6	13*	18	+18
t_2 خاتمه	2	10	<10	6	13*	18	11	16*	28	22*
t_3 شروع	0	0	0	+6	10	10	10*	19	18	18
t_3 خاتمه	6	10	<10	<19	19	18	15*	28	28	30
t_4 شروع	0	0	0	6	10	10	18	19	18	18
t_4 خاتمه	6	10	9	18	19	18	28	28	28	30
vm	3	2	3	3	3	2	2	3	2	3

برای محاسبه بزرگی زمانی الگوریتم پیشنهادی، فرض کنید به عنوان ورودی یک گراف جریان کاری با n گره داده شده باشد. دقت کنید که گراف جریان کاری، یک گراف جهت‌دار

می‌تواند هزینه اجرای به مراتب کمتری از دو روش IC-PCP و IC-PCPD2 در جریان‌های کاری Montage و LIGO را ارائه دهد.

۶. ارزیابی روش پیشنهادی

در این بخش، شبیه‌سازی‌های انجام شده برای ارزیابی روش پیشنهادی را بررسی می‌کنیم. برای نشان دادن کارایی روش پیشنهادی آن را با دو روش IC-PCP و IC-PCPD2 [۸] مورد مقایسه قرار داده‌ایم.

قبل از بررسی ارزیابی لازم می‌دانیم برخی تفاوت‌های اساسی و مزایای مابین روش ارائه شده در این مقاله را با روش‌های پیشنهادی در [۸] را که در منابع علمی بسیاری تا کنون مورد استفاده و کاربرد قرار گرفته است را ذکر نماییم. در مرجع مذکور، طولانی‌ترین مسیر وابسته به ماشین مجازی است. این امر باعث می‌شود که الگوریتم معرفی شده توسط نویسندگان هر بار با انتخاب یک ماشین مجازی، رفتاری متفاوت از قبل داشته باشد. با این حال نویسندگان سعی کرده‌اند با ترفندی از این امر تبری جویند، به طوری که آنها بدون ذکر دلیل انتخاب طولانی‌ترین مسیر را تنها بر اساس سریع‌ترین ماشین بدست آورده‌اند، در حالی که ممکن است اصلاً سریع‌ترین ماشین هیچ‌گاه در جریان کاری مورد استفاده قرار نگیرد. نکته دوم آنکه در این الگوریتم همواره برای تخمین زودترین زمان شروع از زودترین زمان خاتمه گره‌های مقدم بر یک گره بعلاوه زمان انتقال داده استفاده می‌شود، که این امر به نوبه خود شامل دو مشکل اساسی است. ابتدا آنکه ممکن است دو گره متوالی بر روی یک ماشین باشند و نیازی به زمان انتقال نداشته باشند در حالی که همواره این امر محاسبه شده است. این امر سبب می‌شود که برای گره‌های مابعد گره مورد نظر زمان کمتری جهت اجرا تخصیص یابد. به عبارت دیگر مشکل اساسی‌تر دوم در این روش آن است که اگر گره‌های مابعد یک گره به تعداد زیادی باشند و گره‌های مقدم تعداد کمی داشته باشند، از آنجایی که الگوریتم مسیر بحرانی جزئی زمان انتقال را به گره‌های ماقبل گره مورد نظر منتقل کرده است به اجبار

مناسب را برای آنها مشخص می‌کند، لذا وجود زنجیره‌ای که تمام عناصر آن تکراری است ضرورتی ندارد. ایجاد مجموعه مرتب جزئی به صورت بازگشتی از عنصر بیشینه مشبکه آغاز شده و هر بار والدی انتخاب می‌شود که دارای بیشترین زمان خاتمه است. این امر سبب می‌شود هر وظیفه در جریان کاری تنها یک بار بررسی و تنها زنجیره‌های دارای عنصر جدید ایجاد شوند و در نتیجه بزرگی زمانی ساختن مجموعه مرتب جزئی برابر با $O(n)$ گردد. در گام بعدی، برای زمانبندی یک وظیفه در مسیر بحرانی، الگوریتم پیشنهادی ماشین‌های مجازی را برای یافتن کندترین ماشین که زمان انقضا را نیز نقض نکند، بررسی می‌کند. از آنجایی که سرعت پردازش ماشین‌های مجازی توسط نوع آنها مشخص می‌شود (بخش ۳ را ببینید)، بنابراین کافی است برای هر وظیفه m نوع ماشین مجازی، h_1, h_2, \dots, h_m بررسی شود. با توجه به این که زمان شروع و پایان هر گره در جریان کاری تنها یک بار محاسبه می‌شود، می‌توان نتیجه گرفت که بزرگی زمانی اختصاص ماشین مجازی به کلیه گره‌های جریان کاری برابر با $O(mn)$ است. دقت داشته باشید که در بدترین حالت هر کار می‌تواند روی یک نوع ماشین مجازی متفاوت زمانبندی شود. به بیان دیگر تعداد حداکثر انواع ماشین مجازی می‌تواند برابر با تعداد کارهای جریان کاری باشد. این امر سبب می‌شود بزرگی زمانی مرحله زمانبندی وظایف به $O(n^2)$ تغییر کند. در گام انتهایی روش پیشنهادی، به کلیه گره‌های متصل به گره‌های مسیر بحرانی قیده‌های شروع و پایان اختصاص داده می‌شود. از آنجایی که برای هر گره تنها یک بار زمان شروع و خاتمه محاسبه می‌شود و در گراف جریان کاری هر گره حداکثر می‌تواند به $n-1$ گره متصل باشد، بنابراین بزرگی زمان اختصاص قیود برابر با $O(n^2)$ است. بر اساس آنچه بیان شد، بزرگی زمانی کلی الگوریتم پیشنهادی برابر با مجموع بزرگی زمانی گام‌های الگوریتم و برابر با $O(n^2)$ است.

اگرچه بزرگی زمانی روش پیشنهادی برابر با بزرگی زمانی دو روش IC-PCP و IC-PCPD2 [۸] است، اما نتایج حاصل از مقایسه این سه روش نشان می‌دهد که روش پیشنهادی

از آنجایی که معیار کارایی روش پیشنهادی، کمینه کردن هزینه است، در ادامه به مقایسه این معیار با سایر روش‌ها خواهیم پرداخت. با توجه به ویژگی‌های مختلف جریان‌های کاری استفاده شده، نرمال کردن هزینه‌های اجرا اهمیت بسزایی دارد [۸]. میزان یک هزینه را نرمال‌سازی شده گوئیم، هرگاه هزینه اجرای یک روش را بر یک هزینه پایه تقسیم کنیم. این هزینه پایه برابر با ارزان‌ترین زمانبندی است.

برای یافتن این زمانبندی، کل جریان کاری بر روی تنها یک نوع ماشین مجازی که دارای کمینه هزینه است، اجرا می‌شود. حال هزینه نرمال‌شده جریان کاری W به صورت زیر تعریف می‌شود [۸]:

$$NormalizedCost = \frac{C_w^{app}}{C_w^{cheap}} \quad (3)$$

در این رابطه C_w^{app} نشان‌دهنده هزینه حاصل از زمانبندی جریان کاری با روش app و C_w^{cheap} بیان‌کننده هزینه پایه است.

جدول (۴): مشخصات ماشین‌های مجازی

ماشین مجازی			
نوع ۳	نوع ۲	نوع ۱	
4	2	1	تعداد هسته‌ها
300	200	100	قدرت پردازش (MIPS)
2048	1024	512	رم (MB)
2048	1024	512	حافظه (MB)
2048	1024	512	پهنای باند (Mbps)

شکل‌های (۶) و (۷) هزینه نرمال شده را برای روش‌های مورد مقایسه نشان می‌دهد. نتایج نشان می‌دهد که روش پیشنهادی در جریان کاری Montage نسبت به IC-PCP به میزان ۷٪ و نسبت به IC-PCPD2 به میزان ۱۴/۵٪ هزینه را کاهش داده است. در ضمن در هزینه روش پیشنهادی در جریان کاری LIGO به ترتیب به میزان ۴ و ۷/۵ درصد نسبت به IC-PCP و IC-PCPD2 بهبود داشته است.

دقت داشته باشید که هزینه نرمال‌شده در الگوریتم IC-PCPD2 نسبت به IC-PCP با افزایش اندازه جریان کاری تقریباً ثابت

گره‌های مابعد این گره را دچار محدودیت زمانی کرده است. اما واضح است که اگر عکس این عمل رخ دهد یعنی تعداد گره‌های ماقبل گره‌ای که زمان انتقال را در آن استفاده می‌کنیم زیاد باشد و گره‌های مابعد وجود نداشته باشد یا کم باشد (دقیقا مانند مثال ذکر شده در مقاله که هیچ‌گاه گره‌های مابعد برای گره‌هایی که زمان انتقال را در آنها محاسبه می‌شود، وجود ندارد) در این صورت استفاده از زودترین زمان شروع ارائه شده در مقاله ارزشمند است. لذا نمی‌توان با توجه به دلایل ذکر شده استفاده از این روش را الزاما ارزشمند دانست. لذا در این مقاله ما زمان انتقال را بین ماشین‌ها همواره صفر در نظر می‌گیریم. واضح است که می‌توانیم مانند آنچه در مقاله [۸] ذکر شده است زمان انتقال را دخیل کرد ولی ما با حذف زمان انتقال حداقل به الگوریتم روشن‌تر و قابل‌اتکاتری می‌رسیم.

در شبیه‌سازی‌ها، دو نوع مختلف از جریان‌های کار واقعی شامل Montage و LIGO در نظر گرفته شده است. از این جریان‌های کاری به طور گسترده برای ارزیابی عملکرد الگوریتم‌های زمانبندی استفاده شده است [۱]. علاوه بر این، سه اندازه مختلف برای مقایسه نتایج در نظر گرفته شده است که عبارتند از کوچک (تا ۱۰۰ وظیفه در جریان کاری)، متوسط (تا ۵۰۰ وظیفه در جریان کاری) و بزرگ (بین ۵۰۰ تا ۱۰۰۰ وظیفه در جریان کاری). برای پیاده‌سازی روش پیشنهادی از ابزار WorkflowSim استفاده شده است. محیط شبیه‌سازی شامل یک مرکز داده است که از سه نوع ماشین مجازی مختلف تشکیل شده است. مشخصات مرکز داده و مشخصات ماشین‌های مجازی به ترتیب در جداول (۳) و (۴) ذکر شده‌اند.

جدول (۳): مشخصات مرکز داده

پارامتر	مقدار
شناسه هاست	1-20
تعداد هسته‌ها	4
قدرت پردازش	20000 MIPS
رم	204800 MB
حافظه	1000000 MB
پهنای باند	102400 Mbps

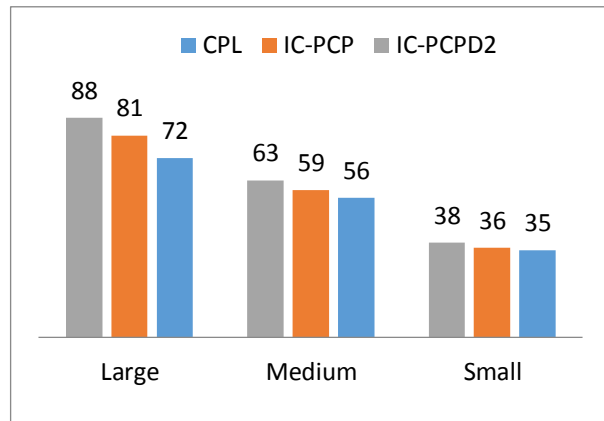
جدید از مسیر بحرانی، طولانی‌ترین مسیر را یافته و آن را به کندترین ماشین اختصاص می‌دهد به نحوی که زمان انقضای آن مسیر نقض نشود. با زمانبندی گره‌های مسیر بحرانی یکسری قید زمان برای گره‌های تالی و مقدم، گره‌های این مسیر ایجاد می‌شود. الگوریتم پیشنهادی با در نظر گرفتن این قیود اقدام به زمانبندی وظیفه‌های باقی مانده در زیرجریان‌های کاری جدید می‌کند. تابع هدف الگوریتم پیشنهادی کمینه کردن هزینه جریان کاری است. نتایج حاصل از شبیه‌سازی‌ها بیانگر بهینه‌تر بودن راهکار پیشنهادی از نظر هزینه در مقایسه یا سایر روش‌ها در جریان‌های کاری واقعی است.

از آنجایی که در این مقاله ما به بیان زمانبندی ایستا که در آن ساختار جریان کاری و وابستگی‌های بین وظایف از پیش تعریف شده است، پرداخته‌ایم، به عنوان کار آتی، در نظر داریم الگوریتم پیشنهادی را برای زمانبندی پویا گسترش دهیم. از طرف دیگر، با آنکه نتایج حاصل از شبیه‌سازی بیانگر کارایی بیشتر روش پیشنهادی است، اما همچنان این روش از بهینه سراسری فاصله دارد. دلیل این امر ماهیت اکتشافی روش ارائه شده است. لذا برای ادامه کار، ارائه یک مدل بهینه‌سازی بر پایه مفاهیم بیان شده در این مقاله پیشنهاد می‌شود. دقت داشته باشید که نحوه رسیدن به نقطه بهینه در چنین مدل‌هایی می‌تواند با استفاده از استدلال‌های بهینه‌سازی بررسی شود.

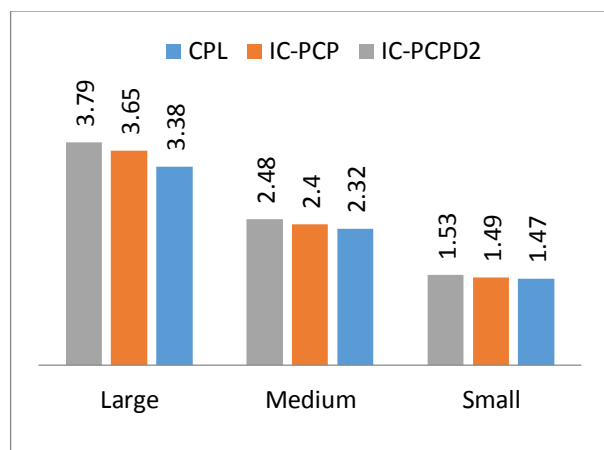
به عنوان یک کار آتی دیگر، محققان می‌توانند با استفاده از ابزارهای حدی در ریاضیات میزان نزدیکی جواب‌های به دست آمده را نسبت به بهینه سراسری (با افزایش تعداد وظایف جریان کاری) یافته و راه‌کاری برای ارتقای روش پیشنهاد ارائه کنند. علاوه بر موارد بیان شده، توجه به زمان‌های انتقال داده بین وظایف در زمانبندی جریان‌های کاری می‌تواند به عنوان ادامه کار این مقاله در نظر گرفته شود.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

بوده است. کاهش بهتر هزینه روش پیشنهادی نسبت به دو روش دیگر در جریان کاری Montage نسبت به جریان کاری LIGO مربوط به تفاوت در ساختار این دو جریان کاری است.



شکل ۶. مقایسه هزینه نرمال شده برای جریان کاری MONTAGE



شکل ۷. مقایسه هزینه نرمال شده برای جریان کاری LIGO

۷. نتیجه‌گیری و کارهای آتی

زمانبندی جریان‌های کاری به عنوان یک مسئله زمان چندجمله‌ای غیرقطعی سخت (NP-complete) شناخته می‌شود، لذا ارائه راه‌حلی ابتکاری در این حوزه اهمیت بسزایی دارد. در این مقاله ما یک الگوریتم زمانبندی برای جریان‌های کاری بر اساس مسیر بحرانی ارائه کردیم. روش پیشنهادی به صورت موازی برای هر زیرجریان کاری گره مجزا، یک شبهه شبکه ایجاد کرده و سپس زنجیره‌های هر شبهه شبکه را لیست می‌کند. آنگاه در بین زنجیره‌های لیست شده بر اساس تعریف

