باسمه تعالی

پایان‌نامه کارشناسی ارشد

الگوریتم ترکیبی M–بعدی برای زمانبندی جریان کاری علمی در محاسبات ابری

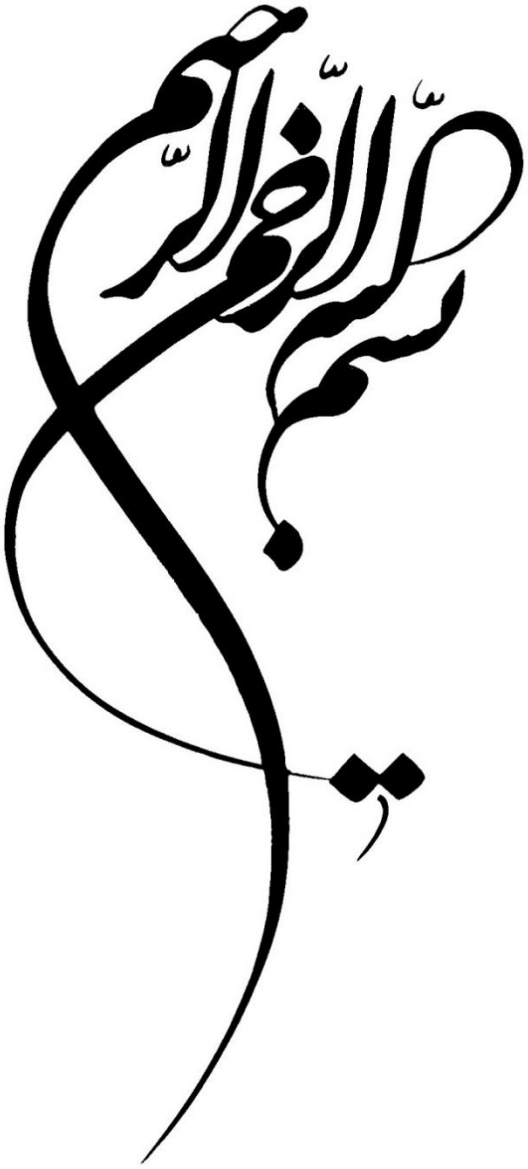
تهیه و تنظیم:

زهراء الحمداوی

استاد راهنما:

دکتر سید مجید مزینانی

بهمن ۹۸


بسم الله الرحمن الرحیم

تقدیر و تشکر

سپاس مخصوص خداوند مهربان که به انسان توانایی و دانایی بخشید تا به بندگانش شفقت ورزد، مهربانی کند و در حل مشکلاتشان یاری‌شان نماید. از راحت خویش بگذرد و آسایش هم نوعان را مقدم دارد، با او معامله کند و در این خلوص انباز نگیرد و خوش باشد که پروردگار سمیع و بصیر است.

لذا اکنون‌که در سایه‌سار بنده‌نوازی‌هایش پایان‌نامه حاضر به انجام رسیده است بر خود لازم می‌دانم تا مراتب سپاس را از بزرگوارانی به‌جا آورم که اگر دست یاری گرشان نبود، هرگز این پایان‌نامه به انجام نمی‌رسید.

ابتدا از استاد گران‌قدرم جناب آقای دکتر سید مجید مزینانی که زحمت راهنمایی این پایان‌نامه را بر عهده داشتند، کمال سپاس را دارم.

سپاس آخر را به خانواده‌ی عزیزم تقدیم می‌کنم که حضورشان در فضای زندگی‌ام مصداق بی‌ریای سخاوت بوده است.

چکیده

محاسبات ابری در زمینه زمان‌بندی جریان کار به ویژه برای جریان کار علمی با محبوبیت رو به رشدی مواجه شده است. با استفاده از این محاسبات ابری در حال ظهور، می‌توان از منابع نامحدود مجازی با حداقل سرمایه بهره مند شد. یکی از فرآیندهای چالش برانگیز سیستم مدیریت جریان کار (WfMS) در محیط محاسبات ابری، زمان‌بندی کارهای موجود در برنامه کاربردی جریان کاری علمی (SWFA) روی منابع محاسباتی موجود و در عین حال بهینه‌سازی هزینه اجرای آن است. چندین رویکرد برای بهینه‌سازی هزینه به منظور بهبود جنبه‌های اقتصادی SWFS در محیط ابری ارائه شده اند. هدف این پایان نامه، ارائه یک الگوریتم M بعدی ترکیبی است که یک الگوریتم فرا اکتشافی همانند: فوق اکتشافی مشتق زمانی تکاملی (CTDHH)، الگوریتم زمان‌بندی ترکیبی با هزینه بهینه (HCHS)، بهینه سازی ازدحام ذرات (PSO) و الگوریتم ژنتیک (GA) را به همراه الگوریتم‌های اکتشافی مانند: الگوریتم‌های IC-PCPD2 و IC-Loss، مورد استفاده قرار می‌دهد. در پایان الگوریتم پیشنهادی ما با الگوریتم پایه از لحاظ زمان اجرا و هزینه مقایسه شده است و نتایج نشان دهنده‌ی بهبود جوابها می باشد. الگوریتم پیشنهادی، برای زمان‌بندی جریان‌های کاری ابری پایدارتر است و به نظر می‌رسد با احتمال بیشتری زمانبندی‌های قابل قبولی را تولید کند.

کلمات کلیدی: جریان کاری، جریان کاری علمی، محاسبات ابری، زمان‌بندی، بهینه‌سازی هزینه، فرا اکتشافی، اکتشافی.

فهرست مطالب

فصل اول: مقدمه ۹

۱-۱- مقدمه ۱۰

۱-۲- تعریف مسأله ۱۱

۱-۳- چالش‌های موجود ۱۵

۱-۴- اهداف تحقیق ۱۵

۱-۵- ساختار پایان‌نامه ۱۵

فصل دوم: مفاهیم اولیه ۱۶

۲-۱- مقدمه ۱۷

۲-۲- محاسبات ابری ۱۸

۲-۳- ویژگی های ضروری ۲۲

۲-۴- مزایای کلیدی محاسبات ابری ۲۴

۲-۵- لایه های معماری محاسبات ابری ۲۷

۲-۶- مدل کارگزاری ۲۹

۲-۷- چالش های محاسبات ابری ۳۰

۲-۸- جریان های کار ۳۴

۲-۹- مفهوم زمان بندی ۳۷

۲-۱۰- زمان بندی جریان های کاری ۳۸

۲-۱۱- زمان بندی جریان کار در ابر ۳۹

۲-۱۲- اهداف زمان بندی جریان کار ۴۰

۲-۱۳- استراتژی بهینه سازی ۴۲

فصل سوم: کارهای پیشین ۴۶

۳-۱- تکنیک های مبتنی بر اکتشافی ۴۷

۳-۲- تکنیک های مبتنی بر فرا اکتشافی ۵۱

فصل چهارم: راهکار پیشنهادی **۵۷**
۴-۱- مقدمه ۵۸
۴-۲- الگوریتم پیشنهادی ۵۹
۴-۲-۱- جمعیت اولیه با استفاده از الگوریتم IC-PCPD ۶۲
۴-۲-۲- یک جمعیت اولیه با استفاده از الگوریتم IC-LOSS ۶۳
۴-۲-۳- رویکرد فرا اکتشاف مشتق شده از زمان تکمیل CTDHH ۶۳
۴-۲-۴- رویکرد ترکیبی زمانبندی ترکیبی با هزینه موثر HCHS ۶۵
۴-۲-۵- رویکردی الگوریتم ژنتیک(GA) ۶۵
۴-۲-۶- رویکرد بهینه سازی مبتنی بر ازدحام ذرات (PSO) ۶۶
فصل پنجم: ارزیابی نتایج ۶۷
۵-۱- محیط شبیه سازی ۶۸
۵-۲- تنظیمات آزمایش ۶۸
۵-۳- ارزیابی نتایج ۶۹
فصل ششم: نتیجه گیری ۷۵
۶-۱- نتیجه گیری ۷۶
۶-۲- کار آینده ۷۷
مراجع ۷۸

فهرست اشکال

شکل ۲-۱- طبقه بندی محاسبات توزیع شده [۶] ۱۸
شکل ۲-۲- قش کارگزار در زمان بندی ابر[۱۴] ۲۲
شکل ۲-۳- پشته محاسبات ابری [۱۷] ