

دریافت مقاله: ۱۴۰۲/۰۱/۱۸

پذیرش مقاله: ۱۴۰۲/۰۴/۱۳

نوع مقاله: پژوهشی

زمان بندی بهینه در محیط ترکیبی رایانش مه و ابر با استفاده از تئوری بار بخش پذیر

سید مجتبی کاظمی

دانشجوی دکتری کامپیوتر، دانشکده مهندسی، دانشگاه آزاد اسلامی واحد قم، قم، ایران
پست الکترونیکی: mojkaz@gmail.com

شمس اله قنبری*

استادیار دانشکده مهندسی، دانشگاه آزاد اسلامی واحد آشتیان، آشتیان، ایران
پست الکترونیکی: myrshg@gmail.com

منوچهر کاظمی

دانشیار دانشکده علوم، دانشگاه آزاد اسلامی واحد آشتیان، آشتیان، ایران
پست الکترونیکی: Univer_ka@yahoo.com

محمد عثمان

استاد دانشکده فناوری ارتباطات و شبکه، دانشگاه پوترا مالزی (UPM)، کوالالمپور، مالزی
پست الکترونیکی: mothman@upm.edu.my

چکیده

و الگوریتم مربوطه پیشنهاد می دهد. آزمایش های تجربی نشان داده است که این روش نسبت به روش های رایج مانند تقسیم مساوی، زمان پردازش نهایی را حدود هشت برابر کاهش می دهد.

واژه های کلیدی: رایانش مه، تئوری بار بخش پذیر، سیستم های محاسباتی توزیع شده، بارهای داده محور.

۱. مقدمه

افزایش سرعت پردازنده ها در دهه های اخیر، حیرت انگیز و از رشد نمایی برخوردار بوده است. اما این رشد حدی دارد و آن "سرعت متناهی انتشار سیگنال در اروشاد سیم" یا "حد سرعت نور" می باشد [۱]. با فرض محال سرعت انتشاری معادل سرعت نور، نهایت سرعت

محدودیت سرعت پردازنده ها، مانعی برای پردازش بارهای دارای داده های انبوه^۱ می باشد. تنها راه افزایش سرعت پردازش و کاهش زمان پردازش نهایی^۲، استفاده از سیستم های موازی و یا گسترده می باشد. در این صورت زمان بندی مطرح می شود. نوع جدیدی از زمان بندی از سال ۱۹۸۸ براساس "تئوری بار تقسیم پذیر"^۳ ارائه گردیده و پژوهش هایی در این زمینه، در محیط های مختلف رایانشی به غیر از رایانش مه، انجام شده است. این مقاله یک زمان بندی بهینه با استفاده از تئوری بار تقسیم پذیر را در محیط ترکیبی مه- ابر با ارائه معادلات و حل آنها

* نویسنده مسئول

1- Data Intensive Loads
2- Finish Time
3- Divisible Load Theory -DLT

از سال ۱۹۸۸ رده جدیدی از زمان بندی در یک مقاله توسط چنگ^{۱۷} و ربرتازی^{۱۸} [۴] و در مقاله دیگری توسط آگراوال^{۱۹} [۵] به طور مجزا پیشنهاد شد. این زمان بندی به چگونگی پردازش برنامه هایی که دارای داده های حجیم و به شدت مبتنی بر داده می باشند و قابلیت تقسیم پذیری دلخواه و بدون وابستگی به یکدیگر را دارند، در یک سیستم گسترده می پردازد. این زمان بندی که زمان بندی بار تقسیم پذیر گفته می شود، بر اساس تئوری بار بخش پذیر^{۲۰} بنا نهاده شده است. این نظریه، موازی سازی داده ها را در یک بار محاسباتی شناسایی و از آن بهره برداری می کند. همچنین محدودیت های وابستگی سیستم، مانند تاخیرهای ارتباطی و ویژگی های پردازنده را در نظر می گیرد. این نظریه، بارهای محاسباتی را به بخش های مختلف، به گونه ای تقسیم می کند که پردازش این بخش ها بر روی پردازنده های جداگانه، به طور هم زمان پایان یابند [۶]. در این روش پردازنده ها دارای برنامه های یکسان هستند که در طبقه بندی سیستم های پردازش موازی فلااین-جانسون^{۲۱}، منطبق بر معماری "یک دستورالعمل و چندین داده"^{۲۲} می باشد [۸]. نکته قابل توجه و کلیدی این نوع زمان بندی این است که، تاخیرات پیوندهای ارتباطی بین پردازنده ها، مدنظر و مورد محاسبه قرار می گیرد [۶].

این مقاله به طور خاص به زمان بندی براساس DLT در محیط مه، همراه با محیط ابر با لحاظ تاخیرات مختلف ارتباطی بین پردازنده ها، پرداخته است. گرچه زمان بندی های بسیاری بر اساس DLT در محیط ابر [۲، ۳] و خوشه [۷] و لبه [۸] انجام شده است و به نتایج قابل توجهی رسیده است، ولی نوآوری ما ارائه روش جدیدی برای زمان بندی بر اساس تئوری بار بخش پذیر در بستر ترکیبی رایانش ابر و مه می باشد. در این راستا، نمودار زمانی توزیع بار و معادله نحوه محاسبه و تقسیم بار اختصاص

ممکن در حدود ۱۰ گیگا هرتز است. گرچه عملاً سرعت انتشار سیگنال، حداکثر تا ۷۰٪ سرعت نور خواهد بود. محدودیت سرعت پردازنده ها مانعی برای انجام محاسبات انبوه و یا پردازش فایل های داده ای بسیار بزرگ می باشد و تنها راه افزایش سرعت پردازش و افزایش کارایی، استفاده از چندین پردازنده به صورت هم زمان و به عبارتی استفاده از سیستم های موازی و یا توزیع یافته مانند رایانش توری^۴، رایانش خوشه^۵، رایانش ابر و و رایانش لبه^۶ و رایانش مه می باشد [۱].

در اینجا است که مسئله زمان بندی مطرح می شود. هدف اصلی زمان بندی، رسیدن به مواردی مانند استفاده بهینه از زمان CPU، اختصاص بهینه منابع محاسباتی به برنامه ها، صف بندی، اولویت بندی کارها^۷ و وظیفه ها^۸ برای پردازش جهت به دست آوردن کمترین زمان انتظار^۹، زمان پاسخ^{۱۰}، و استفاده بهینه از منابع شبکه است. حاصل زمان بندی، به دست آوردن حداکثر توان عملیاتی، قابلیت اعتماد با در نظر گرفتن مقاومت در مقابل خرابی ها و از کار افتادگی منابع محاسباتی، توزیع عادلانه بار محاسباتی، کمینه ساختن زمان رفت و برگشت کار، مقیاس پذیری برای استفاده بهینه از تمام منابع و کاهش مصرف انرژی و افزایش کیفیت سرویس^{۱۱} می باشد [۲، ۳].

آنچه در ادبیات زمان بندی بیشتر رایج است، "زمان بندی منبع"^{۱۲} و "زمان بندی وظیفه"^{۱۳} و "زمان بندی کار یا برنامه کاربردی"^{۱۴} می باشد که در بعضی مقالات از دو زمان بندی آخر به عنوان "زمان بندی بار"^{۱۵} نیز نام برده شده است و با آنچه ما از آن به عنوان "زمان بندی بار بخش پذیر دلخواه"^{۱۶} نام می بریم، متفاوت است.

- 4- Grid Computing
- 5- Cluster Computing
- 6- Edge Computing
- 7- Jobs
- 8- Tasks
- 9- Waiting Time
- 10- Response Time
- 11- Quality of Service –QoS-
- 12- Resource Scheduling
- 13- Task Scheduling
- 14- Job/Application Scheduling
- 15- Load Scheduling
- 16- Arbitrarily Divisible Load

17- Cheng

18- Robertazzi

19- Agraval

20- Divisible Load Theory –DLT-

21- Flynn–Johnson

22- Single Instruction Multiple Data –SIMD-

به تأخیر^{۲۳} کاربر مناسب می‌باشند [۱۳]. در رایانش مه، برخلاف رایانش ابری، تمام دستگاه‌ها و گره‌ها^{۲۴} توسط شبکه‌هایی با سرعت و پهنای باند زیاد به هم وصل نیستند [۱۴]. انواع دستگاه‌های محیط مه شامل کامپیوترهای سرویس‌دهنده^{۲۵}، دستگاه‌های شبکه^{۲۶}، سرویس‌دهنده‌های کوچک ابری^{۲۷}، ایستگاه‌های پایه^{۲۸}، و خودروها^{۲۹} می‌باشند [۱۵].

بنابراین، برای زمان‌بندی در محیط مه، باید ضمن لحاظ موارد فوق، کمترین زمان ممکن پردازش و اولویت‌های کارها و وظایف را هم در نظر گرفته و به عدم توان محاسباتی و عدم توان ذخیره سازی سیستم‌ها یا گره‌های هسته مه در مقایسه با ابر، نیز توجه داشت. همچنین لازم است که اتصالات غیرهمگون^{۳۰} و سرعت‌های متفاوت پردازشگرها و شدت محاسبات و ارتباطات و توپولوژی‌های میان ارتباطی و سیاست زمان‌بندی نیز به حساب آورده شود [۶]. با توجه به موارد فوق، زمان‌بندی در محیط مه بر مبنای DLT، به دلیل در نظر گرفتن تاخیرات اتصالات بین پردازنده‌ها و عدم محدودیت همگونی پردازنده‌ها، پیشنهاد گردیده است.

۲-۲- تئوری بار تقسیم‌پذیر^{۳۱}

تئوری بار بخش‌پذیر یک روش با بنیان قوی ریاضی می‌باشد و اولین بار بر اساس دو مقاله در ۱۹۸۸ توسط چنگ، ربرتازی و آگراوال ابداع شد [۴، ۵]. این تئوری، موازی‌سازی داده‌ها را در یک بار شدید محاسباتی و داده محور^{۳۲}، معین و استخراج می‌کند و محدودیت‌های وابستگی سیستمی مانند تأخیرهای ارتباط و ویژگی‌های پردازنده‌ها را مدنظر قرار می‌دهد و برای تقسیم‌بندی و توزیع بارهای شدید محاسباتی به‌صورت بهینه بین

یافته، جدیدی ارائه شده است و پس از حل معادلات مرتبط، ارزیابی روش گزارش شده است.

۲. ادبیات موضوع

در این بخش نگاهی به موضوعات مرتبط، مانند رایانش مه و ابر، انواع بار تقسیم‌پذیر و بارهای داده محور می‌پردازیم.

۲-۱- رایانش (محاسبات) مه و زمان‌بندی در محیط مه

گرچه رایانش ابری با استفاده فناوری ماشین مجازی توانسته است به عنوان یک الگوی جدید برای ارائه همه نوع خدمات محاسباتی پویا به کاربران به عنوان یک سرویس در مراکز داده مدرن مدنظر قرار گیرد، ولی افزایش دستگاه‌های IoT باعث شده است که در رایانش ابری، شبکه با تأخیر زیاد و ازدیاد ترافیک و ازدحام مواجه شود. [۹، ۱۰]. بنابراین، برای مدیریت این رشد سریع، و رفع مشکلات ناشی از آن، مفاهیم و فناوری‌های جدیدی لازم بود و در نتیجه رایانش مه پیشنهاد شد [۱۱].

رایانش مه، مفهومی معرفی شده توسط شرکت سیسکو، توسعه رایانش ابری است که از هسته به لبه شبکه منتقل شده است. این باعث می‌شود که محاسبات در لبه شبکه که دارای فقط یک hop فاصله با دستگاه‌های اینترنت اشیا یا کاربران نهایی می‌باشد، انجام شود. در واقع محاسبات مه یک جایگزین برای محاسبات ابر نیست، بلکه یک مکمل قدرتمند است که پردازش را در لبه شبکه ممکن می‌سازد و امکان تعامل با ابر را دارد. بنابراین رایانش مه برای توسعه ابر و برای انجام سرویس‌هایی همچون محاسبات و ذخیره‌سازی و دیگر سرویس‌های شبکه، در لبه شبکه پیشنهاد شده است [۱۱].

پس به طور کلی طبق آنچه در طبیعت است، مه، ابر نزدیک به زمین است [۱۲]. رایانش مه به کاربر نهایی نزدیک است و مزیت تأخیر کم و آگاهی از محل دستگاه‌ها را دارد. بنابراین لایه مه برای پردازش تقاضاهای حساس

23- Real-Time
24- Nodes
25- Servers
26- Network Devices
27- Cloudlets
28- Base Stations
29- Vehicles
30- Heterogeneous Links
31- Divisible Load Theory – DLT-
32- Data Intensive

۲-۳- طبقه بندی انواع بار

۲-۳-۱- بارهای غیر قابل تقسیم

این بارها مستقل، غیرقابل تقسیم و به طور کلی با اندازه‌های مختلف می‌باشند. این بدین معنی است که بار نمی‌تواند بیشتر از این تقسیم شود و باید به طور تمام و کمال در یک پردازنده، پردازش شود. این بارها هیچ رابطه از پیش معینی ندارند.

۲-۳-۲- بارهای قابل تقسیم ماژولار

این بارها از قبل بر اساس بعضی ویژگی‌های سیستم و یا بار، به قسمت‌های کوچک‌تر و یا وظیفه‌هایی تقسیم می‌شوند. این بارهای کوچک به عنوان وظیفه/ زیروظیفه یا ماژول نامیده می‌شوند. وقتی گفته می‌شود که پردازش یک بار کامل شده است که تمام ماژول‌های پردازش شده باشند.

۲-۳-۳- بارهای قابل تقسیم دلخواه

این نوع بار، ویژگی خاصی دارد که تمام عناصر آن، خواستار یک نوع پردازش واحد می‌باشند. این بارها می‌توانند به طور دلخواه به هر تعدادی از بخش‌های کوچک‌تر تقسیم‌بندی شوند. این بخش‌های بار ممکن است رابطه از پیش معین شده‌ای داشته یا نداشته باشند. اگر بخش‌های بار روابط از پیش معین شده‌ای نداشته باشند، آنگاه هر بخش از بار می‌تواند مستقلاً پردازش شود. در این مقاله جهت زمان‌بندی، بارهای به طور دلخواه بخش‌پذیری را بین چندین پردازنده مدنظر قرار می‌دهد که روابط از پیش تعریف شده‌ای ندارند. این چنین بارهایی، با کاربردهای زیادی مواجه شده‌اند. بعضی از آن‌ها عبارت‌اند از: استخراج ویژگی و تشخیص لبه در پردازش تصویر، پردازش سیگنال، بینایی ماشین، تشخیص چهره، تشخیص الگو، محاسبات ماتریس‌ها، محاسبات مالی و محاسبات مهندسی [۱۷].

۲-۴- بارهای داده‌محور^{۳۶} و محاسبه‌محور^{۳۷}

رویکردهای پردازش موازی را می‌توان به طور کلی به

پردازنده‌ها، در شبکه به کار گرفته می‌شود. نکته کلیدی اینجاست که ضمن لحاظ تاخیر پیوندهای ارتباطی، نحوه تقسیم بندی بایستی به گونه‌ای باشد که پردازش تمام بارها در یک زمان به پایان برسند [۶]. ابتدا چنگ این تئوری را برای شبکه‌ای از حسگرها پیشنهاد کرد ولی بعدها کاربردهای قابل توجهی در محاسبات بیدرنگ [۱۶] جبر خطی و ماتریس‌ها [۱۷]، محاسبات پایگاه‌های داده [۱۸]، پردازش داده‌های رادار و حسگرهای مادون قرمز [۱۹]، پردازش تصویر [۲۰]، بینایی ماشین [۲۱]، برنامه‌های ویدیو و محیط‌های چندرسانه‌ای [۲۲]، نسل ۵، سیستم‌های مراقبت‌های بهداشتی، رسانه‌های اجتماعی، تجارت، دولت، تحقیقات علمی و چندین کاربرد دیگر برای داده‌های حجیم پیشنهاد شد. در تمام کاربردهای ذکر شده، بایستی حجم عظیمی از داده‌ها پردازش شوند و هدف نهایی، کمینه کردن زمان پردازش بارهای محاسباتی است [۲۵-۲۳، ۶].

DLT در توپولوژی‌های مختلفی همچون STAR، BUS و MESH و در محاسبات توزیع شده^{۳۴} از جمله GRID و CLUSTER و CLOUD مورد استفاده قرار گرفته است [۲۶]. همچنین اخیراً چالش‌های مختلفی در DLT مدنظر قرار گرفته است که برای اطلاعات بیشتر می‌توانید به منابع [۲۷ و ۲۸] مراجعه نمایید.

در مقاله حاضر، با استفاده از قابلیت‌های موجود در DLT برای زمان‌بندی در محیط مه راه‌حلی پیشنهاد شده است. مدل پیشنهادی برای زمانی که داده‌های ارسال شده به گره‌های مه^{۳۵} به طور دلخواه بخش‌پذیر و حجیم و انبوه و شدیداً محاسباتی باشند، فوق‌العاده موثر است. برای مدل پیشنهادی که ساختار درخت یک سطحی دارد، معادله طراحی و سپس حل شده و الگوریتم پیشنهاد شده است. نتایج تجربی نشان می‌دهد که روش DLT می‌تواند به طور قابل ملاحظه‌ای کارایی را در زمان‌بندی مه بالا ببرد.

33- Real-Time

34- Distributed Computing

35- Fog Nodes

36- Data Intensive

37- Compute intensive

میانی و خدمت دهنده مه را به جای استفاده از محیط ابر پیشنهاد می‌دهد. در آن مقاله علاوه بر زمان‌بندی وظیفه در محیط مه، از الگوریتم Ford-Fulkerson برای تعدیل بار در شبکه‌ای از گره‌های مه استفاده شده است.

(۳) - زمان‌بندی وظیفه و واگذاری منابع در محیط مه با کاربرد صنعتی و ساخت و تولید طی مقاله [۳۳] که در سال ۲۰۱۸ انتشار یافته، بیان گردیده است. در آن مقاله اشاره به لزوم استفاده از محیط مه برای تشخیص خرابی و تحلیل حالت دستگاه‌ها در خطوط مونتاژ به‌عنوان لایه‌ای بین دستگاه‌های نهایی و ابر صنعتی دارد؛ زیرا محیط مه دارای ویژگی زمان تأخیر کمتر نسبت به ابر می‌باشد و سرویس‌های با تأخیر کم و حساس به زمان پاسخ را فراهم می‌کند. در آن مقاله به محدودیت منابع و در نتیجه محدودیت استفاده از کاربردهای فناوری جدید مجازی‌سازی برای استفاده از مزایای محاسبات مه است، به‌عنوان یک چالش اشاره کرده و راه‌حل جدیدی را برای رفع این معضل پیشنهاد می‌کند. آن مقاله یک مدل جدید زمان‌بندی وظیفه‌ها را بر اساس نقش containerها پیشنهاد می‌کند نتایج نشان داده شده، حاکی از آن است که الگوریتم پیشنهادی زمان‌بندی وظیفه‌ها و سازوکار اختصاص مجدد، می‌تواند تا خیرات وظیفه‌ها را به طور مؤثر کاهش دهد و تعداد وظیفه‌های هم‌زمان در گره‌های مه را افزایش و بهبود دهد.

(۴) - در محیط رایانش لبه تلفن همراه، در مرجع [۸] بر اساس تئوری بارهای تقسیم‌پذیر، یک مدل نظری برای بخش‌بندی برنامه کاربردی تقسیم‌پذیر دلخواه با هدف به حداقل رساندن زمان اتمام برنامه‌ها، ارائه شده است. این مقاله تصاویر OCR^{۴۲} را به‌عنوان نوعی داده معمولی که قابل تقسیم دلخواه است، برای محیط‌های محاسباتی متحرک و پویا، مناسب تشخیص داده است و با استفاده از تئوری بار قابل تقسیم، داده‌ها را به بخش‌های کوچک‌تر تقسیم و به واحدهای محاسباتی موجود، به گونه‌ای اختصاص داده است که محاسبات در کمترین زمان ممکن انجام گردد.

دو دسته محاسبات‌محور و داده‌محور طبقه‌بندی کرد [۲۹، ۳۰]. داده‌محور یا شدیداً داده‌ای، برای توصیف برنامه‌هایی استفاده می‌شود که محدود به ورودی/خروجی (I/O) هستند یا نیاز به پردازش حجم زیادی از داده دارند. چنین برنامه‌هایی بیشتر زمان پردازش خود را به I/O و جابه‌جایی و دستکاری داده‌ها اختصاص می‌دهند.

محاسبات‌محور، برای توصیف برنامه‌های کاربردی که محدود به محاسبه هستند استفاده می‌شود. چنین برنامه‌هایی بیشتر زمان اجرای خود را بر خلاف I/O به نیازمندی‌های محاسباتی اختصاص می‌دهند و معمولاً به حجم کمی از داده‌ها نیاز دارند.

۳. پیشینه کارهای انجام شده

مقالات متعددی برای زمان‌بندی کار/برنامه^{۳۸} و زمان‌بندی وظیفه^{۳۹} و زمان‌بندی منابع^{۴۰} در محیط مه ارائه شده است [۳۴-۳۱]. همچنین در زمینه زمان‌بندی بر اساس DLT نیز مقالات اندکی در محیط‌های مختلف، به استثنای محیط رایانشی مه، ارائه شده است [۳۷-۳۵، ۷، ۸]. در ادامه بعضی از آنها بررسی و خلاصه آن‌ها در جدول شماره (۱) ارائه شده است.

(۱) - مسئله زمان‌بندی بیدرنگ برای افزایش کیفیت سرویس^{۴۱} و تضمین کارایی در محیط رایانش خوشه‌ای در مرجع [۷] ارائه شده است. در آن مقاله برای نحوه تقسیم بار کاری برای زمان‌بندی بیدرنگ، روش‌های OPR، EPR، پیشنهاد شده است که OPR براساس DLT و EPR براساس تقسیم مساوی بار است.

(۲) - در مرجع [۳۴] برای پردازش داده‌های بزرگ و بخصوص استفاده از سرویس‌های بیدرنگ یک مدل سلسله‌مراتبی شامل حسگرهای اینترنت اشیا در پایین‌ترین لایه و سپس گره‌های پردازشی ابتدایی در لایه مه و سپس گره‌های اصلی پردازشی و لایه بالاتر گره‌های

38- Job Scheduling
39- Task Scheduling
40- Resource Scheduling
41- Quality of Service -QoS

طبق بررسی‌های انجام شده، استفاده از تئوری بار بخش‌پذیر تنها در مرجع [۸] در محیط محاسبات لبه تلفن همراه استفاده شده است. در این مرجع فرض شده است که تمام بارها بین دستگاه‌های لبه تلفن همراه توزیع می‌شوند، در حالی که در روش پیشنهادی ما، با توجه به شرایط و اندازه بار، تقسیم بار علاوه بر گره‌های مه جاری که شامل هر دستگاهی می‌شود، بین مه‌های مجاور و ابر نیز امکان پذیر می‌باشد.

۴. مدل پیشنهادی

۴-۱. شمای کلی سیستم پیشنهادی و چهارچوب

توپولوژی مدل سیستم پیشنهادی بر اساس درخت یک سطحی یا ستاره^{۴۶} است. در این مدل پردازنده ریشه مجهز به یک پردازنده جلویی یا پیش پردازنده^{۴۷} می‌باشد. پیش‌پردازنده، یک پردازنده تقریباً مستقل است که اجازه می‌دهد، گره ریشه درگیر توزیع بارها نشود و بخشی از بار را خودش پردازش کند و بدون انتظار تکمیل پردازش و یا اشغال بودن، بخش‌های بار را به سایر گره‌های در دسترس جهت پردازش، ارسال کند. در مدل پیشنهادی، سرعت پردازنده‌ها متفاوت و سرعت پیوندها در گره‌های مه محلی یکسان و در مه‌های همسایه و ابر، متفاوت می‌باشند. همچنین انتشار بار در مه محلی به صورت "توزیع ترتیبی بار" و در مه‌های مجاور و ابر به صورت "هم زمان با شروع همزمان" (پردازش در گره‌های فرزند به محض شروع به دریافت بار، آغاز می‌شود) می‌باشد [۲۳]. شمای کلی مدل پیشنهادی در شکل (۱) ارائه شده است.

۴-۲. فرموله کردن مسئله

داده‌ها و رکوردهایی که از طرف حسگرها و دستگاه‌های اینترنت اشیا از طریق محیط بی‌سیم و یا باسیم به گره ریشه یک مه محلی می‌رسد، می‌تواند یک دستگاه هدایت‌کننده^{۴۸} مجهز به پردازنده باشد. نرم‌افزار زمان بند^{۴۹} در این گره

(۵) - نحوه واگذاری بار کاری بر اساس DLT برای طولانی کردن مدت زمان حس کردن حسگرها در مرجع [۳۵] ارائه گردیده است. در هنگام تصمیم‌گیری در باره میزان کار اختصاص یافته، میزان انرژی باقی‌مانده روی حسگرهای پردازشی در نظر گرفته می‌شود و به حسگرهایی که انرژی کمتری دارند، بار کمتری اختصاص می‌یابد. با این استراتژی تعداد دفعات حس کردن حسگر افزایش می‌یابد.

(۶) - مرجع [۳۶] به مصالحه^{۴۳} زمان و انرژی در پردازش بارهای تقسیم‌پذیر روی سیستم‌های ناهمگون می‌پردازد، برای جلوگیری از هزینه‌های اجرای بارها، پیشنهاد می‌دهد که قسمت‌های بار تقسیم شده بر اساس تئوری بار بخش‌پذیر روی پردازشگرهای کارگر^{۴۴}، در چندین مرحله^{۴۵} توزیع شوند. در آن مقاله نشان داده شده که رویکرد کلاسیک تقسیم بار به صورت مساوی برای الگوریتم‌های با پیچیدگی غیرخطی مناسب نیست مگر این که روش‌های چابک تقسیم‌بندی داده‌ها مانند تئوری بار بخش‌پذیر به کار برده شود.

(۷) - نویسندگان مقاله مرجع [۳۷]، برای تقسیم و توزیع بار کاری در بین منابع محاسباتی، یک زیرساخت محاسباتی درون ترانزیتی، متشکل از یک مجموعه از منابع محاسباتی، شامل یک مرکز داده ابری و مقدار مشخصی از مراکز داده را که مبدا (کاربر) و مقصد (مرکز داده ابری) را به هم متصل می‌کند، پیشنهاد کرده‌اند تا به کمترین زمان پردازش دست یابند. آنها بر این باورند که داده‌های نجومی در مقیاس بزرگ (مانند داده‌های تولید شده در مرکز نجوم SLOAN آمریکا) نمی‌توانند به طور مؤثر و بدون سیستم‌های محاسباتی مبتنی بر شبکه پردازش شوند. در آن مقاله برای تقسیم بهینه بار قابل تقسیم در شبکه‌های توپولوژی‌های مختلف، استفاده از تئوری بار بخش‌پذیر را برای رسیدن به کوتاه‌ترین زمان پردازش، در سیستم‌های محاسبات ابری پیشنهاد شده است. جدول (۱) خلاصه‌ای از این بررسی را نمایش می‌دهد.

46- Star

47- front end processor - FEP

48- Router

49- Scheduler

43- trade-offs

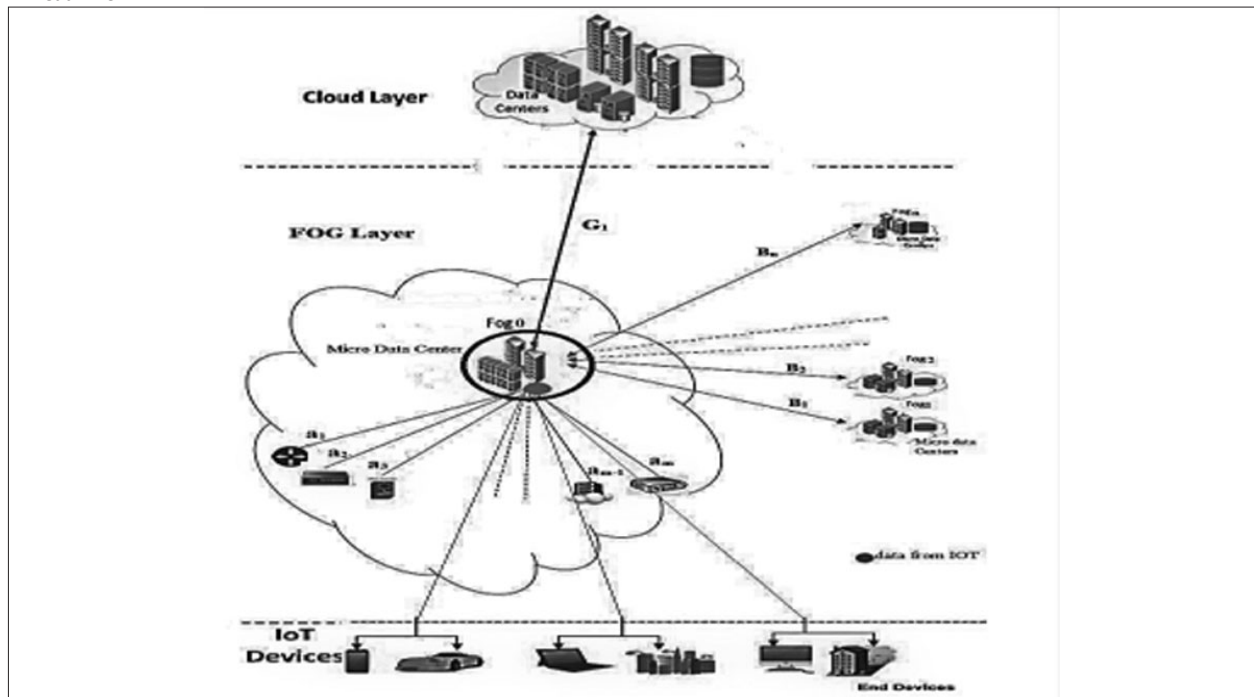
44- worker

45- installment

جدول (۱): کارهای مرتبط با زمان‌بندی در محیط مه و زمان‌بندی بر مبنای DLT

منبع	کاربرد	الگوریتم پیشنهادی	متغیرهای هدف	تکنیک استفاده شده	ابزار شبیه‌سازی
A Lin, et al, 2010 [7]	زمان‌بندی بار تقسیم‌پذیر برای محاسبات خوشه‌ای	Cluster-Based, Real-Time Scheduling Algorithm Maximum Cost Derivative First (MCDF) algorithm	حداقل تعداد پردازنده‌های لازم برای رعایت آخرین فرصت	توسعه DLT استاندارد، برای انجام محاسبات خوشه‌ای و تقسیم بار	Discrete Simulator
Tajul Islamand M.M.A Hashem 2018 [34]	زمان‌بندی وظیفه و توازن بار برای داده‌های بزرگ در مه	Ford-fulkerson + priority based Queue	Network Latency, Throughput and data Centric, Maximum Data Flow	با استفاده از الگوریتم Ford-Fulkerson و توازن بار در منابع موجود می‌پردازد.	کدهای نوشته شده توسط نویسنده مقاله
Luxiu Yin, Juan Luo, [33] 2018	خطوط تولید صنعتی هوشمند.	Using container instead of VM	اختصاص منابع به وظیفه‌ها	استفاده از Container	GNOME, Linkpack
Li, Bo, et al, 2018 [8]	محاسبه بارهای قابل تقسیم دلخواه در محاسبات لبه تلفن همراه	Back-Propagation Algorithm on Network of workstations	اندازه بخش‌ها و Makespan	روش ابتکاری برای تقسیم بندی و توزیع برنامه‌های کاربردی بر اساس Closed form	کدهای نوشته شده توسط نویسنده مقاله و iFogsim
Haiyan Shi, et al, 2012 [35]	زمان‌بندی بار تقسیم‌پذیر در شبکه حسگرها	Energy Dependent Divisible Load Theory Algorithm	بخش‌هایی از بار که بین حسگرهای بی‌سیم تقسیم می‌شوند.	به حسگرهای بی‌سیم با انرژی کمتر، بار کمتری داده می‌شود	کدهای نوشته شده توسط نویسنده مقاله
Marszałkowski, & Drozdowski 2019 [36]	Time-Energy Trade-off	Processor Sorting Rule (PSR) and the Load Chunk Sizing Algorithm	Makespan of Time and Energy	DLT	ANOVA and Heuristic
Wang, Xiaoli Et al, 2018 [37]	زمان‌بندی بهینه بارهای بزرگ با استفاده از مدل محاسبات درون ترانزیتی	Genetic Algorithm (GA) الگوریتم ژنتیک	ضرایب تقسیم بار بین گره‌های درون ترانزیتی و ابر	طراحی یک تکنیک بهینه‌سازی کلی برای دستیابی به توزیع بار بهینه در بین منابع محاسباتی ناهمگن	کدهای نوشته شده توسط نویسنده مقاله

1- Dead Line



شکل (۱): شمای کلی مدل

جدول (۲) متغیرهای مستقل و وابسته به همراه علامت اختصاری آن‌ها

علامت	نام متغیر	نوع متغیر
w_i	زمان صرف شده توسط پردازشگر P_i برای محاسبه بار داده شده تقسیم بر زمان صرف شده توسط پردازنده استاندارد برای محاسبه همان بار. ما آن را نرخ پردازش، پردازنده A_m می‌نامیم. مثلاً عدد ۲ برای این متغیر بیانگر این است که سرعت این پردازنده نصف سرعت پردازنده استاندارد است	مستقل
z_i	زمان صرف شده توسط پیوند L_i برای انتقال بار داده شده تقسیم بر زمان صرف شده توسط پیوند استاندارد برای انتقال همان بار داده شده. ما آن را نرخ انتقال پیوند A_m می‌نامیم. مثلاً عدد ۲ برای این متغیر بیانگر این است که سرعت این پیوند نصف سرعت پیوند استاندارد است	
w_{Fj}	نرخ پردازش J آمین مه	
z_{Fj}	نرخ انتقال داده پیوند به J آمین مه	
w_c	نرخ پردازش پردازنده ابر	
z_c	نرخ انتقال داده پیوند گره ریشه به ابر	
n	تعداد پردازنده‌های عمل‌کننده مه مجاور	
m	تعداد پردازنده‌های عمل‌کننده مه محلی به غیر از پردازنده ریشه	
z	نرخ انتقال داده‌ها از گره ریشه مه به پردازنده‌های وابسته محلی یکسان فرض شده است	
1	زمان پردازش یک واحد بار در پردازنده استاندارد و زمان انتقال یک واحد بار توسط یک پیوند استاندارد	
T_{cp}	مقدار زمان لازم جهت پردازش یک واحد بار توسط پردازنده استاندارد	
T_{cm}	مقدار زمان لازم جهت انتقال یک واحد بار بر روی یک پیوند استاندارد	
Finish Time	زمان کل کار از شروع توزیع بار تا پردازش بارها و برگشت نتیجه	
α_i	بخش‌هایی از بار که به گره‌های مه جاری تخصیص داده می‌شود	
β_j	بخش‌هایی از بار که به گره‌های مه‌های همجوار تخصیص داده می‌شود	
γ	بخشی از بار که به ابر تخصیص داده می‌شود	

۱. گره ریشه مه محلی^۱ که اطلاعات را از طریق حسگرها و دستگاه‌های اینترنت اشیا دریافت می‌کند و خود بخشی (α_0) از آن را پردازش می‌کند و زمان ($\alpha_0 T_{cp}$) را برای پردازش صرف می‌کند ولی به دلیل این که بارها درون خودش هست، زمانی برای انتقال اطلاعات صرف نمی‌کند.

۲. تعدادی گره در مه محلی که بخش‌هایی از اطلاعات پردازش می‌کنند که زمان ($\alpha_m T_{cm}$) را برای انتقال سهم اطلاعات به گره m ام و زمان ($\alpha_m T_{cp}$) را برای پردازش سهم گره m ام صرف می‌کند.

۳. تعدادی مه مجاور که بخش‌هایی از اطلاعات ($\beta_1, \beta_2, \dots, \beta_n$) از گره ریشه به آنها ارسال و پردازش می‌شود که زمان ($\beta_n T_{cm}$) را برای انتقال سهم اطلاعات به مه n ام و زمان ($\beta_1 T_{cp}$) را برای

مقیم است. در مدل پیشنهادی گره ریشه مجهز به پردازنده جلویی^۲ یا پیش پردازنده است که اجازه می‌دهد گره ریشه بخشی از بار را خودش پردازش کند و بدون انتظار تکمیل پردازش و یا اشغال بودن، بقیه بخش‌های بار را به سایر گره‌های در دسترس جهت پردازش ارسال کند. از طرفی گره ریشه می‌تواند با مه‌های دیگری که همجوار آن هست، و نیز با محیط ابر ارتباط داشته باشد و بخش‌هایی از بار را بین تمام گره‌های در دسترس و آماده، تقسیم کند.

۴-۳. معرفی پارامترها و تعریف‌ها

جدول (۲) متغیرهای مربوطه و نسبت و علامت اختصاری آنها را نشان می‌دهد.

۴-۴. نمودار زمانی مدل پیشنهادی

شکل (۱) شمای کلی مدل پیشنهادی را مشخص می‌کند. همان‌طور که در شکل (۱) دیده می‌شود، این مدل دارای ۴ مولفه است.

کسری از یک واحد بار کاری باید اختصاص یابد تا زمان اتمام پردازش، کمینه شود.

در مرحله سوم: الگوریتم مربوطه استخراج می‌گردد. از این الگوریتم برای برنامه نویسی و پیاده‌سازی و شبیه‌سازی مدل استفاده می‌شود.

۴-۵-۱- مرحله اول - ایجاد فرم بسته

$$(\alpha_0 T_{cp})w_0 = (\alpha_1 T_{cm})z_1 + (\alpha_1 T_{cp})w_1 \quad (1)$$

$$(\alpha_1 T_{cp})w_1 = (\alpha_2 T_{cm})z_2 + (\alpha_2 T_{cp})w_2 \quad (2)$$

$$(\alpha_2 T_{cp})w_2 = (\alpha_3 T_{cm})z_3 + (\alpha_3 T_{cp})w_3 \quad (3)$$

...

$$(\alpha_{m-1} T_{cp})w_{m-1} = (\alpha_m T_{cm})z_m + (\alpha_m T_{cp})w_m \quad (4)$$

$$(\alpha_0 T_{cp})w_0 = (\gamma T_{cm})z_c + (\gamma T_{cp})w_c \quad (5)$$

$$(\alpha_0 T_{cp})w_0 = (\beta_1 T_{cm})z_{F_1} + (\beta_1 T_{cp})w_{F_1} \quad (6)$$

$$(\alpha_0 T_{cp})w_0 = (\beta_2 T_{cm})z_{F_2} + (\beta_2 T_{cp})w_{F_2} \quad (7)$$

...

$$(\alpha_0 T_{cp})w_0 = (\beta_n T_{cm})z_{F_n} + (\beta_n T_{cp})w_{F_n} \quad (8)$$

$$\alpha_0 + \alpha_1 + \alpha_2 + \dots + \alpha_m + \beta_1 + \beta_2 + \dots \quad (9)$$

$$z_1 = z_2 = z_3 = \dots = z_m = z \quad (10)$$

۴-۵-۲- مرحله دوم - حل فرم بسته

از معادله (۱) داریم:

$$\alpha_0 = \frac{w_1 T_{cp} + z T_{cm}}{w_0 T_{cp}} \cdot \alpha_1 \quad (11)$$

از معادله (۱-۴) داریم:

$$\alpha_i = \frac{w_{i-1} T_{cp}}{w_i T_{cp} + z T_{cm}} \cdot \alpha_{i-1}, \quad i = 2, 3, \dots, m \quad (12)$$

اگر قرار دهیم $\kappa_i = \frac{w_{i-1} T_{cp}}{w_i T_{cp} + z T_{cm}}$ آنگاه از معادله (۱۲)

داریم:

$$\alpha_i = \kappa_i \cdot \alpha_{i-1}, \quad i = 2, 3, \dots, m \quad (13)$$

و همچنین داریم:

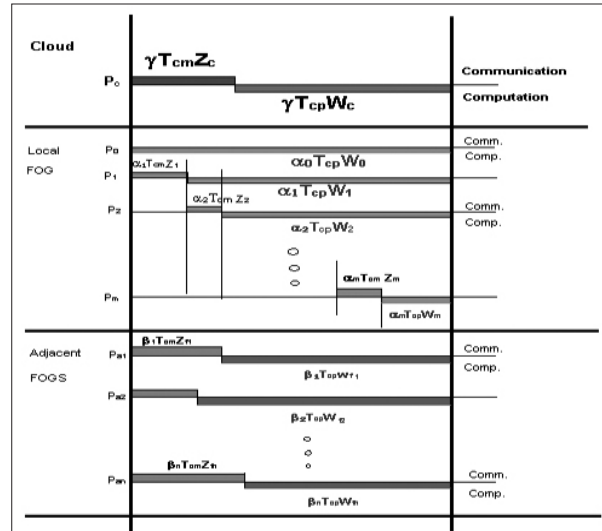
$$\alpha_i = (\prod_{l=2}^i \kappa_l) \cdot \alpha_1, \quad i = 2, 3, \dots, m \quad (14)$$

از معادله (۵) داریم:

$$\gamma = \frac{w_0 T_{cp}}{w_c T_{cp} + z_c T_{cm}} \cdot \alpha_0 \quad (15)$$

و توسط معادله (۱۲) داریم:

$$\frac{w_1 T_{cp} + z T_{cm}}{w_c T_{cp} + z_c T_{cm}} \cdot \alpha_1 = \gamma = \frac{w_0 T_{cp}}{w_c T_{cp} + z_c T_{cm}} \cdot \alpha_0 \quad (16)$$



شکل (۲) نمودار زمانی توزیع بار و پردازش آنها

پردازش سهم مه n ام صرف می‌کند.

۴. ابر که بخشی (Y) از اطلاعات به آنجا ارسال و پردازش می‌شود و زمان (gamma Tcm)zc را برای انتقال سهم اطلاعات و زمان (gamma Tcp)wc را برای پردازش سهم اطلاعات صرف می‌کند.

مهم این است که تمام این زمان‌ها باید با هم مساوی باشند تا شرط بهینگی برقرار باشد [۶]. نمودار زمانی مدل پیشنهادی طبق شکل (۲) می‌باشد.

۴-۵. توصیف ریاضی مدل پیشنهادی

توصیف ریاضی مدل سیستم پیشنهادی، با توجه به بخش‌های ۴/۲، ۴/۳ و ۴/۴ در سه مرحله انجام می‌شود.

در مرحله اول: یک معادله برای محاسبه زمان پایان ۵۲ یک بار شدیداً داده‌ای و داده محور که بین چندین پردازنده در دسترس در مه محلی و در مه‌های مجاور و ابر تقسیم می‌شوند، استخراج می‌شود. این معادله در واقع تابع هزینه سیستم است. در این معادله نرخ پردازش پردازنده‌ها و نرخ ارتباط بین آنها در نظر گرفته می‌شود. بار داده‌ای شدید و قابل تقسیم دلخواه به گره ریشه که دارای پیش پردازنده می‌باشد وارد می‌شود.

در مرحله دوم: معادله پیشنهادی حل می‌شود و مشخص می‌کند به هر یک از پردازنده‌های در دسترس، چه

از معادله (۸) داریم:

$$\beta_j = \frac{w_0 T_{cp}}{w_{F_j} T_{cp} + z_{F_j} T_{cm}} \cdot \alpha_0, \quad j = 1, 2, \dots, n \quad (17)$$

با ترکیب معادله (۱۱) و معادله (۱۷) داریم:

$$\beta_j = \frac{w_1 T_{cp} + z T_{cm}}{w_{F_j} T_{cp} + z_{F_j} T_{cm}} \cdot \alpha_1, \quad j = 1, 2, \dots, n \quad (18)$$

با استفاده از معادله (۹) و معادله (۱۱) و معادله (۱۴)

داریم:

$$\Delta \cdot \alpha_1 = 1 \quad (19)$$

که در آن:

$$\Delta = \left(\frac{w_1 T_{cp} + z T_{cm}}{w_0 T_{cp}} + 1 + \sum_{i=2}^m \prod_{i=1}^i \frac{w_{i-1} T_{cp}}{w_i T_{cp} + z T_{cm}} \right) + \frac{w_1 T_{cp} + z T_{cm}}{z_c T_{cm} + w_c T_{cp}} + \sum_{j=1}^n \frac{w_1 T_{cp} + z T_{cm}}{w_{F_j} T_{cp} + z_{F_j} T_{cm}} \quad (20)$$

در ادامه معادله زیر را به دست می آوریم:

$$\alpha_1 = \frac{1}{\Delta} \quad (21)$$

از معادله (۱۱) داریم:

$$\alpha_0 = \frac{w_1 T_{cp} + z T_{cm}}{w_0 T_{cp}} \cdot \frac{1}{\Delta} \quad (22)$$

همچنین به دست می آوریم:

$$\alpha_0 = \frac{1}{\kappa_1}, \quad \alpha_1 = \frac{w_1 T_{cp} + z T_{cm}}{w_0 T_{cp}} \cdot \frac{1}{\Delta} \quad (23)$$

بر اساس معادله (۱۴) و معادله (۲۱)، تقسیمات بهینه

بار به صورت زیر به دست می آید:

$$\alpha_i = \prod_{i=2}^i \frac{w_{i-1} T_{cp}}{w_i T_{cp} + z T_{cm}} \cdot \frac{1}{\Delta} \quad (24)$$

و زمان پایان کلی طبق معادله زیر محاسبه می شود:

$$Finish\ Time = T_{f,m+n+1} = (\alpha_0 T_{cp}) \quad (25)$$

$$w_0 = (w_1 T_{cp} + z T_{cm}) \cdot \frac{1}{\Delta}$$

همچنین $Speedup^{oz}$ طبق معادله زیر محاسبه می شود:

$$Speedup_{m+n+1} = \frac{T_{p,1}}{T_{p,m+n+1}} = \frac{T_{cp} w_0}{\alpha_0 T_{cp} w_0} = \frac{1}{\alpha_0} \quad (26)$$

۳-۵-۴. مرحله سوم - الگوریتم پیشنهادی

الگوریتم مدل پیشنهادی به صورت زیر می باشد.

الگوریتم ۱: محاسبه و تعیین بخش هایی از بار که به

پردازنده ها اختصاص می یابد

ورودی:

بار قابل تقسیم که مقدارش یک واحد در نظر گرفته

می شود

خروجی:

α_i سهم هر یک از پردازنده های (گره های) می باشد از

یک واحد بار که i بین ۰ تا m می باشد.

β_j سهم هر یک از مه مجاور از یک واحد بار از یک واحد

بار مه j بین ۱ تا n می باشد.

γ سهم ابر از یک واحد بار می باشد.

- 1: $i \leftarrow 1$
- 2: **while** ($i \leq m$) **do**
- 3: $\kappa_i = \frac{w_{i-1} T_{cp}}{w_i T_{cp} + z T_{cm}}$
- 4: **end while**
- 5: $\Delta = \left(\frac{w_1 T_{cp} + z T_{cm}}{w_0 T_{cp}} + 1 + \sum_{i=2}^m \prod_{i=1}^i \frac{w_{i-1} T_{cp}}{w_i T_{cp} + z T_{cm}} \right) + \frac{w_1 T_{cp} + z T_{cm}}{z_c T_{cm} + w_c T_{cp}} + \sum_{j=1}^n \frac{w_1 T_{cp} + z T_{cm}}{w_{F_j} T_{cp} + z_{F_j} T_{cm}}$
- 6: $\alpha_1 = \frac{1}{\Delta}$
- 7: $\alpha_1 = \alpha_0 \cdot \kappa_1$
- 8: $i \leftarrow 2$
- 9: **while** ($i \leq m$) **do**
- 10: $\alpha_i \leftarrow \kappa_i \cdot \alpha_{i-1}$
- 11: **end while**
- 12: $j \leftarrow 1$
- 13: **while** ($j \leq n$) **do**
- 14: $\beta_j = \frac{w_1 T_{cp} + z T_{cm}}{w_{F_j} T_{cp} + z_{F_j} T_{cm}} \cdot \alpha_1$
- 15: **end while**
- 16: $\gamma \leftarrow \frac{w_1 T_{cp} + z T_{cm}}{w_c T_{cp} + z_c T_{cm}} \cdot \alpha_1$
- 17: $Finish\ Time \leftarrow (w_1 T_{cp} + z T_{cm}) \cdot \frac{1}{\Delta}$
- 18: $Speedup \leftarrow \frac{T_{cp} w_0}{\alpha_0 T_{cp} w_0} = \frac{1}{\alpha_0}$

در الگوریتم فوق، پارامترهای استفاده شده طبق

جدول شماره (۱) می باشد. ورودی ها عبارتند از m (تعداد

گره های مه محلی در اطراف گره ریشه) و n (تعداد مه های

همسایه). خروجی شامل بخش هایی از بار است که بایستی

به گره ریشه و گره های دیگر مه محلی اختصاص یابند و با

$\alpha_0, \alpha_1, \dots, \alpha_m$ نمایش می شوند و $\beta_1, \beta_2, \dots, \beta_n$

بخش های از بار هستند که باید به مه های مجاور اختصاص

یابند و γ بخشی از بار است که باید به ابر اختصاص یابد.

طبق مرحله دوم (حل فرم بسته)، ابتدا α_1 و α_i محاسبه

طبق جدول (۳) براساس اطلاعات زیر در نظر گرفته شده است.

پردازشگرهای گره‌های مه براساس محصول روتر CISCO IOX (IR1101) با پردازنده ARM 64 بیتی با سرعت 1.2 GHZ و 4 GB حافظه اصلی و 64 GB حافظه جانبی مد نظر قرار گرفته است.

پردازشگر ابر نیز بر اساس ظرفیت پردازش محیط‌های ابر مانند AZURE شرکت مایکروسافت و AWS شرکت آمازون و و IBM Soft Layer شرکت آی‌بی‌ام می‌باشد و از بهترین پردازشگر ممکن کنونی یعنی AMD 3900 X RYZENA با فرکانس پایه 3.8 GHZ و ۱۲ هسته‌ای و حافظه اصلی 64 GB و حافظه جانبی نامحدود (بیش از ۱۰۰۰ ترابایت) مد نظر قرار گرفته است.

پردازشگر استاندارد نیز بر اساس آنچه معمول است با پردازنده ۶۴ بیتی و ۷ هسته و 3.2 GHZ و 4 GB حافظه اصلی و 512 GB حافظه جانبی مدنظر قرار گرفته است.

سرعت پیوند بین مه و ابر 64 Mb/S و سرعت پیوند بین گره‌های مه محلی و پیوند بین مه محلی و مه‌های مجاور 5 Mb/s در نظر گرفته شده است. بیشینه این مقدار 20 Mb/s است [۳۸].

پردازنده استاندارد که در اینجا به آن اشاره می‌شود ممکن است هر پردازنده موجود در شبکه یا برخی از پردازنده‌های ساختگی باشد که به عنوان یک مرجع مناسب عمل می‌کند [۶].

پیوند استاندارد که در اینجا به آن اشاره می‌شود ممکن است هر پیوند موجود در شبکه یا برخی از پیوندهای ساختگی باشد که به عنوان یک مرجع مناسب عمل می‌کند و جهت اطمینان، از کمینه مقدار پیوند موجود هم کمتر یعنی حدود 5 Mb/s انتخاب کرده‌ایم [۳۸، ۶].

بنابراین سرعت پردازنده مه یک پنجم تا یک سوم پردازنده استاندارد در نظر گرفته می‌شود.

سرعت پردازنده ابر سه تا پنج برابر سرعت پردازنده استاندارد در نظر گرفته می‌شود.

می‌شوند و براساس آنها، بقیه آلفاها (α_i) و بتاها (β_j) و (γ) و سرانجام "زمان پایان" محاسبه می‌شوند.

خط ۳ برای محاسبه K_i و براساس فرمول ۱۳ می‌باشد. خط ۵ و ۶ مقدار α_1 را براساس فرمولهای ۲۰ و ۲۱ محاسبه می‌نماید. α_0 در خط ۶ براساس فرمول ۱۱ و $\alpha_2, \alpha_3, \dots, \alpha_m$ در خط ۹ و ۱۰ براساس فرمول ۱۳ محاسبه شده‌اند. به این ترتیب آلفاها به دست می‌آیند. اکنون β_j ها در خط ۱۳ و ۱۴ با توجه به فرمول ۱۸ محاسبه می‌شوند. سرانجام γ در خط ۱۶ براساس فرمول ۱۶ محاسبه می‌شود. در خط ۱۷ زمان پایان براساس فرمول ۲۵ محاسبه می‌شود.

۵. ارزیابی کارایی و تجزیه و تحلیل

داده‌های تولید شده توسط اینترنت اشیا به گره ریشه وارد می‌شوند. این داده‌ها همگی دارای رکوردهای مشابه بوده و حجم عظیمی را تشکیل می‌دهند. تمام بار وارد شده، به عنوان یک واحد (عدد یک-۱) در نظر گرفته می‌شود و هدف از شبیه‌سازی، تعیین کسری از این بار واحد می‌باشد که بین گره‌ها تقسیم می‌شود. در نتیجه نیاز به وجود یک مجموعه داده با حجم مشخص نیست.

برای سرعت پیوندهای ارتباطی، یک پیوند استاندارد با سرعت واحد در نظر گرفته شده و سرعت پیوندهای دیگر، متناسب با این استاندارد به صورت تصادفی در یک دامنه مشخص در هنگام اجرا در نظر گرفته می‌شود. برای سرعت پردازنده‌ها، یک پردازنده استاندارد با سرعت واحد در نظر گرفته شده و سرعت پردازنده‌های دیگر، متناسب با این استاندارد به صورت تصادفی در یک دامنه مشخص در هنگام اجرا در نظر گرفته می‌شود. جدول (۳) در بخش بعدی، این مفروضات را نمایش می‌دهد.

۵-۱- تنظیمات پیاده‌سازی

در این پیاده‌سازی، سرعت پردازش گره‌های مه و مه‌های مجاور و ابر و همچنین سرعت پیوندهای مرتبط،

جدول (۳): جدول مفروضات سرعت‌های پردازش و ارتباط

مقدار متغیر	نام متغیر	شرح متغیر
۲-۵	w_i	زمان صرف شده توسط پردازشگر P_i برای محاسبه بار داده شده تقسیم بر زمان صرف شده توسط پردازنده استاندارد برای محاسبه همان بار. مثلاً عدد ۵ نمایانگر این است که سرعت پردازشگر یک پنجم سرعت پردازشگر استاندارد است) این عدد نرخ پردازش پردازنده I ام مستقر در مه محلی می‌باشد.
۳-۴	Z_i	زمان صرف شده توسط پیوند L_i برای انتقال بار داده شده تقسیم بر زمان صرف شده توسط پیوند استاندارد برای انتقال همان بار. (نرخ انتقال پیوند i ام) عدد ۴ نمایانگر این است که سرعت پیوند یک چهارم سرعت پیوند استاندارد است). این عدد نرخ انتقال در مه محلی بین گره ریشه و گره i می‌باشد.
۱/۵۰-۲/۵۰	w_{Fi}	نرخ پردازش i امین مه ($i=0$ به مه محلی اشاره می‌کند)
۳-۴	Z_{Fi}	نرخ انتقال داده پیوند بین گره ریشه و i امین مه
۰/۲۰-۰/۵۰	W_c	نرخ پردازش پردازنده ابر
۰/۰۷-۰/۱۰	Z_c	نرخ انتقال داده پیوند گره ریشه به ابر
۱	T_{cp}	مقدار زمان لازم جهت پردازش یک واحد بار توسط پردازنده استاندارد
۱	T_{cm}	مقدار زمان لازم جهت انتقال یک واحد بار بر روی یک پیوند استاندارد
۴-۳۲	m	تعداد پردازنده‌های عمل‌کننده در مه محلی به غیر از پردازنده ریشه
۴-۳۲	n	تعداد مه‌های مجاور

نوع زمان‌بندی رایج دیگر مقایسه گردیده است. این دو نوع عبارتند از:

الف: تقسیم بار به صورت مساوی بین تمام گره‌های آماده پردازش (گراف به رنگ نارنجی)

ب: تقسیم بار به صورت تصادفی بین تمام گره‌های آماده پردازش (گراف به رنگ خاکستری)

در این حالت این مقادیر تصادفی به عنوان اندازه وظیفه‌ها^{۱۱} در نظر گرفته می‌شود و سپس واگذاری این وظیفه‌ها بین تمام گره‌های آماده پردازش براساس واگذاری بار بیشتر به پردازنده قویتر که این الگوریتم برگرفته از الگوریتم (Best Fit-BF) [۱] می‌باشد.

نتایج شبیه‌سازی شده نشان می‌دهد که الگوریتم پیشنهادی ما در مقایسه با سایرین، در محیط مه حدود هفت برابر سریع‌تر و زمان اتمام کار در حد یک هشتم می‌باشد.

۳-۵. نتایج پیاده سازی

تمامی آزمایش‌ها و مقایسه‌ها، برای بررسی اثر نوع زمان‌بندی و تعداد گره‌های آماده پردازش، بر "زمان پایان" کار و افزایش سرعت محاسبه انجام می‌گردد. تعداد گره‌های مه محلی ثابت و تعداد مه‌های مجاور متغیر در نظر گرفته می‌شود. برای بررسی بیشتر، گره‌های ثابت

سرعت پیوند مه‌های محلی برابر یک چهارم تا یک سوم پیوند استاندارد در نظر گرفته می‌شود [۶، ۳۸].

سرعت پیوند مه‌های همسایه برابر یک چهارم تا یک سوم پیوند استاندارد در نظر گرفته می‌شود [۶، ۳۸].

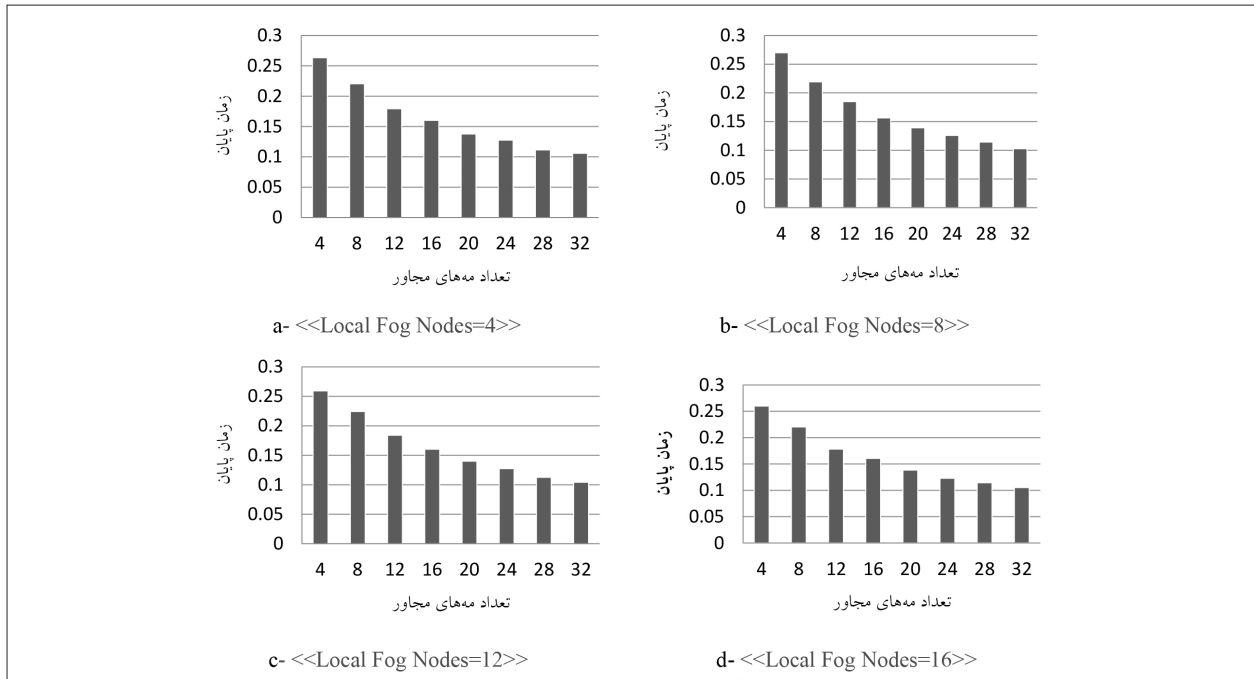
سرعت پیوند ابر ۴ تا ۱۰ برابر پیوند استاندارد در نظر گرفته می‌شود [۶، ۳۸].

بنابراین با لحاظ جدول مربوط به تعاریف متغیرها (جدول ۱) مقادیری که در شبیه‌سازی در برنامه پایتون برای این مدل مدنظر قرار گرفته به شرح جدول (۳) می‌باشد.

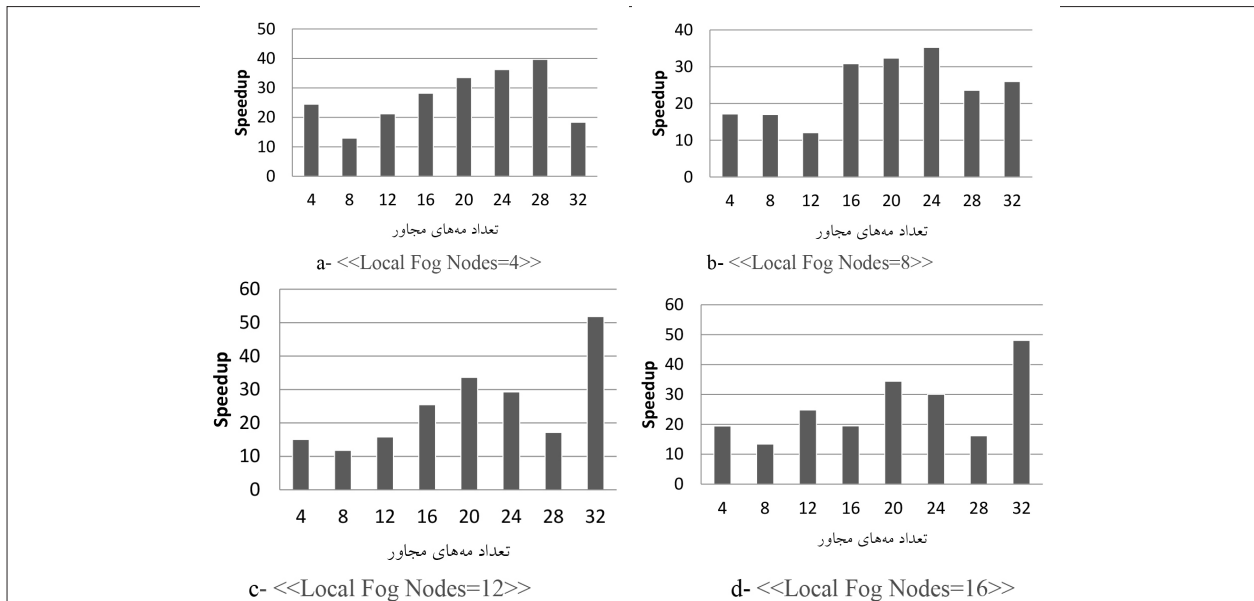
۲-۵. روش پیاده سازی

به دلیل بعضی محدودیت‌های شبیه‌سازهای موجود همچون Fogsim، برای پیاده‌سازی از زبان پایتون استفاده شده است. می‌دانیم که هدف اصلی این مقاله، تقسیم بار وارد شده به پردازنده ریشه، بین گره‌های در دسترس به صورت بهینه می‌باشد به گونه‌ای که ضمن لحاظ تاخیرات مربوط به پیوندهای ارتباطی بین گره‌ها، "زمان پایان" پردازش کل بار وارد شده، کمینه گردد. این گره‌ها شامل گره‌های آماده پردازش در مه محلی، و مه‌های مجاور و ابر می‌باشند.

برای نشان دادن کارایی روش پیشنهادی، نتایج استفاده از تئوری بار بخش‌پذیر (گراف آبی رنگ) با دو



شکل (۴) نمودار زمان اتمام کار بر اساس DLT



شکل (۵) نمودار Speedup بر اساس DLT

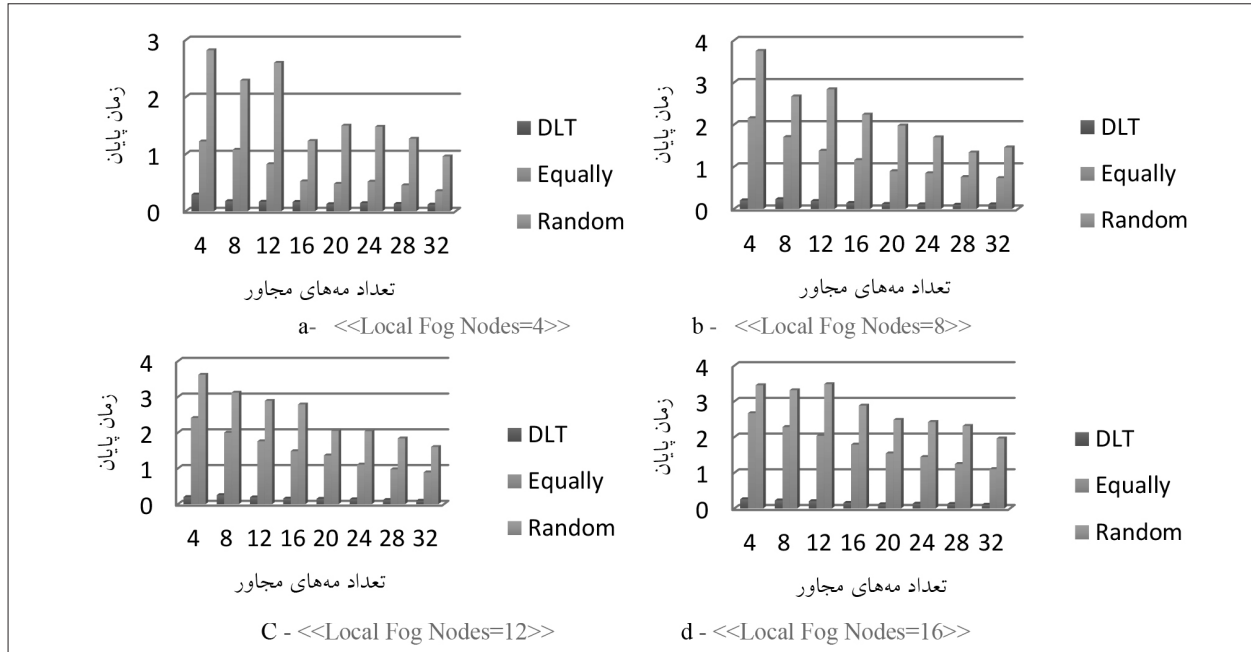
آماده پردازش مه محلی برابر ۴ در نظر گرفته شده است (شکل a) و تعداد مه‌های مجاور ۴، ۸، ۱۲، ۱۶، ۲۰، ۲۴، ۲۸ و ۳۲ در نظر گرفته شده‌اند. جهت بررسی بیشتر، این آزمایش برای تعداد گره‌های مه محلی علاوه بر ۴، برای ۸ (b) و ۱۲ (c) و (۱۶) هم تکرار شده است. ۲-۳-۵. بررسی سرعت بر اساس DLT (شکل ۵)

در هر حالت، برای اعداد ثابت ۴، ۸، ۱۲ و ۱۶ شبیه‌سازی شده‌اند.

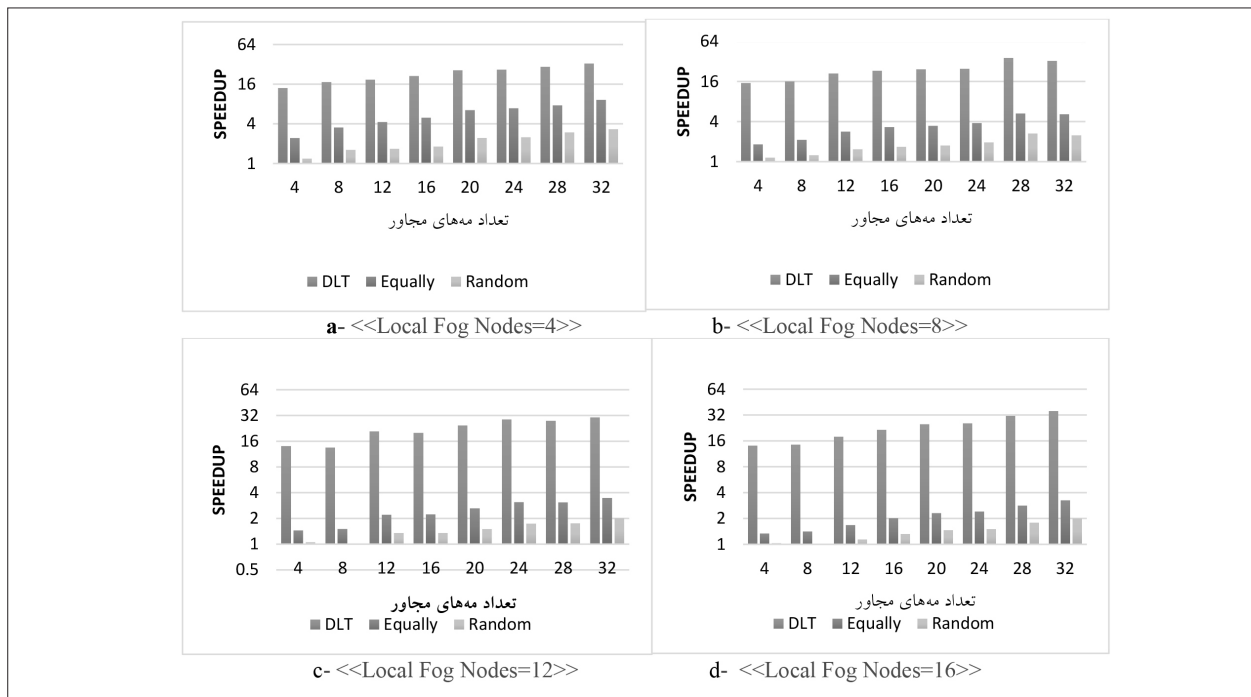
۵-۳-۱- بررسی زمان پایان^{۵۶} بر اساس DLT

در شکل (۴) نمودار زمان اتمام یک واحد کار را که بر اساس تئوری بار بخش‌پذیر^{۵۷} به دست آمده است، نشان داده شده است. در این آزمایش ابتدا تعداد گره‌های

56- Finish Time
57- Divisible Load Theory –DLT-



شکل (۷) نمودار مقایسه زمان اتمام کار بین روش‌های "تقسیم براساس DLT"، "تقسیم مساوی" و "تقسیم تصادفی و اختصاص بار بزرگتر به گره سریع‌تر"



شکل (۸): نمودار مقایسه Speedup بین روش‌های "تقسیم براساس DLT"، "تقسیم مساوی" و "تقسیم تصادفی و اختصاص بار بزرگتر به گره سریع‌تر"

به پردازنده سریع‌تر)، نشان می‌دهد. در این آزمایش تعداد گره‌های آماده پردازش مه محلی برابر ۴ در نظر گرفته شده است و تعداد مه‌های مجاور ۴، ۸، ۱۲، ۱۶، ۲۰، ۲۴، ۲۸ و ۳۲ در نظر گرفته شده‌اند. جهت بررسی بیشتر، این آزمایش برای تعداد گره‌های مه محلی علاوه بر ۴، برای ۸، ۱۲ و ۱۶ هم تکرار شده است. آزمایش نشان می‌دهد که بعد

۳-۳-۵. مقایسه زمان پایان^{۵۸} بین روش‌های مختلف در شکل (۷) نمودار زمان اتمام کار را به صورت مقایسه‌ای بین سه روش فوق (تقسیم بار بر اساس تئوری بار بخش‌پذیر- تقسیم بار به صورت مساوی- تقسیم بار به صورت تصادفی و واگذاری بخش بزرگتر

Algorithm in Cloud Computing: A Multi-criteria Approach. "Azerbaijan Journal of High Performance Computing 2, no. 1 (2019): 29-38.

[4] Cheng, Yuan-Chieh and Thomas G. Robertazzi. "Distributed computation with communication delay (distributed intelligent sensor networks)." IEEE transactions on aerospace and electronic systems 24, no. 6 (1988): 700-712.

[5] Agrawal, Rakesh, and H. V. Jagadish. "Partitioning techniques for large-grained parallelism." IEEE Transactions on Computers 37, no. 12 (1988): 1627-1634.

[6] Bharadwaj, Veeravalli, Debasish Ghose, Venkataraman Mani, and Thomas G. Robertazzi. Scheduling divisible loads in parallel and distributed systems. Vol. 8. John Wiley & Sons, 1996.

[7] Lin, Xuan, Anwar Mamat, Ying Lu, Jitender Deogun, and Steve Goddard. "Real-time scheduling of divisible loads in cluster computing environments." Journal of Parallel and Distributed Computing 70, no. 3 (2010): 296-308.

[8] Li, Bo, Min He, Wei Wu, Arun Kumar Sangaiah, and Gwanggil Jeon. "Computation offloading algorithm for arbitrarily divisible applications in mobile edge computing environments: An OCR case." Sustainability 10, no. 5 (2018): 1611.

[9] Iorga, Michaela, Larry Feldman, Robert Barton, Michael J. Martin, Nedim S. Goren, and Charif Mahmoudi. "Fog computing conceptual model." (2018).

[10] Mahmud, Redowan, Ramamohanarao Kotagiri, and Rajkumar Buyya. "Fog computing: A taxonomy, survey and future directions." In Internet of everything, pp. 103-130. Springer, Singapore, 2018.

[11] Cisco, "Cisco global cloud index: Forecast and methodology, 2015-2020," Cisco, 2015.

[12] Mouradian, Carla, Diala Naboulsi, Sami Yangui, Roch H. Glitho, Monique J. Morrow, and Paul A. Polakos. "A comprehensive survey on fog computing: State-of-the-art and research challenges." IEEE communications surveys & tutorials 20, no. 1 (2017): 416-464.

[13] Li, Guangshun, Yuncui Liu, Junhua Wu, Dandan Lin, and Shuaishuai Zhao. "Methods of resource scheduling based on optimized fuzzy clustering in fog computing." Sensors 19, no. 9 (2019): 2122.

[14] Chu, Hong-Min, Shao-Wen Yang, Padmanabhan Pillai, and Yen-Kuang Chen. "Scheduling in visual fog computing: NP-completeness and practical efficient solutions." In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[15] Haddad, Emile. "Runtime reallocation of divisible load under processor execution deadlines." In Proceedings of Third Workshop on Parallel and Distributed Real-Time Systems, pp. 30-31. IEEE, 1995.

[16] Chuprat, Suriyati. "Divisible load scheduling of real-time task on heterogeneous clusters." In 2010 International Symposium on Information Technology, vol. 2, pp. 721-726. IEEE, 2010.

[17] Chan, S. K., Veeravalli Bharadwaj, and Debasish Ghose. "Large matrix-vector products on distributed bus networks

از روش DLT روش، تقسیم مساوی، مناسب‌تر است.

۳-۴-۵. مقایسه سرعت پردازش (Speedup) بین

روش‌های مختلف

در شکل (۸) نمودار Speedup به صورت مقایسه‌ای

بین سه روش "تقسیم بر اساس تئوری بار بخش‌پذیر" و

"تقسیم مساوی" و "تقسیم تصادفی" را نشان می‌دهد. در

این آزمایش ابتدا تعداد گره‌های آماده پردازش مه محلی

برابر ۴ در نظر گرفته شده است و تعداد مه‌های مجاور ۴،

۸، ۱۲، ۱۶، ۲۰، ۲۴، ۲۸ و ۳۲ در نظر گرفته شده‌اند. جهت

بررسی بیشتر، این آزمایش برای تعداد گره‌های مه محلی

علاوه بر ۴، برای ۸، ۱۲ و ۱۶ هم تکرار شده است. آزمایش

نشان می‌دهد که بعد از روش DLT روش تقسیم مساوی،

مناسب‌تر است.

۶. نتیجه‌گیری و خلاصه

در این مقاله الگوریتمی برای کاهش تاخیر^۹ بر مبنای

DLT در محیط رایانش مه پیشنهاد گردیده است. سپس

الگوریتم پیشنهادی بسط داده شده و برای آن کلوز فرم

طراحی و حل گردیده و براساس آنها نتایجی با استفاده

از آزمایش‌های مختلف به دست آمده است. در بخش

نتایج نشان داده شده که زمان‌بندی و تقسیم و واگذاری

بارها به گره‌های پردازشی، با استفاده از روش DLT به

طور ملموسی نسبت به روش‌های سنتی، کارایی بهینه‌تری

داشته است و این باعث شده که زمان اتمام کار، تا حدود ۸

برابراکاهش یابد. Speedup هم افزایش قابل توجهی داشته

که در حدود ۷ برابر روش‌های سنتی می‌باشد.

۷. منابع

[1] Parhami, Behrooz. Introduction to parallel processing: algorithms and architectures. Springer Science & Business Media, 2006.

[2] Kumar, Mohit, Subhash Chander Sharma, Anubhav Goel, and Santar Pal Singh. "A comprehensive survey for scheduling techniques in cloud computing." Journal of Network and Computer Applications 143 (2019)1-33.

[3] Ghanbari, Shamsollah. "Priority-aware Job Scheduling

59- Latency

- [31] Gupta, Rajni. "Resource provisioning and scheduling techniques of iot based applications in fog computing." *International Journal of Fog Computing (IJFC)* 2, no. 2 (2019): 57-70.
- [32] Mtshali, Mxolisi, Hlabishi Kobo, Sabelo Dlamini, Matthew Adigun, and Pragasen Mudali. "Multi-objective optimization approach for task scheduling in fog computing." In *2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1-6. IEEE, 2019.
- [33] Yin, Luxiu, Juan Luo, and Haibo Luo. "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing." *IEEE Transactions on Industrial Informatics* 14, no. 10 (2018): 4712-4721.
- [34] Islam, Tajul, and M. M. A. Hashem. "Task scheduling for big data management in fog infrastructure." In *2018 21st International Conference of Computer and Information Technology (ICIT)*, pp. 1-6. IEEE, 2018.
- [35] Shi, Haiyan, Wanliang Wang, and Ngaiming Kwok. "Energy dependent divisible load theory for wireless sensor network workload allocation." *Mathematical Problems in Engineering* 2012 (2012).
- [36] Marszałkowski, Jędrzej, Maciej Drozdowski, and Gaurav Singh. "Time-energy trade-offs in processing divisible loads on heterogeneous hierarchical memory systems." *Journal of Parallel and Distributed Computing* 144 (2020): 206-219.
- [37] Wang, Xiaoli, Bharadwaj Veeravalli, and Omer Farooq Rana. "An optimal task-scheduling strategy for large-scale astronomical workloads using in-transit computation model." *International Journal of Computational Intelligence Systems* 11, no. 1 (2018): 600-607.
- [38] Shakarami, Ali, Ali Shahidinejad, and Mostafa Ghobaei-Arani. "An autonomous computation offloading strategy in Mobile Edge Computing: A deep learning-based hybrid approach." *Journal of Network and Computer Applications* 178 (2021): 102974.
- with communication delays using the divisible load paradigm: performance analysis and simulation." *Mathematics and Computers in Simulation* 58, no. 1 (2001): 71-92.
- [18] Ko, Kwangil, and Thomas G. Robertazzi. "Signature search time evaluation in flat file databases." *IEEE Transactions on Aerospace and Electronic Systems* 44, no. 2 (2008): 493-502.
- [19] Hung, Jui Tsun, and Thomas G. Robertazzi. "Switching in sequential tree networks." *IEEE Transactions on Aerospace and Electronic Systems* 40, no. 3 (2004): 968-982.
- [20] Veeravalli, Bharadwaj, and Surendra Ranganath. "Theoretical and experimental study on large size image processing applications using divisible load paradigm on distributed bus networks." *Image and Vision Computing* 20, no. 13-14 (2002): 917-935.
- [21] Bharadwaj, Veeravalli, Xiaolin Li, and Chi Chung Ko. "Efficient partitioning and scheduling of computer vision and image processing data on bus networks using divisible load analysis." *Image and Vision Computing* 18, no. 11 (2000): 919-938.
- [22] Li, Ping, Bharadwaj Veeravalli, and Ashraf A. Kassim. "Design and implementation of parallel video encoding strategies using divisible load analysis." *IEEE Transactions on Circuits and Systems for Video Technology* 15, no. 9 (2005): 1098-1112.
- [23] Robertazzi, Thomas G., and Li Shi. *Networking and Computation: Technology, Modeling and Performance*. Springer Nature, 2020.
- [24] Suresh, Sundaram, Hyoung Joong Kim, Cui Run, and Thomas G. Robertazzi. "Scheduling nonlinear divisible loads in a single level tree network." *The Journal of Supercomputing* 61, no. 3 (2012): 1068-1088.
- [25] Cao, Yang, Fei Wu, and Thomas Robertazzi. "Integrating Amdahl-like Laws and Divisible Load Theory." *arXiv preprint arXiv:1902.01952* (2019).
- [26] Ghanbari, Shamsollah, and Mohamed Othman. "Comprehensive review on divisible load theory: concepts, strategies, and approaches." *Mathematical Problems in Engineering* 2014 (2014).
- [27] Ghanbari, Shamsollah, and Mohamed Othman. "Time cheating in divisible load scheduling: sensitivity analysis, results and open problems." *Procedia Computer Science* 125 (2018): 935-943.
- [28] Ghanbari, Shamsollah, and Mohamed Othman. "A multi-criteria approach to time cheating in the divisible load scheduling." *Iran Journal of Computer Science* (2021): 1-12.
- [29] Johnston, W.E., 1998, July. High-speed, wide area, data intensive computing: A ten year retrospective. In *Proceedings. The Seventh International Symposium on High Performance Distributed Computing* (Cat. No. 98TB100244) (pp. 280-291). IEEE.
- [30] Gokhale, M., Cohen, J., Yoo, A., Miller, W.M., Jacob, A., Ulmer, C. and Pearce, R., 2008. Hardware technologies for high-performance data-intensive computing. *Computer*, 41(4), pp.60-68