

WSTMOS: A Method For Optimizing Throughput, Energy, And Latency In Cloud Workflow Scheduling

Arash Ghorbannia Delavar^{*}, Reza Akraminejad^{**}, Sahar Mozafari^{***}

^{*} Department of Computer Engineering and Information Technology, Payam Noor University of Tehran, P.O. Box 19395-4697, Tehran, Iran

^{**} Department of Computer Engineering and Information Technology, Payam Noor University of Tehran, Tehran, Iran

^{***} Department of Computer Engineering and Information Technology, Payam Noor University of Tehran, Tehran, Iran

Abstract

The application of cloud computing in different datacenters around the world has led to the generation of more CO₂ gas. In this paper, the energy and throughput-aware algorithm for scheduling instance-intensive IoT workflows with batch processing in clouds is studied, and a method for scheduling cloud workflows to optimize energy, throughput, and latency is introduced. In the proposed method, throughput, energy, and latency have been improved in comparison to previous methods by creating distance parameters, clustering inputs, and considering real execution time. In the WSTMOS method by considering special parameters and real execution time, we managed to reach the optimized objective function. Moreover, in the proposed method parameter of the time distance of tasks to virtual machines for decreasing the number of migrations in virtual machines was applied. In the studied method clustering the workflow inputs to low, medium, and high load groups and also distributing the load on more suitable servers by the use of the threshold detector, were done. The average energy, cost, average latency, throughput, and load balancing were optimized in comparison to previous studies. Finally, energy and cost were optimized on average by 4.8 and 4.4 percent respectively.

Keywords

Scheduling, Cloud Computing, Load Balancing, Energy, Cost, Batch Processing, Throughput.

WSTMOS: روشی جهت بهینه سازی توان عملیاتی، انرژی و تاخیر در زمانبندی جریان‌های کاری ابری

آرش قربان نیا دلاور^{*}، رضا اکرمی نژاد^{**}، سحر مظفری^{**}

^{*} عضو هیأت علمی، گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه پیام نور تهران، ص. پ. ۴۶۹۷-۱۹۳۹۵، تهران، ایران

^{**} دانشجوی کارشناسی ارشد، گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه پیام نور تهران، تهران، ایران

تاریخ پذیرش: ۱۴۰۱/۱۰/۱۰

تاریخ دریافت: ۱۴۰۱/۰۵/۰۶

نوع مقاله: پژوهشی

چکیده

استفاده از رایانش ابری در مراکز داده مختلف در سراسر دنیا، منجر به تولید بیشتر گاز دی‌اکسیدکربن می‌شود، که در آن مسئله انرژی و توان یکی از مسائل مهم می‌باشد. در این مقاله، الگوریتم آگاه به انرژی و توان عملیاتی برای زمان‌بندی جریان‌های کاری نمونه - فشرده اینترنت اشیا با پردازش دسته‌ای در ابرها مورد مطالعه قرار گرفته و روشی جهت زمان‌بندی جریان‌های کاری ابری برای بهینه سازی انرژی، توان عملیاتی و تاخیر ارائه شده است. در روش پیشنهادی نسبت به روش قبلی، با ایجاد پارامترهای فاصله، دسته‌بندی ورودی‌ها و همچنین زمان اجرای واقعی، توان عملیاتی، انرژی و تاخیر بهینه شده است. روش WSTMOS با در نظر گرفتن پارامترهای شاخص و زمان واقعی، به تابع صلاحیت بهینه‌ای دست یافته است. همچنین در روش پیشنهادی از پارامتر فاصله زمانی وظیفه نسبت به ماشین‌های مجازی، برای کاهش تعداد مهاجرت‌های ماشین‌های مجازی، استفاده شده است. در این مطالعه با دسته‌بندی ورودی‌های جریان کاری به گروه‌های کم، متوسط و پر حجم و همچنین توزیع بار مناسب بر روی سرورهای مناسب‌تر با استفاده از آشکارساز آستانه پردازنده‌ها، انجام شده است. میانگین انرژی، هزینه و میانگین تأخیر، توان و بار کاری نسبت به روش‌های قبلی بهینه شده است. در نهایت میزان انرژی و هزینه به طور میانگین ۴.۸ و ۴.۴ درصد، نسبت به روش مورد مطالعه کاهش یافته است.

واژه‌های کلیدی: زمان‌بندی، محاسبات ابری، متعادل سازی بار، انرژی، هزینه، پردازش دسته‌ای، توان عملیاتی.

۱. مقدمه

تا حدود ۳۰۰ میلیون وات را مصرف کرده‌اند. در سال ۲۰۱۳، مراکز داده‌ها در آمریکا، مجموعاً ۹۱ میلیارد کیلووات‌ساعت انرژی الکتریکی مصرف کرده و ۹۷ میلیون تن دی‌اکسیدکربن تولید کرده‌اند [۴]. انتظار می‌رود انتشار دی‌اکسیدکربن تولید شده توسط فناوری اطلاعات و ارتباطات، از ۱,۴ میلیارد تن متریک فراتر رود. تخمین زده می‌شود که مراکز داده‌ها مسئول حدود ۱۸٪ از کل انرژی مصرف‌شده توسط همه سیستم‌های اطلاعاتی و ارتباطی در جهان هستند؛ بنابراین، بسیاری از رویکردهای کارآمد در سطح ساختمان، در سیستم‌های خنک‌کننده، در شبکه مرکز داده‌ها و با استفاده از منابع محاسباتی، مورد بررسی قرار گرفته‌اند.

یکی از مسائل مهم نخبگان علمی در دهه ۲۰ زمان‌بندی جریان کار ابری می‌باشد که بهینه سازی توان عملیاتی و همچنین مصرف انرژی با انتخاب پارامترهای شاخص انجام شده است و همچنین کاهش تاخیر در طبقه‌بندی ورودی‌ها نکته‌ای مهم می‌باشد که در آن، هزینه و گرمای مرکز داده با ایجاد یک تعادل بار بهینه شده است [۲].

وظایف را با توجه به زمان درخواست و تصدیق، محاسبه و پارامتر فاصله در نظر گرفته‌ایم و دسته‌بندی وظایف را نیز به سه دسته کم، متوسط و با حجم بالا انجام داده‌ایم. در ادامه، ساختار تحقیق به شرح زیر است. در بخش ۲ به بررسی کارهای مرتبط می‌پردازیم. در بخش ۳ روش پیشنهادی را مورد بحث قرار می‌دهیم. در بخش ۴ الگوریتم، شکل، تابع هدف و نوآوری‌های ارائه شده را بیان می‌کنیم. در بخش ۵ پارامترها و نتایج شبیه‌سازی و اطلاعات و نمودارهای به‌دست‌آمده را بررسی می‌کنیم. در انتها نیز در بخش ۶ به کارهای آینده می‌پردازیم.

۲. کارهای پیشین

مسئله مصرف انرژی و گرمای توان پردازشی پردازنده‌ها به یکی از نگرانی‌های اصلی در سراسر جهان تبدیل شده است. تحقیقات و آثار زیادی در تلاش هستند تا مصرف انرژی را در ابرها را بهینه نمایند. مدل مصرف انرژی در سیستم‌های کامپیوتری، اولین مشکلی است که باید حل شود. علاوه بر مصرف انرژی، هزینه نیز یکی از مسائل می‌باشد که در نظر گرفته نشده است [۲].

در [۱] یک روش زمان‌بندی وظیفه ایستای چندهدفه برای طراحی دستگاه‌های نهفته ارائه شده است. وظایف به صورت یک گراف، مدل شده و با در نظر گرفتن یک زیرساخت سخت‌افزاری برای سیستم نهفته، روشی برای نگاشت و زمان‌بندی وظایف بر روی معماری سخت‌افزاری ارائه شده است. در این روش زمان‌بندی پارامترهای زمان اجرای وظایف، انرژی مصرفی به‌عنوان اهداف بهینه‌سازی طی یک الگوریتم بهینه‌سازی ژنتیک بهینه شده است اما دسته‌بندی ورودی‌ها و آستانه در نظر گرفته نشده است تا با استفاده از آن‌ها به انرژی بهینه‌تری برسد.

در رویکرد [۴]، روش زمان‌بندی آگاه به انرژی برای ارسال وظایف به ماشین‌های فیزیکی با کم کردن مصرف انرژی تا حد امکان، برای کاهش مصرف انرژی و هزینه انجام گرفته شده است. با توجه به دسته‌بندی ورودی‌ها و همچنین فاصله می‌توان انرژی و هزینه را کاهش داد. الگوریتم‌های تکاملی بسیاری برای یافتن یک طرح زمان‌بندی گردش کار مؤثر پیشنهاد شده‌اند. در رویکرد [۲]، روشی دو هدفه بر مبنای الگوریتم ژنتیک ایجاد کردند که زمان پاسخ را کوتاه می‌کند.

در [۵] نیز یک تکنیک مهاجرت ماشین مجازی، مبتنی بر الگوریتم Firefly را انجام گرفته است.

استراتژی پیشنهادی در [۱۲]، به طور پویا ماشین مجازی را به ماشین‌های فیزیکی اختصاص داده است تا میزان مصرف انرژی و توافق سطح خدمات را بهینه کند، درحالی‌که هدف اولیه رویکرد PPRGear^۴ این است که مصرف انرژی را بهینه نماید، اما دسته‌بندی ورودی‌ها و فاصله در نظر گرفته نشده است.

مجازی‌سازی، یک فناوری کلیدی برای دستیابی به بهره‌وری انرژی در مراکز داده‌ها است. فرایند تخصیص و زمان‌بندی منابع، روشی است که منابع موجود را به یک وظیفه اختصاص می‌دهند. ماشین‌های مجازی را می‌توان، ایجاد و حذف کرد و بسته به تصمیمات آگاه به توان، بین کامپیوترهای میزبان انتقال داد. مدیریت حافظه مجازی کارآمد در برنامه‌ریزی کار، تثبیت حجم کاری، خوشه‌بندی درخواست‌ها، انتخاب خدمات ابری یا محلی و غیره از مواردی هستند که حائز اهمیت بوده و همیشه مورد توجه قرار می‌گیرند [۴].

اختصاص منابع، به مقدار و زمان مشخصی انجام می‌شود. زمان‌بندی جریان کار برای این منظور استفاده می‌شود تا به وظایفی که به یکدیگر مرتبط هستند، زمان خاصی اختصاص داده شده و همچنین به ترتیب معینی اجرا شوند تا این تقسیم کار به صورت بهینه انجام شود.

اختصاص منابع، به مقدار و زمان مشخصی انجام می‌شود. زمان‌بندی جریان کار برای این منظور استفاده می‌شود تا به وظایفی که به یکدیگر مرتبط هستند، زمان خاصی اختصاص داده شده و همچنین به ترتیب معینی اجرا شوند تا این تقسیم کار به صورت بهینه انجام شود.

فرایند زمان‌بندی به دو صورت ثابت و پویا انجام می‌شود:

- **زمان‌بندی ثابت:** در این مدل، زمان‌بند تمام اطلاعات در مورد هر وظیفه و منبع را قبل از خود وظیفه می‌داند. بار اضافی زمان اجرا، کمینه است.
- **زمان‌بندی پویا:** قبل از اجرا، از اطلاعات آگاهی ندارد. در این زمان‌بندی، بار اضافی زمان اجرا بیشینه است. کل زمان اجرای وظیفه، تا زمانی که اجرا انجام نشود، معلوم نیست [۴].

در این تحقیق الگوریتم زمان‌بندی آگاه به انرژی و هزینه را، همراه با پردازش خوشه‌ای^۲ برای جریان‌های کاری نمونه - فشرده اینترنت اشیا^۳ در ابر بررسی کرده و روش جدیدی بر مبنای این الگوریتم ارائه داده‌ایم. همچنین میزان توان مصرفی، زمان تأخیر و پارامتر متعادل‌سازی بار را با چارچوب تغییر مسیر پویا که در دیتاست Microsoft Azure نسخه ۲۰۱۹ پیاده‌سازی شده است، مورد مطالعه قرار گرفته شده است [۳]. در مطالعه ارائه شده، زمان واقعی

¹ Runtime overload
² Batch processing
³ Internet of Things (IoT)

⁴ Performance-to-power-ratio

صرف انرژی موردی انجام نشده است.

در مقاله [۱۶] عوامل تأثیرگذار بر خدمات ابری از جمله زمان اجرا و بار کاری در نظر گرفته شده است و سه سناریوی بار کاری مختلف (۱) الگوی بار کاری زیاد و قابل پیش‌بینی (۲) الگوی بار کاری زیاد و غیر قابل پیش‌بینی و (۳) الگوی بار کاری روشن - خاموش بررسی شده است. اما به مسئله‌ی انرژی و هزینه اشاره‌ای نشده است.

در تحقیق [۱۶] برای بررسی رفتار مصرف‌کنندگان در فضای ابری راهکارهای مناسبی جهت کاهش هزینه و ضایعات برای این فضای ابری وجود دارد. این مطالعه به بررسی این موارد با انجام یک تحلیل جامع از سوابق مایکروسافت ۲۰۱۹ Azure پرداخته و گام‌های ذیل را در پی گرفته است. نخست هزینه ۶۶۸۷ مصرف‌کننده Azure ابری را بررسی کرده که این تعداد تقریباً ۲.۷ میلیون ماشین مجازی ایجاد کرده‌اند. تعدادی زیادی از این ماشین‌های مجازی کم‌بار یا پر بار برای منابع هستند. در گام دوم نقاط ضایعات فضای ابری (Cwp)^۲ را ارائه کرده است تا به صورت کمی، میزان ضایعات هر ماشین مجازی را ارزیابی کند. به علاوه ماشین‌های مجازی که منابع ابری را به صورت بهینه مصرف می‌کنند، ماشین‌های مجازی سبز و آن‌هایی که منابع ابری را هدر می‌دهند، قرمز نام‌گذاری کرده است.

در گام سوم، شاخص ضایعات ابری که (Cwi)^۳ نامیده شده است را برای دسته‌بندی مصرف‌کنندگان Azure به‌عنوان قرمز، سبز، یا نرمال بر اساس میزان مصرف بهینه آن‌ها از منابع ابری، استفاده کرده است. به علاوه نرخ استفاده از فضای ابری (Cus)^۴ را برای رتبه‌دهی به کارایی نسبی به مصرف‌کنندگان Azure با توجه به ضایعات ابری آن‌ها معرفی نموده و در نهایت الگوریتمی را برای مشخص کردن ماشین‌های مجازی قرمز ارائه و در آن آستانه را در نظر گرفته است. اما به دسته‌بندی ورودی‌ها اشاره‌ای نشده است.

در [۱۲]، تأثیر سخت‌افزار و نرم‌افزارهای مختلف را بر روی مصرف انرژی سیستم‌های کامپیوتری، لیست کردند. آن‌ها دریافتند که پردازنده بخشی است که بیش‌ترین تأثیر را بر مصرف انرژی در سیستم‌ها، بین همه اجزا دارد. همچنین نشان دادند که استفاده از پردازش ماشین فیزیکی و مصرف انرژی به صورت خطی نیست. آنها

در [۷] راهبردهای تخصیص حافظه مجازی را مطالعه کردند و در روش SDN^۱ ماشین‌های مجازی با توجه به تغییرات ram, cpu دیسک و پهنای باند، دسته بندی شدند. اما با افزایش ورودی‌های داده‌های نامتعادل، میزان مصرف انرژی و هزینه مورد توجه قرار نگرفته است.

در [۱۳] الگوریتمی از نوع ابتکاری به نام EPETS ارائه شده که به مبحث کاهش انرژی پرداخته است. در این الگوریتم دو مرحله زمان‌بندی وجود دارد که در مرحله اول بدون در نظر گرفتن انرژی، با توجه به زمان ضرب‌الاجل، زمان اجرای وظایف را کاهش می‌داد. در مرحله دوم، با در نظر گرفتن انرژی مصرفی و زمان ضرب‌الاجل، زمان‌بندی وظایف را انجام می‌داد. اما الگوریتم مورد مطالعه، وظایف ترتیبی که در جریان‌های کاری استفاده می‌شوند را استفاده نمی‌کرد و وظایف بدون ارتباط خاصی وارد سیستم می‌شدند.

در [۱۴] چارچوبی جهت تغییر پویا در سیستم‌های محاسباتی با جریان‌های ارتجاعی، ارائه شده است. در این روش، بررسی تغییر جهت دینامیک جریان‌های داده بلادرنگ، ارائه یک مدل برنامه کاربردی جریان عمومی و گروه‌بندی جریان داده، همچنین رسمی کردن مشکل بهینه‌سازی تعادل بار و تغییر جهت جریان داده در میان چندین نمونه از یک اپراتور در زمان اجرا، توسط یک استراتژی متعادل کردن وزن بالا برای بهبود تعادل بار یک مرکز داده در سطح دوبعدی در نظر گرفته شده که به طور قابل توجهی در بهبود عملکرد در تأخیر سیستم، توان عملیاتی و معیارهای متعادل‌کننده بار مؤثر بوده و تأثیر قابل توجهی در کاهش هزینه در فضای ابری داشته است. اما مهم‌ترین نکته‌ای که در فضای ابری وجود دارد بحث انرژی می‌باشد که در نظر گرفته نشده است.