- [1] Juve G., Chervenak A. L., Deelman E., Bharathi S., Mehta G., Vahi K., "Characterizing and profiling scientific workflows," *Future Generation Comp. Syst.* vol. 29, no. 3, pp. 682-692, 2013.
- [2] Arunarani A. R., Manjula D., Sugumaran V., "Task scheduling techniques in cloud computing: A literature survey," *Future Gener. Comput. Syst.*, vol. 91, pp. 407-415, 2019.
- [3] Mershad K., Artail H., Saghir M. A. R., Hajj H., Awad M., "A study of the performance of a cloud datacenter server," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 590-603, 2017.
- [4] Sadooghi I., Martin J. H., Li T., Brandstatter K., Maheshwari K., de Lacerda Ruivo T. P. P., Garzoglio G., Timm S., Zhao Y., Raicu I., "Understanding the performance and potential of cloud computing for scientific applications," *IEEE Transactions on Cloud Computing*, vol. 5, no. 2, pp. 358-371, 2017.
- [5] Topcuoglu H., Hariri S., Wu M., "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 3, pp. 260-274, 2002.
- [6] Son J. H., Kim J., Kim M., "Extracting the workflow critical path from the extended well-formed workflow schema," *J. Comput. Syst. Sci.*, vol. 70, no. 1, pp. 86-106, 2005.
- [7] Abrishami S., Naghibzadeh M., Epema D. H. J., "Cost-Driven Scheduling of Grid Workflows Using Partial Critical Paths," *IEEE Trans. Parallel Distributed Syst.*, vol. 23, no. 8, pp. 1400-1414, 2012.
- [8] Abrishami S., Naghibzadeh M., Epema D. H. J., "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," *Future Generation Comp. Syst.*, vol. 29, no. 1, pp. 158-169, 2013.
- [9] Wu F., Wu Q., Tan Y., Li R., Wang W., "Pcp-b2: Partial critical path budget balanced scheduling algorithms for scientific workflow applications," *Future Generation Comp. Syst.*, vol. 60, pp. 22-34, 2016.
- [10] Yuan Y., Li X., Wang Q., Zhu X., "Deadline division-based heuristic for cost optimization in workflow scheduling," *Inf. Sci.*, vol. 179, no. 15, pp. 2562-2575, 2009.
- [11] Jailalita, Singh S., Dutta M., "Critical path based scheduling algorithm for workflow applications in cloud computing," in: *2016 International Conference on Advances in Computing, Communication, Automation (ICACCA) (Spring)*, pp. 1-6, 2016.
- [12] Calheiros R.N., Buyya R., "Meeting deadlines of scientific workflows in public clouds with tasks replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1787-1796, 2014.
- [13] Cai Z., Li X., Gupta J. N. D., "Critical path-based iterative heuristic for workflow scheduling in utility and cloud computing," in: *Service-Oriented Computing - 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5*, pp. 207-221, 2013.
- [14] Poola D., Garg S. K., Buyya R., Yang Y., Ramamohanarao K., "Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds," *AINA*, pp. 858-865, 2014.
- [15] Arabnejad V., Bubendorfer K., Ng B., Chard K., "A deadline constrained critical path heuristic for cost-effectively scheduling workflows," in: *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, pp. 242-250, 2015.
- [16] Arabnejad V., Bubendorfer K., Ng B., "Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources," *Future Generation Comp. Syst.*, vol. 75, pp. 348-364, 2017.
- [17] Ghafouri R., Movaghar A., Mohsenzadeh M., "A budget constrained scheduling algorithm for executing workflow application in infrastructure as a service clouds," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 1, pp. 241-268, 2019.
- [18] Matani A., Darvisay A., "A Novel Critical-Path Based Scheduling Algorithm for Stochastic Workflow in Distributed Computing Systems," in *International Congress on High-Performance*

Computing and Big Data Analysis, pp. 476-489, 2019.

[19] Pan Y., Wang S., Wu L., Xia Y., Zheng W., Pang S., Zeng Z., Chen P., Li Y., "A Novel Approach to Scheduling Workflows Upon Cloud Resources with Fluctuating Performance," *Mob. Networks Appl.*, vol. 25, no. 2, pp. 690-700, 2020.

[20] Davey B. A., Priestley H. A., "Introduction to Lattices and Order," Cambridge University Press, ISBN 978-0-521-78451-1, 2002.