

## زمان بندی بارهای غیر خطی در محیط رایانش مه با استفاده از تئوری بار بخش پذیر

سید مجتبی کاظمی<sup>۱</sup>، شمس اله قنبری<sup>۲\*</sup>، منوچهر کاظمی<sup>۳</sup>، محمد عثمان<sup>۴</sup>

<sup>۱</sup> دکتری کامپیوتر، دانشکده مهندسی، دانشگاه آزاد اسلامی واحد قم، قم، ایران

<sup>۲</sup> استادیار دانشکده مهندسی، دانشگاه آزاد اسلامی واحد آشتیان، آشتیان، ایران

<sup>۳</sup> استادیار دانشکده علوم، دانشگاه آزاد اسلامی واحد آشتیان، آشتیان، ایران

<sup>۴</sup> استاد دانشکده تکنولوژی ارتباطات و شبکه، دانشگاه پوترا مالزی (UPM)، کوالالامپور، مالزی

### چکیده

محدودیت سرعت پردازنده‌ها، مانعی برای پردازش بارهایی که دارای داده‌های انبوه<sup>۱</sup> هستند، می باشد. تنها راه افزایش سرعت پردازش و کاهش زمان پردازش نهایی<sup>۲</sup>، 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"<sup>۳</sup> ارائه گردیده و پژوهشهایی در این زمینه، در محیط های مختلف رایانشی به جز رایانش مه، برای بارهای خطی انجام شده است. این مقاله یک زمان بندی بهینه برای بارهای غیرخطی با استفاده از تئوری بار تقسیم پذیر را در محیط مه با ارائه معادلات و حل آنها و الگوریتم مربوطه پیشنهاد می دهد. آزمایشات تجربی نشان داده است که این روش نسبت به روشهای رایج مانند تقسیم مساوی، زمان پردازش نهایی را حدود هشتاد و پنج برابر کاهش می دهد.

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

### Load scheduling using Divisible Load Theory with Non-Linear Loads

Seyed Mojtaba Kazemi<sup>a</sup>, Shamsollah Ghanbari<sup>b,\*</sup>, Manochehr Kazemi<sup>c</sup>, Mohamed Othman<sup>d,e</sup>

<sup>a</sup> Department of Computer Engineering, Qom Branch, Islamic Azad University, Qom, Iran

<sup>b</sup> Department of Computer Engineering, Ashtian Branch, Islamic Azad University, Ashtian, Iran

<sup>c</sup> Department of Mathematics, Ashtian Branch, Islamic Azad University, Ashtian, Iran

<sup>d</sup> Department of Communication Technology and Network, Universiti Putra Malaysia,

۴۳۴۰۰ UPM, Serdang Selangor D.E., Malaysia

### Abstract

Limiting the speed of processors is an obstacle to processing data-intensive loads. The only way to increase the processing speed and reduce the finish time is to use parallel or distributed systems. In this paper, Scheduling is discussed. A new type of scheduling has been presented since ۱۹۸۸ based on "divisible load theory" and research in this field has been carried out in different computing environments except fog computing for non-linear loads. This article proposes optimal scheduling for non-linear loads using divisible load theory (DLT) in the fog environment by presenting equations and solving them and the corresponding algorithm. Experimental tests have shown that this method reduces the finish time about eighty-five times compared to common methods such as equal division.

**Keywords:** Cloud computing, Cloud storage, Main-memory database, Hot/cold data, Cold data management.

#### تاریخچه مقاله:

تاریخ ارسال: ۱۴۰۱/۱۰/۱۷

تاریخ اصلاحات: ۱۴۰۱/۱۱/۲۲

تاریخ پذیرش: ۱۴۰۱/۱۲/۰۳

تاریخ انتشار: ۱۴۰۱/۱۲/۲۸

#### Keywords:

Fog computing  
Divisible Load Theory (DLT)  
Data Intensive Loads  
Distributed Computing

\*ایمیل نویسنده مسئول:

[myrshg@gmail.com](mailto:myrshg@gmail.com)

<sup>۱</sup> Data Intensive Loads

<sup>۲</sup> Finish Time

<sup>۳</sup> Divisible Load Theory -DLT

## ۱ - مقدمه

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 همزمان و به عبارتی استفاده از سیستم های موازی و یا توزیع یافته مانند رایانش توری<sup>۴</sup>، رایانش خوشه<sup>۵</sup>، رایانش ابر و رایانش لبه<sup>۶</sup> و رایانش مه می باشد [۱].

در اینجاست 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<sup>۷</sup> و وظیفه ها<sup>۸</sup> برای پردازش جهت بدست آوردن کمترین زمان انتظار<sup>۹</sup>، زمان پاسخ<sup>۱۰</sup>، و استفاده بهینه از منابع شبکه است. [۲،۳].

آنچه در ادبیات زمان بندی بیشتر رایج است، "زمان بندی منبع<sup>۱۱</sup>" و "زمان بندی وظیفه<sup>۱۲</sup>" و "زمان بندی کار یا برنامه کاربردی<sup>۱۳</sup>" می باشد که در بعضی مقالات از دو زمان بندی آخر به عنوان "زمان بندی بار"<sup>۱۴</sup> نیز نام برده شده است و با آنچه ما از آن به عنوان "زمان بندی بار بخش پذیر دلخواه"<sup>۱۵</sup> نام می بریم، متفاوت است.

از سال ۱۹۸۸ رده جدیدی از زمان بندی در یک مقاله توسط چنگ<sup>۱۶</sup> و ربرتازی<sup>۱۷</sup> [۴] و در مقاله دیگری توسط آگراوال<sup>۱۸</sup> [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<sup>۱۹</sup> بنا نهاده شده است [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 برای بارهای غیر خطی در بستر ترکیبی رایانش ابر و مه می باشد. در این راستا، نمودار زمانی توزیع بار و معادله نحوه محاسبه و تقسیم بار اختصاص یافته، جدیدی ارائه شده است و پس از حل معادلات مرتبط، ارزیابی روش گزارش شده است.

## ۲ - ادبیات موضوع

### ۱-۲ تئوری بار تقسیم پذیر<sup>۲۰</sup>

تئوری بار بخش پذیر یک روش با بنیان قوی ریاضی می باشد و اولین بار بر اساس دو مقاله در ۱۹۸۸ توسط چنگ، ربرتازی و آگراوال ابداع شد [۴،۵]. این تئوری، موازی سازی داده ها را در یک بار شدیداً محاسباتی و داده محور<sup>۲۱</sup>، معین و استخراج می کند و محدودیت های وابستگی سیستمی مانند تأخیرهای ارتباط و ویژگی های پردازنده ها را مد نظر قرار می دهد و برای تقسیم بندی و توزیع بار های شدید محاسباتی به صورت بهینه بین پردازنده ها، در شبکه به کار گرفته می شود. نکته کلیدی اینجاست که ضمن لحاظ تاخیر لینک های ارتباطی، نحوه تقسیم بندی بایستی به گونه ای باشد که پردازش تمام بارها در یک زمان به پایان برسند [۶]. ابتدا چنگ این تئوری را برای شبکه ای از حسگرها پیشنهاد کرد ولی بعدها کاربردهای قابل توجهی در محاسبات بلادرنگ [۱۶]<sup>۲۲</sup>، جبر خطی و ماتریس ها [۱۷]، محاسبات پایگاه های داده [۱۸]، پردازش داده های رادار و حسگرهای مادون قرمز [۱۹]، پردازش تصویر [۲۰]، بینایی ماشین [۲۱]، برنامه های ویدیو و محیط های چندرسانه ای [۲۲]، ۵G، سیستم های مراقبت های بهداشتی، رسانه های اجتماعی، تجارت، دولت، تحقیقات علمی و چندین کاربرد دیگر برای داده های حجیم پیشنهاد شد. در تمام کاربردهای ذکر شده، بایستی حجم عظیمی از

<sup>۱۴</sup> Load Scheduling

<sup>۱۵</sup> Arbitrarily Divisible Load

<sup>۱۶</sup> Cheng

<sup>۱۷</sup> Robertazzi

<sup>۱۸</sup> Agrawal

<sup>۱۹</sup> Divisible Load Theory –DLT-

<sup>۲۰</sup> Divisible Load Theory – DLT-

<sup>۲۱</sup> Data Intensive

<sup>۲۲</sup> Real-Time

<sup>۴</sup> Grid Computing

<sup>۵</sup> Cluster Computing

<sup>۶</sup> Edge Computing

<sup>۷</sup> Jobs

<sup>۸</sup> Tasks

<sup>۹</sup> Waiting Time

<sup>۱۰</sup> Response Time

<sup>۱۱</sup> Resource Scheduling

<sup>۱۲</sup> Task Scheduling

<sup>۱۳</sup> Job/Application Scheduling

زیادی مواجه شده اند. بعضی از آن ها عبارت اند از: استخراج ویژگی و تشخیص لبه در پردازش تصویر، پردازش سیگنال، بینایی ماشین، تشخیص چهره، تشخیص الگو، محاسبات ماتریس ها، محاسبات مالی و محاسبات مهندسی [۱۷].

### ۲-۳- بارهای داده محور<sup>۲۵</sup> و محاسبه محور<sup>۲۶</sup>

رویکردهای پردازش موازی را می توان به طور کلی به دو دسته محاسبات محور و داده محور طبقه بندی کرد [۲۹،۳۰]. داده محور یا شدیداً داده ای، برای توصیف برنامه هایی استفاده می شود که محدود به ورودی/خروجی (I/O) هستند یا نیاز به پردازش حجم زیادی از داده دارند. چنین برنامه هایی بیشتر زمان پردازش خود را به I/O و جابجایی و دستکاری داده ها اختصاص می دهند.

محاسبات محور، برای توصیف برنامه های کاربردی که محدود به محاسبه هستند استفاده می شود. چنین برنامه هایی بیشتر زمان اجرای خود را بر خلاف I/O به نیازمندی های محاسباتی اختصاص می دهند و معمولاً به حجم کمی از داده ها نیاز دارند.

### ۲-۴- بارهای خطی<sup>۲۷</sup> و بارهای غیر خطی<sup>۲۸</sup>

بارهای خطی، بارهای محاسباتی هستند که زمان پردازش آنها یک تابع خطی از اندازه بار  $N$  است. به عنوان مثال تراکنش های انبوه بانک ها و یا شرکت های اعتباری و یا شرکت های بیمه و یا داده کاوی تبلیغات و یا بررسی سوابق مالی برای اتخاذ تصمیم برای سرمایه گذاری و یا داده های انبوه آزمایشگاه های تحقیقاتی و دانشگاه ها برای مطالعات مختلف [۸].

بارهای غیرخطی بارهای محاسباتی با پیچیدگی زمانی چند جمله ای با مرتبه  $n$  هستند و زمان پردازش آنها متناسب با اندازه بار به توان مرتبه چند جمله ای می باشد ( $N^n$ ). بسیاری از الگوریتم های مورد استفاده در مدل سازی و شبیه سازی بلادرنگ سیستم های پیچیده همچون پردازش تصویر نیازمند زمان محاسباتی بار پردازشی با پیچیدگی غیرخطی هستند. به عنوان مثال، تشخیص خط و یا هر شی دیگری همچون دایره و بیضی با استفاده از تبدیل

داده ها پردازش شوند و هدف نهایی، کمینه کردن زمان پردازش بارهای محاسباتی است [۲۵-۲۳، ۶].

DLT در توپولوژی های مختلفی همچون BUS، STAR و MESH و در محاسبات توزیع شده<sup>۲۳</sup> از جمله GRID و CLUSTER و CLOUD مورد استفاده قرار گرفته است [۲۶]. همچنین اخیراً چالش های مختلفی در DLT مد نظر قرار گرفته است که برای اطلاعات بیشتر می توانید به منابع [۲۷،۲۸] مراجعه نمایید.

در مقاله حاضر، با استفاده از قابلیت های موجود در DLT برای زمان بندی در محیط مه راه حلی پیشنهاد شده است. مدل پیشنهادی برای زمانی که داده های ارسال شده به گره های مه<sup>۲۴</sup> به طور دلخواه بخش پذیر و حجیم و انبوه و شدیداً محاسباتی باشند، فوق العاده موثر است. برای مدل پیشنهادی که ساختار درخت یک سطحی دارد، معادله ای طراحی و سپس حل شده و الگوریتم پیشنهاد شده است. نتایج تجربی نشان می دهد که روش DLT می تواند به طور قابل ملاحظه ای کارایی را در زمان بندی مه بالا ببرد.

### ۲-۲- طبقه بندی انواع بار

#### ۲-۲-۱- بارهای غیر قابل تقسیم

این بارها مستقل، غیر قابل تقسیم و با اندازه های مختلف می باشند و باید بطور تمام و کمال در یک پردازنده، پردازش شود. این بارها هیچ رابطه از پیش معینی ندارند.

#### ۲-۲-۲- بارهای قابل تقسیم ماژولار

این بارها به قسمت های کوچک تر و یا task هایی تقسیم می شوند. این بارهای کوچک به عنوان task/subtask یا module نامیده می شوند.

#### ۲-۲-۳- بارهای قابل تقسیم دلخواه

این بارها می توانند به طور دلخواه به هر تعدادی از بخش های کوچک تر تقسیم بندی شوند و هر بخش از بار می تواند مستقلاً پردازش شود. در این مقاله جهت زمان بندی، بارهای به طور دلخواه بخش پذیری را بین چندین پردازنده مدنظر قرار می دهد که روابط از پیش تعریف شده ای ندارند. این چنین بارهایی، با کاربردهای

<sup>۲۱</sup> Compute intensive

<sup>۲۷</sup> Linear Loads

<sup>۲۸</sup> Non Linear Loads

<sup>۲۳</sup> Distributed Computing

<sup>۲۴</sup> Fog Nodes

<sup>۲۵</sup> Data Intensive

زمان بندی بارهای غیر خطی در محیط مه با استفاده از تئوری بار بخش پذیر .

موجود، به گونه ای اختصاص داده است که محاسبات در کمترین زمان ممکن انجام گردد.

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

(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 می کنند، پیشنهاد کرده اند تا به کمترین زمان پردازش دست یابند. آنها بر این باورند که داده های نجومی در مقیاس بزرگ (مانند داده های تولید شده در مرکز نجوم SLOAN آمریکا) نمی توانند بطور مؤثر و بدون سیستم های محاسباتی مبتنی بر شبکه پردازش شوند. در آن مقاله برای تقسیم بهینه بار قابل تقسیم در شبکه های با توپولوژی های مختلف، استفاده از تئوری بار بخش پذیر را برای رسیدن به کوتاهترین زمان پردازش، در سیستم های محاسبات ابری پیشنهاد شده است.

طبق بررسی های انجام شده، استفاده از تئوری بار بخش پذیر تنها در مرجع [۸] در محیط محاسبات لبه تلفن همراه استفاده شده است. در این مرجع فرض شده است که تمام بارها بین دستگاه های لبه تلفن همراه توزیع می شوند، در حالیکه در روش پیشنهادی ما، با توجه به شرایط و اندازه بار، تقسیم بار علاوه بر گره های مه جاری که شامل هر دستگاهی می شود، بین مه های مجاور و ابر نیز امکان پذیر می باشد.

#### ۴- مدل پیشنهادی

##### ۴-۱- شمای کلی سیستم پیشنهادی

توپولوژی مدل سیستم پیشنهادی بر اساس درخت یک سطحی یا ستاره<sup>۳۵</sup> است. در این مدل پردازنده ریشه مجهز به یک پردازنده جلویی یا پیش پردازنده<sup>۳۶</sup> می باشد. پیش

هاف<sup>۲۹</sup> مربوط به شناسایی لبه در تصویر و تشخیص الگو با استفاده از مدل مارکوف پنهان دوبعدی<sup>۳۰</sup>، کمی کردن بردار یادگیری شبکه عصبی، شبیه سازی دینامیک مولکولی سیستم های ماکرومولکولی، و یا داده های انبوه پروژه های برخورد دهنده آزمایشگاه های فیزیک مراکز هسته ای، به تلاش محاسباتی  $O(N^2)$  نیاز دارند [۸،۱۹].

در اکثر الگوریتم هایی که به پیچیدگی محاسباتی غیرخطی نیاز دارند، می توان بارها را به طور دلخواه تقسیم کرد و آنها را به طور مستقل توسط پردازنده های دیگر پردازش کرد، به طوری که کل زمان پردازش کمتر از زمان پردازش در یک پردازنده واحد باشد [۸،۲۰].

### ۳- پیشینه کارهای انجام شده

مقالات متعددی برای زمان بندی کاربرنامه<sup>۳۱</sup> و زمان بندی وظیفه<sup>۳۲</sup> و زمان بندی منابع<sup>۳۳</sup> در محیط مه ارائه شده است [۳۱-۳۴]. همچنین در زمینه زمان بندی بر اساس DLT نیز مقالات اندکی در محیط های مختلف، به استثنای محیط رایانشی مه، ارائه شده است [۳۷-۳۵،۷،۸].

(۲) - در مرجع [۳۴] برای پردازش داده های بزرگ و بخصوص استفاده از سرویسهای بلادرنگ یک مدل سلسله مراتبی شامل حسگرهای اینترنت اشیا در پایین ترین لایه و سپس گره های پردازشی ابتدایی در لبه لایه مه و سپس گره های اصلی پردازشی و لایه بالاتر گره های میانی و خدمت دهنده مه را به جای استفاده از محیط ابر پیشنهاد می دهد. در آن مقاله علاوه بر زمان بندی وظیفه در محیط مه، از الگوریتم Ford-Fulkerson برای تعدیل بار در شبکه ای از گره های مه استفاده شده است.

(۴) - در محیط رایانش لبه تلفن همراه، در مرجع [۸] بر اساس تئوری بارهای تقسیم پذیر، یک مدل نظری برای بخش بندی برنامه کاربردی تقسیم پذیر دلخواه با هدف به حداقل رساندن زمان اتمام برنامه ها، ارائه شده است. این مقاله تصاویر OCR<sup>۳۴</sup> را به عنوان نوعی داده معمولی که قابل تقسیم دلخواه است، برای محیط های محاسباتی متحرک و پویا، مناسب تشخیص داده است و با استفاده از تئوری بار قابل تقسیم، داده ها را به بخش های کوچکتر تقسیم و به واحدهای محاسباتی

<sup>۳۳</sup> Resource Scheduling

<sup>۳۴</sup> Optical Character Reader

<sup>۳۵</sup> Star

<sup>۳۶</sup> front end processor - FEP

<sup>۲۹</sup> Hough

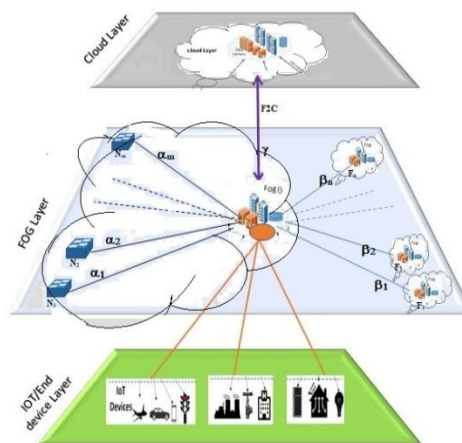
<sup>۳۰</sup> Two Dimension Hidden Markov Model

<sup>۳۱</sup> Job Scheduling

<sup>۳۲</sup> Task Scheduling

و بخش هایی از بار را بین تمام گره های در دسترس و آماده، تقسیم کند.

پردازنده، یک پردازنده تقریبا مستقل است که اجازه می دهد، 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 گره های در دسترس جهت پردازش، ارسال کند. در مدل پیشنهادی، سرعت پردازنده ها متفاوت و سرعت لینک ها در گره های محلی یکسان و در مه های همسایه و ابر، متفاوت می باشند. همچنین انتشار بار در مه محلی به صورت "توزیع ترتیبی بار" و در مه های مجاور و ابر به صورت "همزمان با شروع همزمان" (پردازش در گره های فرزند به محض شروع به دریافت بار، آغاز می شود) می باشد [۲۳]. شمای کلی مدل پیشنهادی در شکل (۱) ارائه شده است.



شکل (۱) شمای کلی مدل

#### ۴-۲- فرموله کردن مسئله

داده ها و رکوردهایی که از طرف حسگرها و دستگاه های اینترنت اشیا از طریق محیط بیسیم و یا با سیم به گره ریشه یک مه محلی می رسد، می تواند یک دستگاه هدایت کننده<sup>۳۷</sup> مجهز به پردازنده باشد. نرم افزار زمان بند<sup>۳۸</sup> در این گره مقیم است. در مدل پیشنهادی گره ریشه مجهز به پردازنده جلویی<sup>۳۹</sup> یا پیش پردازنده است که اجازه می دهد گره ریشه بخشی از بار را خودش پردازش کند و بدون انتظار تکمیل پردازش و یا اشغال بودن، بقیه بخشهای بار را به سایر گره های در دسترس جهت پردازش ارسال کند. از طرفی گره ریشه می تواند با مه های دیگری که همجوار آن هست، و نیز با محیط ابر ارتباط داشته باشد