رویکرد پیشنهادی در [۱۵]، به بررسی مشکل زمان‌بندی مجموعه‌ای از کارها در پردازش موازی پرداخته است که در آن ماشین‌ها دارای ظرفیت و قدرت پردازش متفاوت هستند، که هدف آن به حداقل رساندن زمان کل و میزان کل مصرف انرژی است که در آن وظایف دارای اندازه‌های متفاوت و زمان رسیدن دینامیک می‌باشند. در این روش برای پرداختن به مسئله بهینه‌سازی دو هدفه یک الگوریتم تکاملی سه جمعیتی انجام شده است. اما برای کاهش هزینه و افزایش توان عملیاتی و همچنین بهینه‌سازی

² Cloud Waste Points

³ Cloud Waste Indicator

⁴ Cloud Utilization Score

¹ Software-Defined Network

E_i : مجموعه یال‌های گراف که ارتباط بین وظایف و ترتیب اجرای آن‌ها را نشان می‌دهد.

D_i : زمان مهلت پایان کار^۳ کل گراف

B_i : مجموعه‌ای از گره‌های گراف که می‌توانند به صورت خوشه‌ای پردازش شوند.

در مقاله [۴] با توجه به تعداد بالای جریان‌های کاری نمونه - فشرده، تعداد کل جریان‌های کاری، مجموعه $I = \{I_1, I_2, \dots, I_i, \dots, I_\omega\}$ و ω که در آن تعداد کل جریان‌های کاری می‌باشد. $I_i = \{I_{i1}, I_{i2}, \dots, I_{ij}, \dots, I_{ijj}\}$ مجموعه جریان‌های کاری از نوع W_i است که I_i تعداد کل نمونه جریان‌های کاری W_i و I_{ij} ، i امین نمونه‌ی جریان کاری است.

پارامترهای شاخص استفاده شده در روش پیشنهادی در جدول ۲ نشان داده شده است.

جدول ۲- پارامترهای مورد استفاده در WSTMOS

مجموعه سرورهای مرکز داده	PM
مجموعه ماشین‌های مجازی	VMT
نرخ مصرف انرژی پایه برای ماشین P_m	Φ_m
درصد پردازنده ماشین P_m در زمان t	$U_m(t)$
نرخ مصرف انرژی ماشین P_m در زمان t	$PE_m(t)$
فاصله زمان اجرا	$T_{d'}$
میانگین زمان تخصیص منابع	$Avg(Alloc)$
زمان اجرای وظیفه	T_{exec}
زمان درخواست وظیفه	T_{req}
زمان تصدیق وظیفه	T_{ack}
میزان فاصله وظیفه	T_d
فاصله پیچیدگی رأس و اندازه یال	γ
احتمال اجرای وظیفه	$p(T_e)$
مجموعه وظایف جریان کاری i	RT_i
i امین میزان استفاده از cpu	Up_i
درصد استفاده از cpu در Up_i	Td

اطلاعات مربوط به استفاده از قدرت و کاربرد پردازنده را جمع‌آوری، و یک مدل مصرف توان بر اساس استفاده از پردازنده، و یک پیمانانه^۱ نرم‌افزاری مجازی را طراحی نمودند. در مطالعه مذکور، زمان اجرا در نظر گرفته شده است، در صورتی که اشاره‌ای به هزینه نشده است. مبحث انرژی و کاهش هزینه یکی از مسائل مهم در رایانش ابری می‌باشد که بیشتر الگوریتم‌ها آن را بهینه نموده‌اند اما روش پیشنهادی WSTMOS در مقایسه با روش‌های دیگر، پارامترهای فاصله و زمان اجرای واقعی را جهت این مهم استفاده کرده است. اما در روش‌های دیگر، این موارد به صورت هم‌زمان در نظر گرفته نشده‌اند و همچنین در روش ارائه شده با استفاده از دسته بندی ورودی‌ها، فاصله، زمان اجرای واقعی پارامترهای توان عملیاتی و تاخیر نیز، علاوه بر موارد مذکور، بهینه شده‌اند.

جدول ۱- مقایسه الگوریتم‌های مطالعه و ارائه شده

الگوریتم	انرژی	هزینه	توان عملیاتی	تأخیر	Load-balancing
ECIB	*	*			
Dr-Stream			*	*	*
WSTMOS	*	*	*	*	*

۳. مدل پیشنهادی

در این بخش جهت توصیف مدل ارائه شده، جریان کاری مورد مطالعه همراه با فرمول استفاده شده برای آن، پارامترهای مرکز داده ابری و همچنین مدل انرژی محاسبه شده را با جزئیات بررسی خواهیم کرد. در جدول ۲ پارامترهای استفاده شده در این تحقیق، بیان شده است.

۳.۱ رابطه جریان کاری

چندین مدل مورد بررسی قرار گرفته که در الگوریتم پیشنهادی، استفاده شده است.

هر جریان کاری در سیستم زمانبندی به صورت یک گراف بدون دور جهت‌دار^۲ نشان داده شده است:

$$W_i = (T_i, E_i, D_i, B_i) \quad (1)$$

i : اندیس جریان کاری

T_i : مجموعه وظایف در جریان کاری

¹ Module

² DAG (Directed Acyclic Graph)

³ Deadline

با مشاهده روش‌های بررسی شده و همچنین الگوریتم ECIB، مسئله مهمی که مورد مطالعه دقیق قرار نگرفته است، دسته‌بندی ورودی‌ها، فاصله و نیز زمان اجرای واقعی می‌باشد که در الگوریتم پیشنهادی، بهینه شده است.

طراحی سیستم به این صورت می‌باشد که زمان‌بند در نظر گرفته‌شده، از چند الگوریتم تشکیل شده است. الگوریتم‌های بکار رفته به ترتیب پردازش خوشه‌ای، زمان‌بندی منابع^۲، بررسی بار اضافی^۳ و الگوریتم کاهش منابع^۴ می‌باشد.

۴. الگوریتم ارائه شده WSTMOS

فرایند کلی به این صورت است که در ابتدا با توجه به وظایفی که وارد مرکز داده^۴ می‌شوند، در ابتدای جریان کاری، برای وظیفه موردنظر، زمان پایان مهلت فرعی، محاسبه می‌شود و ماشین‌های مجازی فعال، بر اساس قیمتشان از ارزان‌تر به گران‌تر، مرتب می‌شوند. چنانچه زمان اجرای محاسبه شده، از زمان پایان مهلت فرعی تجاوز نکند، این ماشین به وظیفه موردنظر اختصاص داده می‌شود. در غیراینصورت، باید الگوریتم ScaleUp صدا زده شود. یکی از مسائل زمان‌بندی، انرژی و هزینه در جریان کاری ابر، ماشین‌های مجازی می‌باشد. در این مقاله با کاهش انرژی، هزینه و همچنین وجود آستانه، نسبت به روش‌های مطالعه شده، آن را بهبود داده‌ایم. مراحل الگوریتم در شکل ۱ و فلوچارت در شکل ۲ نمایش داده شده است. در ابتدا الگوریتم WSTMOS که الگوریتم اصلی می‌باشد، چهار زیر الگوریتم BatchProcessing، ResourceScheduling، HandlingOverload و ScaleDownResource را صدا می‌زند.

الگوریتم ۱: WSTMOS

Input: I: Set of workflow instances
 UT: CPU usage upper threshold
 LT: CPU usage lower threshold
 IT: The idle time threshold
 PM: Set of PMs
Output: Resource scheduling policy
 1: for each PM $I_i \in I$
 2: RT_i : BatchProcessing(I_i)
 3: ResourceScheduling(I_i)
 4: HandlingOverload(PM, UT)
 5. ScaleDownResource(PM, LT, UT, IT)
 6. end for

² Resource scheduling

³ Handling overload

⁴ Scale Down resources

برای بهینه‌سازی توان عملیاتی و تأخیر جریان‌های کاری که به صورت تصادفی ارائه شده است، پارامتر اندازه پیچیدگی رأس و یال مورد استفاده واقع شده است.

۳.۲ مدل مرکز داده ابری

مرکز داده دارای M ماشین فیزیکی^۱ و N ماشین مجازی می‌باشد که در زیر نمایش داده شده است [۴]:

$$PM = \{p_1, p_2, \dots, p_m, \dots, p_M\} \quad (۲)$$

$$VMT = \{vmt_1, vmt_2, \dots, vmt_n, \dots, vmt_N\} \quad (۳)$$

۳.۳ مدل انرژی

$$PE_m(t) = \begin{cases} \varphi_m & \text{if } U_m(t) = 0 \\ \varphi_m + \alpha_m & \text{if } 0 < U_m(t) \leq 0.2 \\ \varphi_m + 3\alpha_m & \text{if } 0.2 < U_m(t) \leq 0.5 \\ \varphi_m + 5\alpha_m & \text{if } 0.5 < U_m(t) \leq 0.7 \\ \varphi_m + 8\alpha_m & \text{if } 0.7 < U_m(t) \leq 0.8 \\ \varphi_m + 11\alpha_m & \text{if } 0.8 < U_m(t) \leq 0.9 \\ \varphi_m + 12\alpha_m & \text{if } 0.9 < U_m(t) \leq 1 \end{cases} \quad (۴)$$

که در آن φ_m نرخ مصرف انرژی پایه برای p_m ، α_m ثابتی است که به صورت $\alpha_m = \frac{\varphi_m}{7}$ محاسبه می‌شود و $U_m(t)$ نیز مصرف پردازنده ماشین p_m در زمان t می‌باشد.

$U_m(t)$ با استفاده از رابطه زیر محاسبه می‌شود:

$$U_m(t) = \sum_{n=1}^N r_{m,n}(t) \times vcp_n / cp_m \quad (۵)$$

کل انرژی ماشین p_m در کل زمان، از رابطه زیر به دست می‌آید:

$$E_m = \int_{t \in \omega_m} PE_m(t) dt \quad (۶)$$

که ω_m در آن کل زمان عملکرد p_m است [۴].

پردازش خوشه‌ای وظایف نیز، به تفصیل در [۸]، [۹] و [۱۰] توضیح داده شده است. الگوریتم به کار گرفته شده در این مقاله، الهام گرفته از الگوریتم (ECIB) [۴] می‌باشد که با ادغام نمودن پارامترها و محاسبه زمان واقعی اجرا و آستانه آشکارساز، انرژی را بهینه نموده است. پارامترهای توان عملیاتی و تأخیر در ECIB، در نظر گرفته نشده‌اند.

¹ Physical machine (Host)

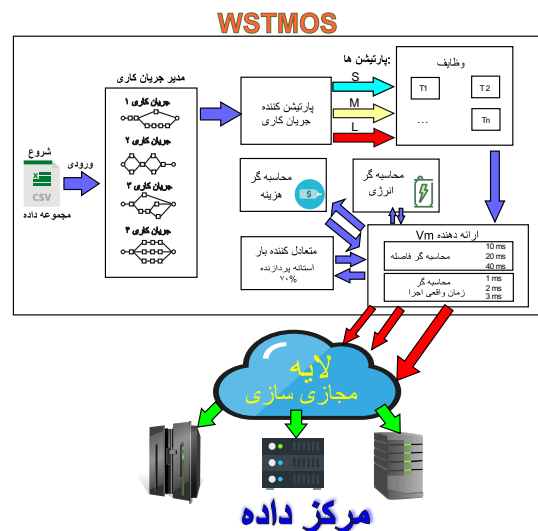
کاری مذکور هست، آن‌را به دسته‌های کوچک، متوسط و بزرگ تقسیم کرده و وظایف دسته‌بندی شده به ارائه‌دهنده ماشین مجازی ارسال می‌شوند تا جهت دریافت منبع مناسب، با توجه به محاسبه‌گر فاصله و زمان واقعی اجرا، به آنها تحویل داده شود. در هر گام زمانی در فرایند زمانبندی، محاسبه‌گرهای انرژی و هزینه و همچنین متعادل‌کننده بار، مسئولیت مدیریت منابع و همچنین خاموش یا روشن نمودن ماشین‌های فیزیکی را با توجه به شرایط موجود دارند. در طبقه بندی داده، در صورتی که low سرریز شود، مهاجرت بر روی mid و high انجام می‌شود. سه وضعیت low, high و mid وجود دارد که بین ۰ تا ۳۰٪، ۳۰ تا ۶۰٪ و بالاتر از ۶۰٪ می‌باشد. زمانی که overload در low انجام می‌شود، می‌تواند به mid و همچنین از mid به high با زیر الگوریتم روش پیشنهادی انجام شود.

زمانی که پردازش دسته‌ای روی محدوده تعریف شده است، می‌توان آستانه تخمین زده شده توسط تابع صلاحیت استفاده نمود و برای اختصاص دسته‌های وظایف به منابع در نظر گرفت. با این شرایط تصمیم‌گیری، هزینه و انرژی نسبت به روش مورد مطالعه کاهش داده شده است و این کار توانسته اتلاف انرژی و هزینه کمتری داشته باشد.

در شکل ۲، ابتدا گره‌های ورودی دریافت و بر اساس زمان ورود و زمان پایان مهلت برای وظیفه، مرتب می‌شوند.

همان‌طور که در شکل ۱ مشاهده می‌شود، بعد از ارسال لیست جریان‌های کاری، مدیر جریان کار، لیست را به تقسیم‌کننده جریان کار می‌فرستد و تقسیم‌کننده، وظایف را به دسته‌های کوچک، متوسط و بزرگ قسمت می‌کند و برحسب نیاز منبع که کار قبلی خود را به اتمام رسانده باشد، با وجود آستانه مشخص می‌شود که در کدام دسته کوچک، متوسط، و بزرگ قرار گرفته است. مفهوم دسته‌بندی با توجه به شکل ۱ می‌تواند کارها را با تخصیص درست منابع وظایف در دسته‌بندی فوق قرار دهد و در این حالت انرژی و هزینه نسبت به روش‌های مورد مطالعه با این راهکار بهینه می‌شود و با توجه به حجم کاری و زمان ورود، خوشه‌بندی می‌کند. این فرایند دسته‌بندی در الگوریتم ۲ که BatchProcessing نام دارد، انجام می‌شود.

در ادامه، الگوریتم ResourceScheduling، زمان اجرا و فاصله را محاسبه کرده، و ماشین مناسب را در اختیار وظیفه قرار می‌دهد. هنگام کار سیستم، انرژی و هزینه مصرف‌شده در مرکز داده، محاسبه و ذخیره می‌شود. همچنین متعادل‌کننده بار^۱ با توجه به مقدار آستانه، مهاجرت ماشین مجازی و خاموش یا روشن کردن سیستم‌ها را بر عهده دارد.



شکل ۱ - معماری سیستم WSTMOS

در شکل ۱ شمای کلی از الگوریتم WSTMOS را مشاهده می‌کنیم. در ابتدا مجموعه جریان‌های کاری، با توجه به نوع ۱ تا ۴ وارد سیستم شده و مدیر جریان کاری، بعد از تشخیص نوع آن، جریان‌های کاری با به پارتیشن‌کننده تحویل می‌دهد و بسته به نوع درصدی که از که نیاز به استفاده از پردازنده توسط جریان

الگوریتم ۲: BatchProcessing

Input: I_i : Set of workflow instance of W_i

Output: RT_i : Set of activity instances ready for Scheduling of W_i

- 1: for each $I_{ij} \in I_i$
- 2: while T_{ijk} is ready
- 3: If T_{ijk} is a batch – processing activity instance
 T_{ijk} to Waiting Queue
- 4: add
- 5: else
- 6: add T_{ijk} to RT_i
- 7: end if
- 8: end while
- 9: end for

¹ Load Balancer

به منبع اختصاص یابد، زمان واقعی درصدی از زمان اتمام شده است که در روابط زیر محاسبه شده است:

$$i = (Up_1, Up_2, \dots, Up_n) \quad (7)$$

$$T_{real} = T_{exec} - (T_{req} + T_{ack}) \quad (8)$$

$$\alpha = \frac{TD \times Up_1}{\sum_{i=1}^n Up_i} \quad (9)$$

$$\beta = \frac{\alpha_i + (TD \times Up_{i+1})}{2} \quad (10)$$

$$T_{ack} = \alpha T_{exec} \quad (11)$$

$$T_{req} = \beta T_{exec} \quad (12)$$

پارامترهای α و β در شبیه‌سازی استفاده شده‌اند. از این رابطه در خط ۷ الگوریتم ۳، وقتی که می‌خواهیم زمان اتمام کار وظیفه را بر روی ماشین مجازی محاسبه کنیم، استفاده کرده‌ایم.

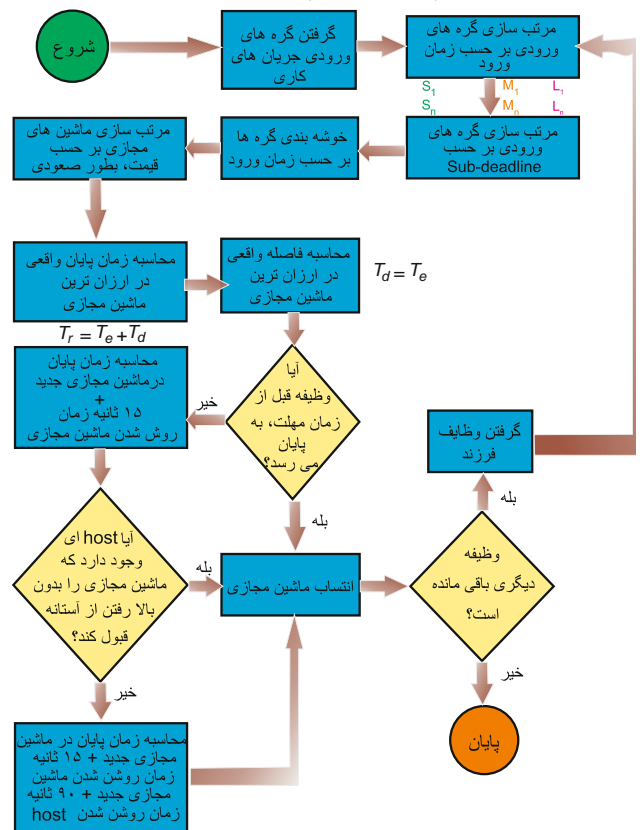
الگوریتم ۳: ResouceSceduling

Input: RT_i : Set of activity instances of W_i ready for scheduling
 APM: Set of active PMs

Output: Resouce scheduling policy

- 1: **for each** activity instance rt_{ia} in RT_i
- 2: calculate the sub-deadline $SD(rt_{ia})$
- 3: sort VMs by ascending order of price
- 4: sort activities by ascending arrival time and sub-deadline
- 5: group activities by arrival time
- 6: **for each** VM
- 7: finish time = $T_e + \gamma T_e - (\alpha + \beta) T_e$
- 8: $T_p = \frac{T_{d'} \times T_r}{Avg(Alloc)}$
- 9: **if** finish time < $SD(rt_{ia})$
- 10: assign rt_{ia} to that vm
- 11: FindFlag=True
- 12: **end if**
- 13: **end for**
- 14: **if** FindFlag==False
- 15: ScaleUp(rt_{ia})
- 16: **end if**
- 17: **end for**

فلوچارت الگوریتم WSTMOS



شکل ۲ - فلوچارت الگوریتم WSTMOS

سپس گره‌های خوشه‌بندی‌شده، و ماشین‌های مجازی موجود در مرکز داده، به صورت صعودی نسبت به پارامتر هزینه ماشین در ساعت، مرتب می‌شوند. بعد از محاسبه فاصله و زمان واقعی اجرای وظیفه بر روی ارزان‌ترین ماشین، چنانچه قبل از زمان پایان مهلت، انجام شود، وظیفه به ماشین انتقال داده می‌شود، در غیر این صورت، زمان، جهت ایجاد ماشین مجازی جدید با در نظر گرفتن ۱۵ ثانیه زمان روشن شدن، محاسبه می‌شود. چنانچه سروری وجود نداشته باشد که ماشین مجازی جدید در آن انتقال داده شود، الگوریتم ScaleUp صدا زده شده و زمان روشن شدن سرور که ۹۰ ثانیه هست، در نظر گرفته می‌شود. بعد از انتساب وظیفه به ماشین مورد نظر، چنانچه وظیفه دیگری بود، وظایف فرزند به ماشین داده می‌شوند، در غیر این صورت، وظیفه‌ای در صف نیست و شبیه‌سازی به اتمام می‌رسد.

۴.۱ زمان اجرای واقعی

در الگوریتم زمان‌بندی، هنگامی که منبع مورد نیاز استفاده و همچنین وظیفه نیز تخصیص داده شد، زمان واقعی اجرا محاسبه می‌شود. با توجه به مدتی زمان کوتاهی که طول می‌کشد تا وظیفه

تعداد بالایی است، نرخ ورود را دسته‌بندی می‌کنیم و هنگامی که کل وظایف در زمان موردنظر رسیدند، ماشین مجازی اختصاص داده شده است.

۵ شبیه‌سازی

جهت شبیه‌سازی از نرم‌افزار Cloudsim 3.0.3 استفاده شده است. برای شروع شبیه‌سازی پارامترهای مورد استفاده در جدول ۲ نشان داده شده است.

۳ نوع ماشین فیزیکی برای مرکز داده در نظر گرفته شده است که نوع آن‌ها و مصرف پایه‌ای انرژی‌شان در حالت سکون در جدول ۲ آمده است. در جدول ۳ نیز انواع ماشین‌های مجازی که در شبیه‌سازی استفاده شده‌اند، به همراه قیمت آن‌ها در هر ساعت، نشان داده شده است. توجه داشته باشید که چنانچه یک ماشین مجازی کمتر از یک ساعت روشن باشد، همان هزینه یک ساعت محاسبه می‌شود. با توجه به اینکه جریان‌های کاری نمونه - فشرده، ساختار ساده‌تری نسبت به جریان‌های کاری علمی دارند و وظایف آن‌ها ساده‌تر است، ۴ نوع جریان کاری برای آزمایش در نظر گرفته شده است که در شکل‌های ۳ (الف) تا (د) مشاهده می‌کنیم.

جدول ۳-انواع ماشین‌های فیزیکی

سرعت هسته (برحسب Mips)	مصرف پایه‌ای انرژی در حالت سکون (برحسب وات)	نام ماشین فیزیکی
1860	86w	HP ProLiant ML110 G4 (Intel Xeon 3040, dual-Processor clocked at 1860 MHz, 4 GB of RAM)
2660	93.7w	HP ProLiant ML110 G5 (Intel Xeon 3075, dual-Processor clocked at 2660 MHz, 4 GB of RAM)
3060	192w	HP ProLiant SL390s G7 (Intel Xeon 5649, dual-Processor clocked at 3060 MHz, 16 GB of RAM)

Input: rt_{ia} : The activity should be executed
Output: Resource scheduling policy
 1: select pm in PMs with minimum resource usage
 2: find Vm with minimum MIPS which can execute activity in time lower than sub-deadline, with notice to power on vm in 15s
 3: **if** vm can be created on pm
 4: create vm on pm and execute rt_{ia} on vm
 5: **else**
 6: power on a new pm which can have that vm with pm power on time 90s + vm power on time 15s and execute rt_{ia} on vm
 7: **end if**

۴،۲ فاصله

برای انتقال وظیفه به ماشین مجازی اختصاص یافته، پارامتر جدید فاصله را ایجاد کرده‌ایم که روابط آن به صورت زیر است:

$$T_r = T_e + T_d \quad (۱۳)$$

$$T_d = T_e + \gamma T_e \quad (۱۴)$$

$$\gamma = \frac{\alpha + \beta}{2} \quad (۱۵)$$

$$T_{d'} = p(T_e) + (1 - \gamma p(T_e))T_e \quad (۱۶)$$

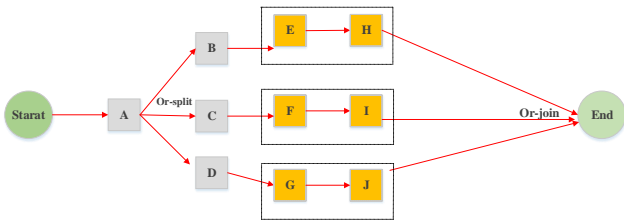
$$T_p = \frac{T_{d'} \times T_r}{Avg(Alloc)} \quad (۱۷)$$

پارامتر γ در شبیه‌سازی استفاده شده است. این رابطه نیز در خط ۷ الگوریتم ۳ استفاده شده است. $Avg(Alloc)$ نیز میانگین زمان تخصیص منابع، که شامل فاصله پیچیدگی راس، فاصله اندازه یال و همچنین دسته‌بندی داده‌ها در زمانبندی جریان کاری در نظر گرفته شده است.

با مشخص شدن پارامتر γ مقادیر T_d و سپس T_r نشان داده می‌شود و مقادیر α و β نیز با آشکارساز آستانه و همچنین زمان واقعی مشخص شده است.

۴،۳ دسته‌بندی برحسب زمان ورود

در هنگام زمانبندی، وقتی که می‌خواهیم وظیفه‌ای را به منبعی منتسب کنیم، به جای آنکه در همان لحظه ورود این کار را انجام دهیم، با توجه به اینکه نرخ ورود وظایف در بازه زمانی کوتاه و



شکل ۳- جریان کاری (د)

وظایف بنفش رنگ که دور آن‌ها خطچین کشیده شده، به این معنی است که می‌توانند به صورت خوشه‌ای پردازش شوند.

۲،۵ مجموعه داده^۱ استفاده شده

بار کاری برای هر وظیفه، عددی تصادفی بین ۲۰۰ تا ۴۰۰ Mips و زمان پایان مهلت برای هر جریان کاری، زمانی تصادفی بین ۳۰۰ تا ۳۶۰ ثانیه در نظر گرفته شده است. همچنین زمان ورود هر وظیفه، زمانی تصادفی بین ۰ تا ۱۰ ثانیه می‌باشد.

فرمت مجموعه داده‌ها، از نوع فایل‌های CSV^۲ است. ساختار به این شکل است که داده‌های هر رکورد، در یک سطر قرار دارند، و فیلدها با کاراکتر “,” از هم جدا می‌شوند. با زدن Enter، ستون جدیدی در نظر گرفته می‌شود.

یک نمونه مجموعه داده استفاده شده به صورت (0.399, false, 30, 0, 101, A) است. این سطر ۹ فیلد دارد، ۰ شماره‌ی رکورد، ۳۹۹ بار کاری لازم برای وظیفه، false فیلدی از نوع bool که نشان می‌دهد وظیفه نمی‌تواند به صورت خوشه‌ای پردازش شود، ۳۰، GroupingChar است که قبلاً استفاده می‌شده و برای مقالات قبلی بوده، ۰ زمان ورود، ۱۰۱ زمان پایان مهلت برای وظیفه (همان Sub-Deadline است که در هنگام ایجاد جریان‌های کاری تصادفی با توجه به زمان پایان مهلت کلی، محاسبه می‌شود) و A نام وظیفه در جریان کاری است. در جدول ۵، توضیحات مربوط به داده‌های فایل ورودی، آمده است.

جدول ۵- توضیحات مجموعه داده

نام داده	توضیحات
Index	ایندکس وظیفه
Task Length	اندازه بار کاری وظیفه
Batch	آیا وظیفه به صورت خوشه‌ای اجرا می‌شود
GroupingChar	نوع خوشه‌بندی
Arrival Time	زمان ورود
Sub-deadline	زمان مهلت پایان محاسبه‌شده
Task Name	نام وظیفه

¹ Dataset

² Comma-separated values

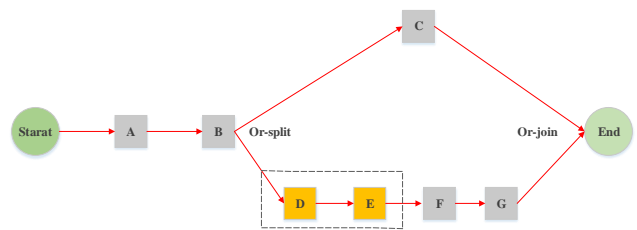
جدول ۴-انواع ماشین‌های مجازی

نوع ماشین	ظرفیت پردازش (Mips)	قیمت (\$)
m4.large	۱۰۰۰	۰.۱
m4.xlarge	۲۰۰۰	۰.۲
m4.2xlarge	۴۰۰۰	۰.۴

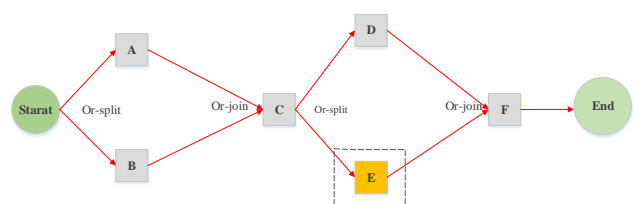
خطوطی که به صورت خطچین به دور بعضی از وظایف کشیده شده است، نشان‌دهنده این است که می‌توان آن‌ها را همراه با وظایف هم‌نامشان در جریان‌های کاری دیگر، خوشه‌ای پردازش کرد.

۱،۵ جریان‌های کاری استفاده شده

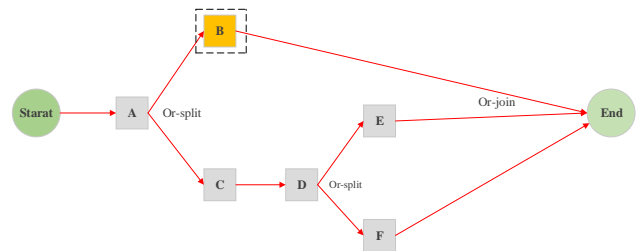
در شبیه‌سازی، ۴ جریان کاری انتخاب شده است که در شکل‌های ۳ (الف) تا (د) نشان داده شده است.



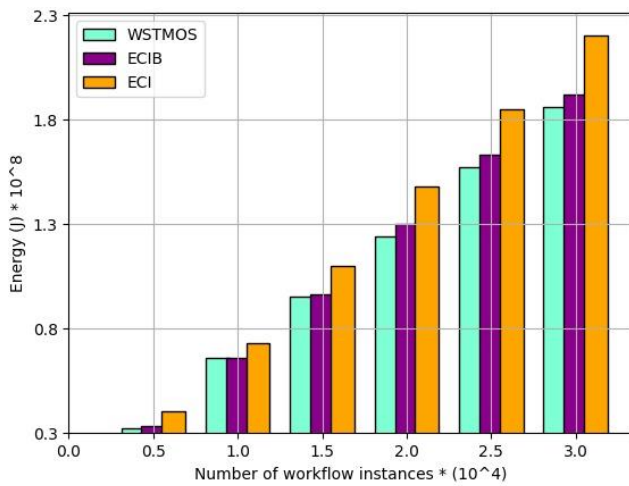
شکل ۳- جریان کاری (الف)



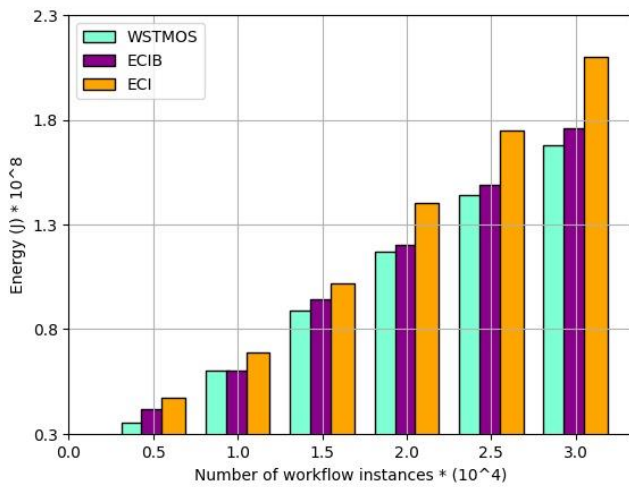
شکل ۳- جریان کاری (ب)



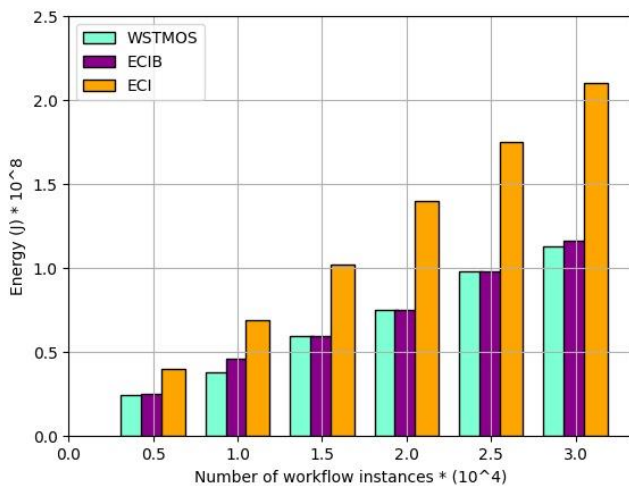
شکل ۳- جریان کاری (ج)



شکل ۴- نمودار انرژی جریان کاری (ب)



شکل ۴- نمودار انرژی جریان کاری (ج)



شکل ۴- نمودار انرژی جریان کاری (د)

۵.۴ نمودارهای هزینه و بررسی نتایج حاصل

در شکل‌های ۵، (الف) تا (د) نیز، تعداد ورودی‌ها مطابق با نمودارهای انرژی می‌باشد. در نمودار جریان کاری (الف)، در تمام

۳.۵ نمودارهای انرژی و بررسی نتایج حاصل

همان‌طور که در نمودارهای انرژی با توجه به ۴ جریان کاری که در نظر گرفته شد، ۴ نمودار انرژی به دست آمد که الگوریتم پیشنهادی WSTMOS با الگوریتم ECIB و ECI مقایسه شده است.

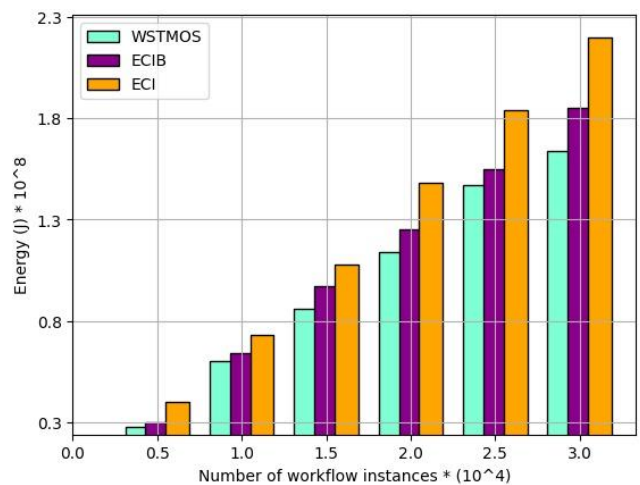
ECI همان الگوریتم ECIB هست، فقط پردازش خوشه‌ای در آن انجام نشده است.

برای ECIB و WSTMOS نیز، اندازه خوشه^۱، ۲۰ وظیفه در نظر گرفته شده است. تعداد جریان‌های کاری ورودی در هر شبیه‌سازی از ۵۰۰۰ تا ۳۰۰۰۰ بوده و در هر مرحله، ۵۰۰۰ تا جریان کار جدید اضافه شده است. در نتیجه برای هر جریان کاری، ۶ شبیه‌سازی انجام شده است. شکل‌های ۴ (الف) تا (د) نشان داده شده است، برای جریان کاری (الف) انرژی به وضوح کاهش یافته است.

برای جریان کاری (ب)، تا وقتی اندازه تعداد ورودی‌ها ۲۵۰۰۰ تا است، انرژی نزدیک است، ولی در ۳۰۰۰۰، کاهش یافته است.

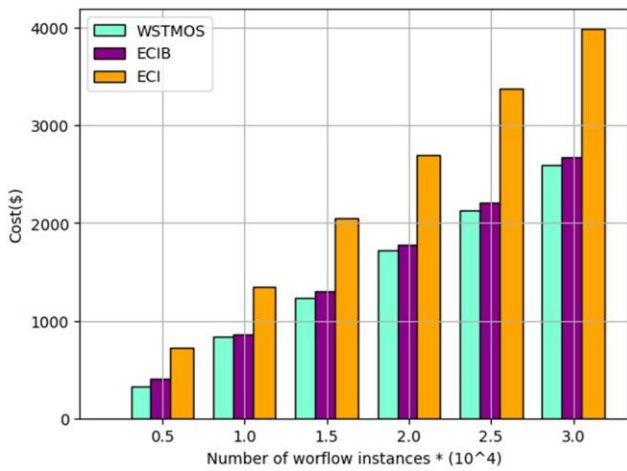
در جریان کاری (ج)، انرژی برای تعداد ورودی جریان کاری تا ۲۵۰۰۰ نزدیک می‌باشد، اما از ۳۰۰۰۰ به بعد، انرژی کاهش یافته است.

در جریان کاری (د) نیز، انرژی در اکثر تعداد ورودی‌ها، متناسب با الگوریتم ECIB است، ولی در تعداد ۱۰۰۰۰ و ۳۰۰۰۰ انرژی کاهش یافته است.



شکل ۴- نمودار انرژی جریان کاری (الف)

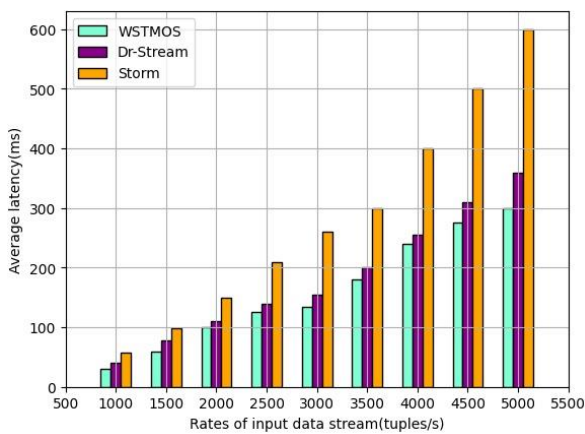
¹ BatchSize



شکل ۵- نمودار هزینه جریان کاری (د)

۵،۵ نمودارهای مقایسه با الگوریتم Dr-Stream

محور X نشان داده نرخ ورود داده‌هاست و محور Y نشان‌دهنده میانگین تأخیر یا همان Average Latency می‌باشد.

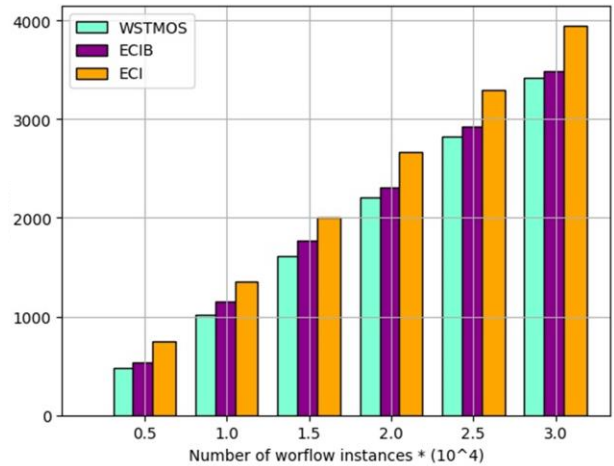


شکل ۶- میانگین تأخیر

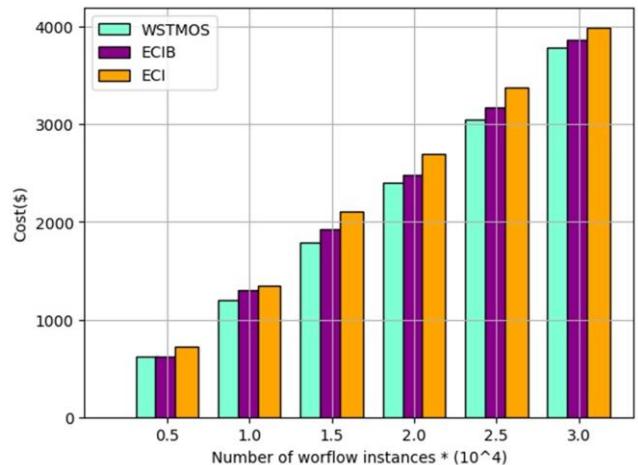
در ابتدا با نرخ ورود داده ۱۰۰۰، تأخیر در Storm، ۵۰ میلی‌ثانیه و در Dr-stream، ۴۰ میلی‌ثانیه و در WSTMOS، ۳۰ میلی‌ثانیه بود. برای تمام ورودی‌ها، WSTMOS میانگین تأخیر ۱۶۰ و برای Dr-stream ۱۸۳.۱۱ می‌باشد که WSTMOS نسبت به Dr-stream، ۱۲٪ کاهش یافته است. در ادامه به بررسی میزان توان پردازشی^۱ پرداخته‌ایم.

موارد کمی، کمتر از ECIB می‌باشد، ولی در ورودی ۱۵۰۰۰ تا هزینه بیشتر کاهش یافته است.

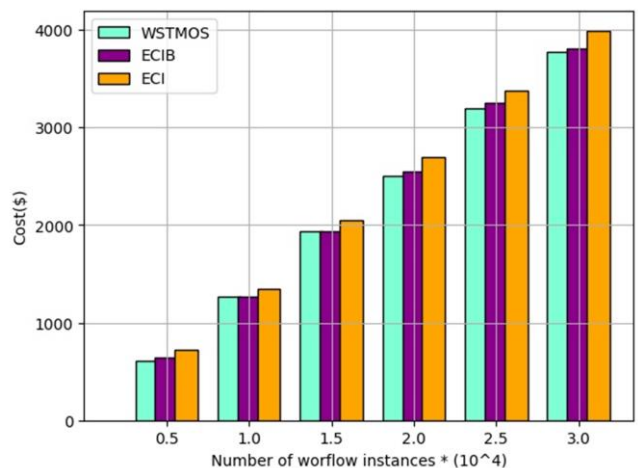
برای جریان کاری (ب)، هزینه در تمام موارد نزدیک به ECIB است، فقط در تعداد ۳۰۰۰۰ تا، کمی کاهش یافته است. برای جریان کاری (ج)، هزینه در ورودی ۲۰۰۰۰ تا، کمی کاهش یافته است و برای جریان کاری (د) نیز هزینه تقریباً برابر است.



شکل ۵- نمودار هزینه جریان کاری (الف)



شکل ۵- نمودار هزینه جریان کاری (ب)



شکل ۵- نمودار انرژی جریان کاری (ج)

¹ Throughput

جدول ۶- میانگین بهینه شدن پارامترها نسبت

به الگوریتم Dr-stream

میانگین بار کاری	میانگین توان	میانگین تأخیر	الگوریتم
٪۲.۴	۱۶۶۹.۴۴	۱۶۰	WSTMOS
٪۱.۵	۱۵۸۰	۱۸۳.۱۱	Dr-Stream

الگوریتم‌های ECIB و ECI را به‌عنوان معیار^۲ در نظر گرفتیم. با توجه به آزمایش‌های انجام شده، میزان انرژی به طور میانگین ۴.۸ درصد و هزینه ۴.۴ درصد، کاهش یافت.

همان‌طور که در جدول‌های ۷ و ۸ نشان داده شده است، نتایج آزمایش‌های با نتایج الگوریتم معیار، مقایسه شده و در اکثر موارد، انرژی و هزینه کاهش یافته است. برای نمونه، میانگین درصد کاهش انرژی برای جریان کاری (الف) تا ۱۱.۲ درصد کاهش داشته که بیشترین مقدار بود. کمترین مقدار نیز برای جریان کاری (ب) است که ۲.۱ درصد بوده است. اما در همه موارد میانگین انرژی کاهش یافت.

در مورد هزینه نیز، میانگین نسبت به میانگین انرژی کاهش کمتری داشته، و بیشترین کاهش هزینه مربوط به جریان کاری (الف) است که ۶.۳ درصد بود.

جدول ۷- میانگین درصد کاهش انرژی در

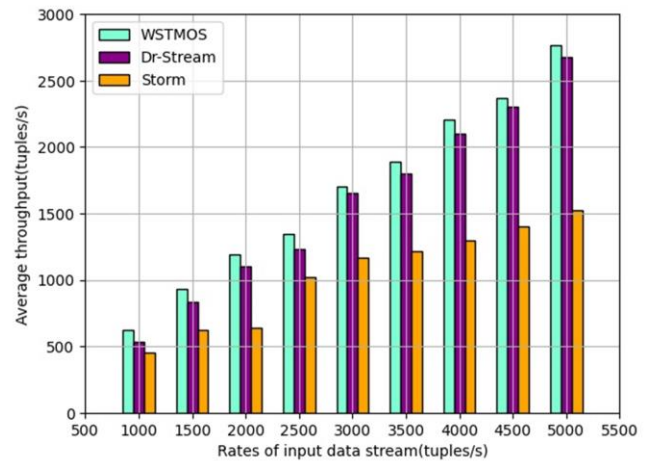
مقایسه با ECIB و Dr-Stream

جریان کاری	میانگین درصد کاهش
الف	۱۱.۲
ب	۲.۱
ج	۲.۵
د	۳.۴

جدول ۸- میانگین درصد کاهش هزینه در

مقایسه با ECIB و Dr-Stream

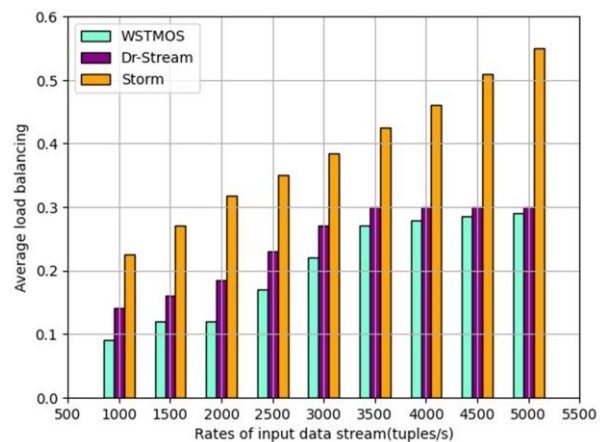
جریان کاری	میانگین درصد کاهش
الف	۶.۳
ب	۵.۸
ج	۲.۱
د	۳.۷



شکل ۷- میانگین توان مصرفی

در شکل ۷ میزان میانگین Throughput برای سه الگوریتم Storm، Dr-stream و WSTMOS آمده است. میانگین کل توان پردازشی تمام ورودی‌ها برای WSTMOS و Dr-stream به ترتیب، ۱۶۶۹.۴۴ و ۱۵۸۰ می‌باشد که WSTMOS نسبت به Dr-stream ۵٪ و ۱۶٪ بهینه شده است.

در شکل ۸ میانگین Load Balancing نشان داده شده است. در این نمودار، محور عمودی نشان‌دهنده میانگین لود Cpu در مرکز داده است، که هرچه قدر پایین‌تر باشد، بهتر است.



شکل ۸- میانگین متعادل سازی بار

میزان میانگین کل ورودی‌های متعادل‌کننده^۱، برای الگوریتم‌های WSTMOS و Dr-stream به ترتیب برابر با ۲٪ و ۲.۴٪ می‌باشد که نسبت کاهش WSTMOS به Dr-stream ۱۵٪ می‌باشد. مشاهده می‌کنیم که با افزایش نرخ ورود، میزان Load Balancing بسیار به الگوریتم Dr-stream میل می‌کند. در جدول ۶ میانگین بهینه شدن تأخیر، توان مصرفی و متعادل سازی بار نسبت به الگوریتم Dr-stream آمده است.

² Benchmark

¹ Load-Balancing

۶ نتیجه گیری و کارهای آتی

با مواردی که انجام شد، مصرف انرژی را در مراکز داده بررسی کردیم. همان طور که اشاره شد، میزان انرژی استفاده شده در مراکز داده، از مهم ترین عامل ها می باشد که بر محیط زیست تأثیر مهمی دارد. الگوریتم WSTMOS را معرفی نمودیم که الهام گرفته از الگوریتم ECIB بوده است.

در روش ارائه شده، قابلیت زمان بندی بر اساس دسته بندی نرخ ورود داده ها به الگوریتم اضافه شد، همچنین محاسبه زمان واقعی اجرای وظیفه در جریان کاری و پارامتر فاصله در این محاسبه، را در نظر گرفتیم. برای آزمایش ها نیز از نرم افزار Cloudsim نسخه ۳.۰.۳ استفاده شده است و به بررسی انرژی مصرفی و هزینه در ۴ جریان کاری انتخابی پرداخته شد.

با توجه به دسته بندی انجام شده بر اساس داده های ورودی، در آینده این روش را می توان با گردآوری نرخ ورود و اجرای الگوریتم های هوش مصنوعی، از جمله شبکه های عصبی^۱ یا الگوریتم های خوشه بندی^۲، برای درصد بالایی از ورودی ها بهینه نمود.

مراجع

- [1] م. نیک سرشت، م. راجی، "یک الگوریتم زمان بندی وظیفه چندهدفه بر اساس الگوریتم ژنتیک برای طراحی سیستم های نهفته"، دوفصلنامه علمی فناوری اطلاعات و ارتباطات ایران، شماره ۴۷ و ۴۸، صفحه ۱۸۶-۱۹۷، بهار و تابستان ۱۴۰۰
- [2] A. Ghorbannia Delavar and . Y. Aryan, "HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems." *Cluster Comput* (2014) 17, p. 129–137, 2013.
- [3] <https://github.com/Azure/AzurePublicDataset/blob/master/AzureFunctionsDataset2019.md>
- [4] W. Dou, X. Xu, . S. Meng, . X. Zhang, . C. Hu, . S. Yu and . J. Yang, "An energy-aware virtual machine scheduling method for service QoS enhancement in clouds over big data." *Concurr. Comput. Pract. p. Exp*, vol. 29 e3909, 2016.
- [5] N. J. Kansal and I. Chana, "Energy-aware Virtual Machine Migration for Cloud Computing. A Firefly Optimization Approach." *Grid Computing*, vol. 14, pp. 327-345, 2016.
- [6] A. Mosa and N. W. Paton, "Optimizing virtual machine placement for energy and SLA in clouds using utility functions." *Journal of Cloud Computing*, vol. 5, no. 1, 2016.
- [7] G. Portaluri, D. Adami, A. Gabbrielli and S. Giordano, "Power Consumption-Aware Virtual Machine Placement in Cloud Data Center." *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 4, pp. 541-550, 2017.
- [8] L. Pufahl, "Modeling and Executing Batch Activities in Business Processes (Ph.D. thesis)." 2018.
- [9] L. Pufahl and D. Karastoyanova, "Enhancing Business Process Flexibility by Flexible Batch Processing." pp. 426-444, 2018.
- [10] L. Pufahl, "Modeling and executing batch activities in business processes." 2018.
- [11] V. Seethalakshmi, V. Govindasamy, V. Akila, G. Sivaranjini, K. Sindhuja and K. Prasanth, "A Survey Of Different Workflow Scheduling Algorithms In Cloud Computing." 2019.
- [12] H. C. Y. T. S. Y. Xiaojun Ruan, "Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds." *Future Generation Computer Systems*, p. <https://doi.org/10.1016/j.future.2019.05.036>, 2019.

¹ Neural network

² Clustering

- [13] M. Hussain, L.-F. Wei, A. Lakhan, S. Wali, S. Ali and A. Hussaina, "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing." *Sustainable Computing: Informatics and Systems*, vol. 30, 2021.
- [14] D. Sun, S. Gao, L. Xunyun, Y. Xindong and B. Rajkumar, "Dynamic redirection of real-time data streams for elastic stream." *Future Generation Computer Systems*, vol. 112, p. 193–208, 2020.
- [15] Y. Wang, Z.-h. Jia and K. Li, "A multi-objective co-evolutionary algorithm of scheduling on parallel non-identical batch machines." *Expert Systems With Applications*, no. <https://doi.org/10.1016/j.eswa.2020.114145>, 2020.
- [16] S. Taherizadeh and M. Grobelnik, "Key influencing factors of the Kubernetes auto-scaler for computing-intensive microservice-native cloud-based applications." *Advances in Engineering Software*, vol. 140, 2020.
- [17] B. Everman, M. Gao and Z. Zong, "Evaluating and reducing cloud waste and cost. A data-driven case study from Azure workloads." *Sustainable Computing: Informatics and Systems*, vol. 35, 2022.
- [18] Yiping Wen, Zhibin Wang, Yu Zhang, Jianxun Liu, Buqing Cao and Jinjun Chen, "Energy and cost aware scheduling with batch processing for instance-intensive IoT workflows in clouds." vol. 101, pp. 39-50, 2019.
- [19] R. medara and R. S. Singh, "A Review on Energy-Aware Scheduling Techniques for Workflows in IaaS Clouds." *Wireless Personal Communications*, vol. 116, 2022.