علامت	نام متغیر	نوع	
$w_i$	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 $i$ ام می نامیم. مثلا عدد ۲ برای این متغیر بیانگر این است که سرعت این پردازنده نصف سرعت پردازنده استاندارد است	مستقل	
$z_i$	زمان صرف شده توسط لینک $L_i$ برای انتقال بار داده شده تقسیم بر زمان صرف شده توسط لینک استاندارد برای انتقال همان بار داده شده. ما آن را نرخ انتقال لینک $i$ ام می نامیم. مثلا عدد ۲ برای این متغیر بیانگر این است که سرعت این لینک نصف سرعت لینک استاندارد است		
$w_{Fj}$	نرخ پردازش $j$ امین مه		
$z_{Fj}$	نرخ انتقال داده لینک به $j$ امین مه		
$w_c$	نرخ پردازش پردازنده ابر		
$z_c$	نرخ انتقال داده لینک گره ریشه به ابر		
$n$	تعداد پردازنده های عمل کننده مه مجاور		
$m$	تعداد پردازنده های عمل کننده مه محلی به غیر از پردازنده ریشه		
$z$	نرخ انتقال داده ها از گره ریشه مه به پردازنده های وابسته محلی یکسان فرض شده است		
$\gamma$	زمان پردازش یک واحد بار در پردازنده استاندارد و زمان انتقال یک واحد بار توسط یک لینک 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 لینک استاندارد		
$Finis h Time$	زمان کل کار از شروع توزیع بار تا پردازش بارها و برگشت نتیجه		وابسته
$\alpha_i$	بخش هایی از بار که به گره های مه جاری تخصیص داده می شود		
$\beta_j$	بخش هایی از بار که به گره های مه های همجوار تخصیص داده می شود		
$\gamma$	بخشی از بار که به ابر تخصیص داده می شود		

<sup>۳۹</sup> Front End Processor -FEP-

<sup>۳۷</sup> Router

<sup>۳۸</sup> Scheduler

$$(\alpha_r T_{cp})^\lambda w_r = (\alpha_r T_{cm})^\lambda z_r + (\alpha_r T_{cp})^\lambda w_r \quad (4-3)$$

...

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

$$(\alpha_r T_{cp})^\lambda w_r = (\gamma T_{cm})^\lambda z_c + (\gamma T_{cp})^\lambda w_c \quad (4-5)$$

$$(\alpha_r T_{cp})^\lambda w_r = (\beta_1 T_{cm})^\lambda z_{F_1} + (\beta_1 T_{cp})^\lambda w_{F_1} \quad (4-6)$$

$$(\alpha_r T_{cp})^\lambda w_r = (\beta_r T_{cm})^\lambda z_{F_r} + (\beta_r T_{cp})^\lambda w_{F_r} \quad (4-7)$$

...

$$(\alpha_r T_{cp})^\lambda w_r = (\beta_n T_{cm})^\lambda z_{F_n} + (\beta_n T_{cp})^\lambda w_{F_n} \quad (4-8)$$

$$\alpha_r + \alpha_1 + \alpha_r + \dots + \alpha_m + \beta_1 + \beta_r + \dots + \beta_n + \gamma = 1 \quad (4-9)$$

$$z_1 = z_r = z_r = \dots = z_m = z \quad \text{and } \lambda \in \mathbb{N} \quad (4-10)$$

#### ۴-۵-۲- مرحله دوم - حل فرم بسته

در مدل غیرخطی حل فرم بسته به شرح زیر است.  
از معادله (۴-۱) داریم:

$$\alpha_r \lambda = \frac{w_1 (T_{cp})^\lambda + z (T_{cm})^\lambda}{w_r (T_{cp})^\lambda} \cdot \alpha_1 \lambda \quad (4-11)$$

اکنون اگر  $K_1 = \frac{w_r (T_{cp})^\lambda}{w_1 (T_{cp})^\lambda + z (T_{cm})^\lambda}$  قرار دهیم آنگاه از معادله (۴-۱۱) داریم:

$$\alpha_r \lambda = \frac{1}{K_1} \cdot \alpha_1 \lambda \quad (4-12)$$

و معادله زیر به دست می آید.

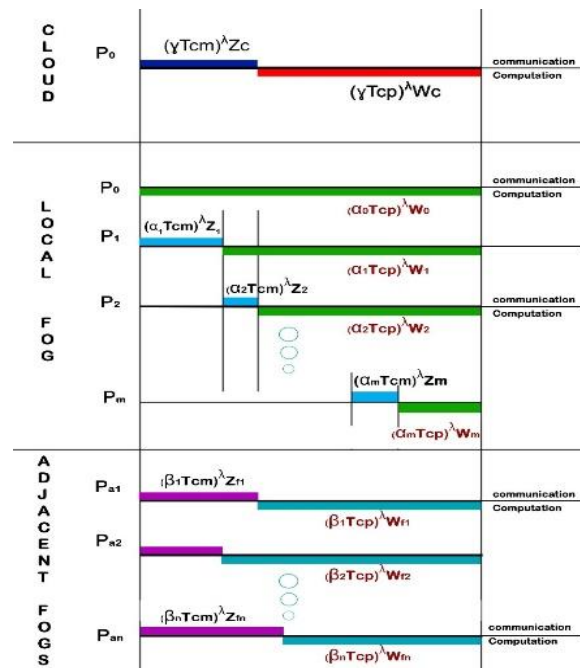
$$\alpha_r = \sqrt[\lambda]{\frac{1}{K_1}} \cdot \alpha_1 \lambda \quad (4-13)$$

#### ۴-۳- معرفی پارامترها و تعریف ها

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

#### ۴-۴- نمودار زمانی مدل پیشنهادی

نمودار زمانی مدل پیشنهادی طبق شکل (۲) می باشد.



شکل (۲) نمودار زمانی توزیع بار و پردازش آنها

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

#### ۴-۵-۵- توصیف ریاضی مدل پیشنهادی

##### ۴-۵-۱- مرحله اول - ایجاد فرم بسته

$$(\alpha_1 T_{cp})^\lambda w_1 = (\alpha_1 T_{cm})^\lambda z_1 + (\alpha_1 T_{cp})^\lambda w_1 \quad (4-1)$$

$$(\alpha_r T_{cp})^\lambda w_r = (\alpha_r T_{cm})^\lambda z_r + (\alpha_r T_{cp})^\lambda w_r \quad (4-2)$$

$$\beta_j^\lambda = \frac{w_j(T_{cp})^\lambda}{w_{F_j}(T_{cp})^\lambda + z_{F_j}(T_{cm})^\lambda} \cdot \alpha_j^\lambda, \quad j = 1, 2, \dots, n \quad (4-24)$$

و توسط معادله (۴-۱۱) و (۴-۲۴) داریم

$$\beta_j^\lambda = \frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_{F_j}(T_{cp})^\lambda + z_{F_j}(T_{cm})^\lambda} \cdot \alpha_1^\lambda, \quad j = 1, 2, \dots, n \quad (4-25)$$

همچنین داریم

$$\beta_j = \sqrt[\lambda]{\frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_{F_j}(T_{cp})^\lambda + z_{F_j}(T_{cm})^\lambda} \cdot \alpha_1^\lambda} \quad (4-26)$$

از آنجاییکه  $\beta_n \geq 0$  و  $\beta_n$  حقیقی است، قرار می دهیم  $k = 0$  و معادله زیر را به دست می آوریم

$$\beta_j = \sqrt[\lambda]{\frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_{F_j}(T_{cp})^\lambda + z_{F_j}(T_{cm})^\lambda} \cdot \alpha_1} \quad j = 1, 2, \dots, n \quad (4-27)$$

بر اساس معادله نرمال سازی (۴-۹) داریم

$$\alpha + \alpha_1 + \alpha_2 + \dots + \alpha_m + \beta_1 + \beta_2 + \dots + \beta_n + \gamma = 1 \quad (4-28)$$

همچنین داریم

$$\alpha_1 \cdot \psi = 1 \quad (4-29)$$

در حالیکه به صورت زیر قرار داده ایم

$$\psi = \left( \sqrt[\lambda]{\frac{1}{\kappa_1}} + 1 + \sum_{i=2}^m \sqrt[\lambda]{\prod_{l=2}^i \kappa_l} \right) + \sqrt[\lambda]{\frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda}} + \sum_{j=1}^n \sqrt[\lambda]{\frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_{F_j}(T_{cp})^\lambda + z_{F_j}(T_{cm})^\lambda}} \quad (4-30)$$

در ادامه به دست می آوریم

$$\alpha_1 = \frac{1}{\psi} \quad (4-31)$$

از معادله (۴-۱۲) داریم

$$\alpha_j = \sqrt[\lambda]{\frac{1}{\kappa_j}} \cdot \alpha_1 = \sqrt[\lambda]{\frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_j(T_{cp})^\lambda}} \cdot \alpha_1 \quad (4-32)$$

بر طبق معادله (۴-۱۷) و معادله (۴-۳۱) مقدار بهینه تقسیمات

بار  $\alpha_i$  را به صورت زیر به دست می آوریم

از آنجاییکه  $\alpha \geq 0$  و  $\alpha$  عددی حقیقی است، بنابراین ما  $k = 0$  قرار می دهیم و به دست می آوریم:

$$\alpha_j = \sqrt[\lambda]{\frac{1}{\kappa_j}} \cdot \alpha_1^\lambda \quad (4-14)$$

سپس داریم

$$\alpha_1 = \sqrt[\lambda]{\kappa_1} \cdot \alpha \quad (4-15)$$

از معادله (۴-۱) تا (۴-۴) داریم:

$$\alpha_i^\lambda = \frac{w_{i-1}(T_{cp})^\lambda}{w_i(T_{cp})^\lambda + z(T_{cm})^\lambda} \cdot \alpha_{i-1}^\lambda, \quad i = 2, 3, \dots, m \quad (4-16)$$

اگر قرار دهیم  $\kappa_i = \frac{w_{i-1}(T_{cp})^\lambda}{w_i(T_{cp})^\lambda + z(T_{cm})^\lambda}$  آنگاه معادله (۴-۱۶) به صورت زیر می شود:

$$\alpha_i^\lambda = \kappa_i \alpha_{i-1}^\lambda \quad (4-17)$$

سپس داریم

$$\alpha_i = \sqrt[\lambda]{\kappa_i} \alpha_{i-1}^\lambda \quad (4-18)$$

از آنجاییکه  $\alpha \geq 0$  و  $\alpha$  عددی حقیقی است، بنابراین ما  $k = 0$  قرار می دهیم و به دست می آوریم:

$$\alpha_i^\lambda = \sqrt[\lambda]{\kappa_i} \cdot \alpha_{i-1}^\lambda = \sqrt[\lambda]{\prod_{l=2}^i \kappa_l} \cdot \alpha_1, \quad i = 1, 2, 3, \dots, m \quad (4-19)$$

از معادله (۴-۱۵) داریم

$$\gamma^\lambda = \frac{w_c(T_{cp})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda} \cdot \alpha^\lambda \quad (4-20)$$

و به وسیله معادله (۴-۱۱) داریم

$$\gamma^\lambda = \frac{w_c(T_{cp})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda} \cdot \alpha^\lambda = \frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda} \cdot \alpha_1^\lambda \quad (4-21)$$

همچنین داریم

$$\gamma = \sqrt[\lambda]{\frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda}} \cdot \alpha_1 \quad (4-22)$$

از آنجاییکه  $\gamma \geq 0$  و  $\gamma$  عددی حقیقی است، بنابراین ما  $k = 0$  قرار می دهیم و به دست می آوریم

$$\gamma = \sqrt[\lambda]{\frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda}} \cdot \alpha_1 \quad (4-23)$$

از معادله (۴-۹) داریم

۱۲:  $j \leftarrow ۲$   
 ۱۳: **while** ( $j \leq n$ ) **do**  
 ۱۴:  $\beta_j \leftarrow \sqrt{\lambda \frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_{F_j}(T_{cp})^\lambda + z_{F_j}(T_{cm})^\lambda}} \cdot \alpha_1$   
 ۱۵: **end while**  
 ۱۶:  $\gamma \leftarrow \sqrt{\lambda \frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda}} \cdot \alpha_1$   
 ۱۷: **Finish Time**  $\leftarrow (w_1 T_{cp}^\lambda + z T_{cm}^\lambda) \cdot \alpha_1^\lambda$   
 ۱۸: **Speedup**  $\leftarrow \frac{T_{p,1}}{T_{p,m+n+1}} = \frac{T_{cp}^\lambda w_1}{(\alpha \cdot T_{cp})^\lambda w_1} = \frac{1}{\alpha^\lambda}$

#### 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 لینک‌های ارتباطی، یک لینک استاندارد با سرعت واحد در نظر گرفته شده و سرعت لینک‌های 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) این مفروضات را نمایش می‌دهد.

#### ۴-۶- تنظیمات پیاده‌سازی

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

$$\alpha_i = \sqrt{\lambda \prod_{l=2}^i \frac{w_{l-1}(T_{cp})^\lambda}{w_l(T_{cp})^\lambda + z(T_{cm})^\lambda}} \cdot \alpha_1 \quad (۴-۳۳)$$

$$\alpha_i = \sqrt{\lambda \prod_{l=2}^i \frac{w_{l-1}(T_{cp})^\lambda}{w_l(T_{cp})^\lambda + z(T_{cm})^\lambda}} \cdot \frac{1}{\psi} \quad (۴-۳۴)$$

زمان پایان<sup>۴</sup> به صورت زیر به دست می‌آید

$$T_{f,m+n+1} = (\alpha \cdot T_{cp})^\lambda w_1 = (w_1 T_{cp}^\lambda + z T_{cm}^\lambda) \cdot \alpha_1^\lambda \quad (۴-۳۵)$$

همچنین ضریب تسریع توسط معادله زیر به دست می‌آید.

$$Speedup_{m+n+1} = \frac{T_{p,1}}{T_{p,m+n+1}} = \frac{T_{cp}^\lambda w_1}{(\alpha \cdot T_{cp})^\lambda w_1} = \frac{1}{\alpha^\lambda} \quad (۴-۳۶)$$

#### ۴-۵-۳- الگوریتم کلی مدل با بار غیر خطی

##### Input:

بار خطی قابل تقسیم که مقدارش یک واحد در نظر گرفته می‌شود و  $\lambda$  توان بار غیر خطی است که دو در نظر گرفته می‌شود

##### Output:

$\alpha_i$  سهم هر یک از پردازنده‌های (گره‌های) مه محلی از یک واحد بار که  $i$  بین ۰ تا  $m$  می‌باشد.  
 $\beta_j$  سهم هر یک از مه مجاور از یک واحد بار از یک واحد بار مه  $j$  بین ۱ تا  $n$  می‌باشد.  
 $\gamma$  سهم کلود از یک واحد بار می‌باشد.

۱:  $i \leftarrow ۱$   
 ۲: **while** ( $i \leq m$ ) **do**  
 ۳:  $\kappa_i \leftarrow \frac{w_1(T_{cp})^\lambda}{w_i(T_{cp})^\lambda + z(T_{cm})^\lambda}$   
 ۴: **end while**  
 ۵:  $\Psi \leftarrow \left( \sqrt{\lambda \frac{1}{\kappa_1}} + 1 + \sum_{i=2}^m \left( \sqrt{\lambda \prod_{l=2}^i \kappa_l} \right) + \sqrt{\lambda \frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_c(T_{cp})^\lambda + z_c(T_{cm})^\lambda}} + \sum_{j=1}^n \sqrt{\lambda \frac{w_1(T_{cp})^\lambda + z(T_{cm})^\lambda}{w_{F_j}(T_{cp})^\lambda + z_{F_j}(T_{cm})^\lambda}} \right)$   
 ۶:  $\alpha_1 \leftarrow \frac{1}{\Psi}$   
 ۷:  $\alpha_i \leftarrow \alpha_1 \cdot \sqrt{\lambda \kappa_i}$   
 ۸:  $i \leftarrow ۲$   
 ۹: **while** ( $i \leq m$ ) **do**  
 ۱۰:  $\alpha_i \leftarrow \kappa_i \cdot \alpha_{i-1}$   
 ۱۱: **end while**

<sup>۴</sup> Finish Time

جدول (۳) جدول مفروضات سرعت های 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. مثلا 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 می باشد
۳-۴	$z_i$	زمان صرف شده توسط لینک $L_i$ برای انتقال بار داده شده تقسیم بر زمان صرف شده توسط لینک استاندارد برای انتقال همان بار. (نرخ انتقال لینک $i$ ام) عدد ۴ نمایانگر این است که سرعت لینک یک چهارم سرعت لینک استاندارد است). این عدد نرخ انتقال در مه محلی بین گره ریشه و گره $i$ می باشد.
۱،۵۰-۲،۵۰	$w_{Fi}$	نرخ پردازش $i$ امین مه ( $i=0$ ) به مه محلی اشاره می کند)
۳-۴	$Z_{Fi}$	نرخ انتقال داده لینک 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 لینک 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 لینک استاندارد
۴-۲۲	$m$	تعداد پردازنده های عمل کننده در مه محلی به غیر از پردازنده ریشه
۴-۲۲	$n$	تعداد مه های مجاور

## ۵-۱- روش پیاده سازی

به دلیل بعضی محدودیت های شبیه سازیهای موجود همچون iFogsim، برای پیاده سازی از زبان پایتون استفاده شده است. برای نشان دادن کارایی روش پیشنهادی، نتایج استفاده از تئوری بار بخش پذیر با دو نوع زمان بندی رایج دیگر مقایسه گردیده است. این دو نوع عبارتند از:

الف: تقسیم بار به صورت مساوی بین تمام گره های آماده پردازش

ب: تقسیم بار به صورت تصادفی بین تمام گره های آماده پردازش

در این حالت این مقادیر تصادفی به عنوان اندازه وظیفه ها<sup>۴۱</sup> در نظر گرفته می شود و سپس واگذاری این وظیفه ها بین تمام گره های آماده پردازش براساس واگذاری بار بیشتر به پردازنده قویتر که این الگوریتم برگرفته از الگوریتم Best Fit-BF می باشد.

نتایج شبیه سازی شده نشان می دهد که زمان اتمام کار در الگوریتم پیشنهادی ما در مقایسه با سایرین، در محیط مه، در حد یک هشناد و پنجم می باشد.

## ۵-۲- نتایج پیاده سازی

تمامی آزمایشات و مقایسات، برای بررسی اثر نوع زمان بندی و تعداد گره های آماده پردازش، بر "زمان پایان" کار و افزایش سرعت محاسبه انجام می گردد. تعداد گره های مه محلی ثابت و تعداد مه های مجاور متغیر در نظر گرفته می شود. برای بررسی بیشتر، گره های ثابت در هر حالت، برای اعداد ثابت ۴، ۸، ۱۲ شبیه سازی شده اند.

### ۵-۲-۱- تقسیم بار براساس روش تئوری بار

#### تقسیم پذیر ۴۲ با بار غیر خطی

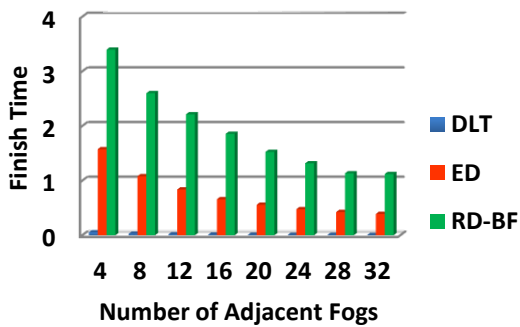
در شکل ۳ نمودار زمان اتمام یک واحد کار را که بر اساس تئوری بار تقسیم پذیر (DLT) در مدل غیر خطی به دست آمده است را نشان می دهد.

<sup>۴۲</sup> Divisible Load Theory (DLT)

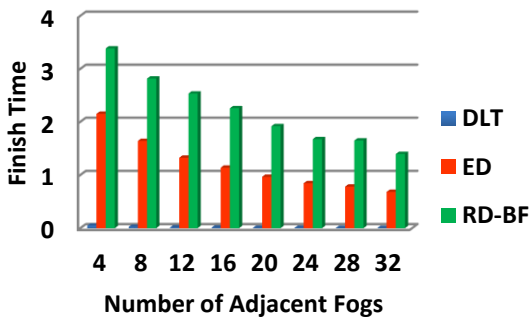
<sup>۴۱</sup> Tasks

### ۵-۲-۲- مقایسه زمان پایان<sup>۴۳</sup> بین روشها در مدل غیر خطی

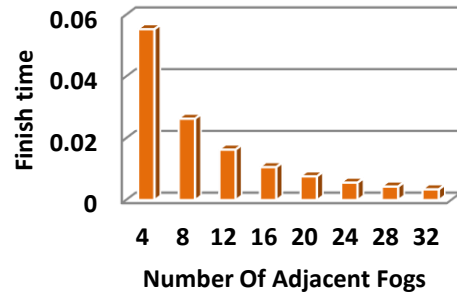
شکل ۴ نمودار مقایسه ای زمان پایان را بین سه روش "تقسیم مساوی (ED)" و "تقسیم تصادفی و واگذاری کار بزرگتر به پردازنده قویتر (RD-BF)" و "تئوری بار تقسیم پذیر<sup>۴۴</sup> (DLT)" با بار غیر خطی نشان می دهد.



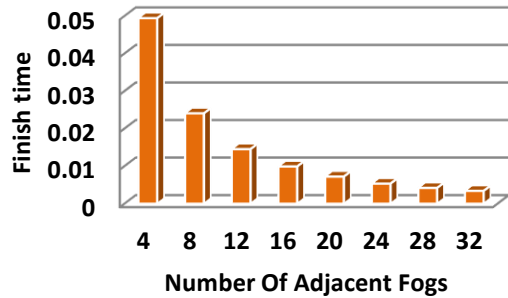
a- DLT Non Linear <<Local Fog Nodes=۴>>



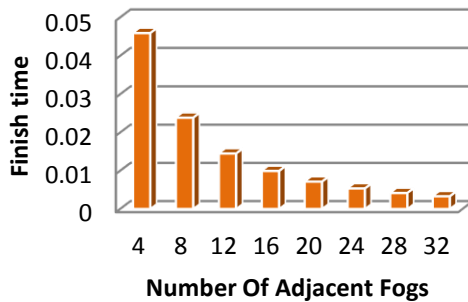
b- DLT Non Linear <<Local Fog Nodes=۸>>



a- DLT Non Linear <<Local Fog Nodes=۴>>



b- DLT Non Linear <<Local Fog Nodes=۸>>



c- DLT Non Linear <<Local Fog Nodes=۱۲>>

شکل ۳ نمودار زمان اتمام کار بر اساس روش DLT با بار غیر خطی

<sup>۴۴</sup> Divisible Load Theory (DLT)

<sup>۴۳</sup> Finish Time

Computing: A Multi-criteria Approach." *Azerbaijan Journal of High-Performance Computing* ۲, no. ۱ (۲۰۱۹): ۲۹-۳۸.

[۴] Cheng, Yuan-Chieh and Thomas G. Robertazzi. "Distributed computation with communication delay (distributed intelligent sensor networks)." *IEEE transactions on aerospace and electronic systems* ۲۴, no. ۶ (۱۹۸۸): ۷۰۰-۷۱۲.

[۵] Agrawal, Rakesh, and H. V. Jagadish. "Partitioning techniques for large-grained parallelism." *IEEE Transactions on Computers* ۳۷, no. ۱۲ (۱۹۸۸): ۱۶۲۷-۱۶۳۴.

[۶] Bharadwaj, Veeravalli, Debasish Ghose, Venkataraman Mani, and Thomas G. Robertazzi. *Scheduling divisible loads in parallel and distributed systems. Vol. ۸. John Wiley & Sons, ۱۹۹۶.*

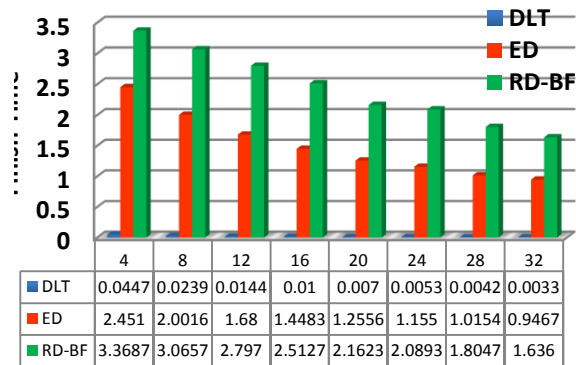
[۷] Lin, Xuan, Anwar Mamat, Ying Lu, Jitender Deogun, and Steve Goddard. "Real-time scheduling of divisible loads in cluster computing environments." *Journal of Parallel and Distributed Computing* ۷۰, no. ۳ (۲۰۱۰): ۲۹۶۶-۳۰۸.

[۸] 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* ۱۰, no. ۵ (۲۰۱۸): ۱۶۱۱.

[۹] Iorga, Michaela, Larry Feldman, Robert Barton, Michael J. Martin, Nedim S. Goren, and Charif Mahmoudi. "Fog computing conceptual model." (۲۰۱۸).

[۱۰] Mahmud, Redowan, Ramamohanarao Kotagiri, and Rajkumar Buyya. "Fog computing: A taxonomy, survey and future directions." In *Internet of everything*, pp. ۱۰۳-۱۳۰. Springer, Singapore, ۲۰۱۸.

[۱۱] Cisco, "Cisco global cloud index: Forecast and methodology, ۲۰۱۵-۲۰۲۰," Cisco, ۲۰۱۵.



c- DLT Non Linear <<Local Fog Nodes=۱۲>>

شکل ۴ مقایسه زمان اتمام کار بین سه روش DLT, ED, RD-BF با بار غیرخطی

## ۶- نتیجه گیری و خلاصه

در این مقاله الگوریتمی برای کاهش تاخیر<sup>۴۵</sup> 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 آزمایشات مختلف به دست آمده است. در بخش نتایج نشان داده شده که زمان بندی و تقسیم و واگذاری بارها به گره های پردازشی، با استفاده از روش DLT به طور ملموسی نسبت به روشهای سنتی، کارایی بهینه تری داشته است و این باعث شده که زمان اتمام کار، تا حدود ۸۵ برابر کاهش یابد.

## منابع

- [۱] Parhami, Behrooz. *Introduction to parallel processing: algorithms and architectures. Springer Science & Business Media, ۲۰۰۶.*
- [۲] 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* ۱۴۳ (۲۰۱۹): ۱-۳۳.
- [۳] Ghanbari, Shamsollah. "Priority-aware Job Scheduling Algorithm in Cloud

<sup>۴۵</sup> Latency

- Aerospace and Electronic Systems* ۴۰, no. ۳ (۲۰۰۴): ۹۶۸-۹۸۲.
- [۲۰] 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* ۲۰, no. ۱۳-۱۴ (۲۰۰۲): ۹۱۷-۹۳۰.
- [۲۱] 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* ۱۸, no. ۱۱ (۲۰۰۰): ۹۱۹-۹۳۸.
- [۲۲] 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* ۱۰, no. ۹ (۲۰۰۰): ۱۰۹۸-۱۱۱۲.
- [۲۳] Robertazzi, Thomas G., and Li Shi. *Networking and Computation: Technology, Modeling and Performance*. Springer Nature, ۲۰۲۰.
- [۲۴] Suresh, Sundaram, Hyoun Joong Kim, Cui Run, and Thomas G. Robertazzi. "Scheduling nonlinear divisible loads in a single level tree network." *The Journal of Supercomputing* ۶۱, no. ۳ (۲۰۱۲): ۱۰۶۸-۱۰۸۸.
- [۲۵] Cao, Yang, Fei Wu, and Thomas Robertazzi. "Integrating Amdahl-like Laws and Divisible Load Theory." *arXiv preprint arXiv: ۱۹۰۲.۰۱۹۰۲* (۲۰۱۹).
- [۲۶] Ghanbari, Shamsollah, and Mohamed Othman. "Comprehensive review on divisible load theory: concepts, strategies, and approaches." *Mathematical Problems in Engineering* ۲۰۱۴ (۲۰۱۴).
- [۲۷] Ghanbari, Shamsollah, and Mohamed Othman. "Time cheating in divisible load scheduling: sensitivity analysis, results and open problems." *Procedia Computer Science* ۱۲۰ (۲۰۱۸): ۹۳۰-۹۴۳.
- [۱۲] 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* ۲۰, no. ۱ (۲۰۱۷): ۴۱۶-۴۶۴.
- [۱۳] Li, Guangshun, Yuncui Liu, Junhua Wu, Dandan Lin, and Shuaishuai Zhao. "Methods of resource scheduling based on optimized fuzzy clustering in fog computing." *Sensors* ۱۹, no. ۹ (۲۰۱۹): ۲۱۲۲.
- [۱۴] 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*. ۲۰۱۸.
- [۱۵] Haddad, Emile. "Runtime reallocation of divisible load under processor execution deadlines." In *Proceedings of Third Workshop on Parallel and Distributed Real-Time Systems*, pp. ۳۰-۳۱. IEEE, ۱۹۹۰.
- [۱۶] Chuprat, Suriayati. "Divisible load scheduling of real-time task on heterogeneous clusters." In ۲۰۱۰ *International Symposium on Information Technology*, vol. ۲, pp. ۷۲۱-۷۲۶. IEEE, ۲۰۱۰.
- [۱۷] Chan, S. K., Veeravalli Bharadwaj, and Debasish Ghose. "Large matrix-vector products on distributed bus networks with communication delays using the divisible load paradigm: performance analysis and simulation." *Mathematics and Computers in Simulation* ۵۸, no. ۱ (۲۰۰۱): ۷۱-۹۲.
- [۱۸] Ko, Kwangil, and Thomas G. Robertazzi. "Signature search time evaluation in flat file databases." *IEEE Transactions on Aerospace and Electronic Systems* ۴۴, no. ۲ (۲۰۰۸): ۴۹۳-۵۰۲.
- [۱۹] Hung, Jui Tsun, and Thomas G. Robertazzi. "Switching in sequential tree networks." *IEEE Transactions on*

- [۳۶] 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* ۱۴۴ (۲۰۲۰): ۲۰۶-۲۱۹.
- [۳۷] 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* ۱۱, no. ۱ (۲۰۱۸): ۶۰۰-۶۰۷.
- [۳۸] 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* ۱۷۸ (۲۰۲۱): ۱۰۲۹۷۴.
- [۲۸] Ghanbari, Shamsollah, and Mohamed Othman. "A multi-criteria approach to time cheating in the divisible load scheduling." *Iran Journal of Computer Science* (۲۰۲۱): ۱-۱۲.
- [۲۹] Johnston, W.E., ۱۹۹۸, July. High-speed, wide area, data intensive computing: A ten years retrospective. In *Proceedings. The Seventh International Symposium on High Performance Distributed Computing* (Cat. No. ۹۸TB۱۰۰۲۴۴) (pp. ۲۸۰-۲۹۱). IEEE.
- [۳۰] Gokhale, M., Cohen, J., Yoo, A., Miller, W.M., Jacob, A., Ulmer, C. and Pearce, R., ۲۰۰۸. Hardware technologies for high-performance data-intensive computing. *Computer*, ۴۱(۴), pp. ۶۰-۶۸
- [۳۱] Gupta, Rajni. "Resource provisioning and scheduling techniques of iot based applications in fog computing." *International Journal of Fog Computing (IJFC)* ۲, no. ۲ (۲۰۱۹): ۵۷-۷۰.
- [۳۲] Mtshali, Mxolisi, Hlabishi Kobo, Sabelo Dlamini, Matthew Adigun, and Pragasen Mudali. "Multi-objective optimization approach for task scheduling in fog computing." In ۲۰۱۹ *International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, pp. ۱-۶. IEEE, ۲۰۱۹.
- [۳۳] 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* ۱۴, no. ۱۰ (۲۰۱۸): ۴۷۱۲-۴۷۲۱..
- [۳۴] Islam, Tajul, and M. M. A. Hashem. "Task scheduling for big data management in fog infrastructure." In ۲۰۱۸ ۲۱st *International Conference of Computer and Information Technology (ICCIT)*, pp. ۱-۶. IEEE, ۲۰۱۸.
- [۳۵] Shi, Haiyan, Wanliang Wang, and Ngaiming Kwok. "Energy dependent divisible load theory for wireless sensor network workload allocation." *Mathematical Problems in Engineering* ۲۰۱۲(۲۰۱۲).

زمان بندی بارهای غیر خطی در محیط مه با استفاده از تئوری بار بخش پذیر.



منوچهر کاظمی - استادیار گروه ریاضی کاربردی، دانشگاه آزاد اسلامی، واحد آشتیان، است. او دارای مدرک لیسانس است از دانشگاه صنعتی خواجه نصیر طوسی (KNTU)، (۱۳۷۷)، کارشناسی ارشد از موسسه مطالعات تکمیلی علوم پایه زنجان (۱۳۸۰) و مدرک دکتری ریاضی کاربردی در حوزه تحلیل عددی از دانشگاه آزاد اسلامی واحد کرج، (۱۳۹۴) می باشد. علایق اصلی وی عبارتند از در زمینه حل عددی معادلات انتگرال، حساب کسری، حل پذیری معادلات انتگرال و علوم کامپیوتر.

نشانه رایانامه ایشان عبارتند از:

Univer\_ka@yahoo.com



سیدمجتبی کاظمی فارغ التحصیل دکتری در رشته مهندسی کامپیوتر گرایش نرم افزار از دانشگاه آزاد اسلامی قم می باشد. وی مدرک کارشناسی ارشد خود را در رشته مهندسی نرم افزار از دانشگاه آزاد اسلامی واحد تهران جنوب (۱۳۷۶) و کارشناسی خود را در رشته سخت افزار کامپیوتر از دانشگاه تهران (۱۳۶۳) دریافت کرد. علایق تحقیقاتی وی شامل برنامه ریزی، زمان بندی بار تقسیم پذیر، محاسبات مه، پردازش موازی و محاسبات توزیع شده است. نشانه رایانامه ایشان عبارتند از:

mojkaz@gmail.com



محمد عثمان دکترای خود را از دانشگاه Kebangsan مالزی دریافت کرده است. او در حال حاضر استاد علوم کامپیوتر در گروه فناوری ارتباطات و شبکه، دانشگاه پوترا مالزی (UPM) است. او پیش از آن معاون مرکز توسعه اطلاعات و ارتباطات بود و در آنجا مسئولیت پردیس شبکه UMPNet، پروژه ارتباطات بی سیم uSport و مرکز داده UPM را بر عهده داشت. او همچنین محقق و هماهنگ کننده ماشین های پرسرعت با آزمایشگاه محاسبات است. نشانه رایانامه ایشان عبارتند از:

mothman@upm.edu.my



شمس اله قنبری، استادیار گروه علوم کامپیوتر، دانشگاه آزاد اسلامی، واحد آشتیان، است. او مدرک دکترای خود را در محاسبات و سیستم های توزیع شده از دانشگاه پوترا مالزی (UPM) در سال ۱۳۹۴ دریافت کرد. وی در سال ۲۰۱۹ و ۲۰۲۳ در دانشگاه دولتی قزاقستان جنوبی در رشته سیستم های توزیع شده (اصول و کاربردها) بعنوان استاد مدعو تدریس نموده است. برخی از علایق تحقیقاتی اخیر وی عبارتند از محاسبات توزیع شده، محاسبات ابری، محاسبات مه، نظریه زمان بندی، نظریه بار تقسیم پذیر. نشانه رایانامه ایشان عبارتند از:

myrshg@gmail.com

روش ارجاع به مقاله : س م. کاظمی - ش. قنبری - م. کاظمی - م. عثمان. زمان بندی بارهای غیر خطی در محیط مه با استفاده از تئوری بار بخش پذیر. فصلنامه محاسبات و سامانه‌های توزیع شده، سال پنجم، شماره ۲، شماره پیاپی ۱۰، صفحه ۱۲۶ تا ۱۳۹، سال ۱۴۰۱

How to cite: Seyed Mojtaba Kazemi, sh. Ghanbary, M. Kazemi, M Othman. "Load scheduling using Divisible Load Theory with Non-Linear Loads, Journal of Distributed Computing and Systems (JDACS), Vol ۵, Issue ۲, Page ۱۲۶-۱۳۹, ۲۰۲۳.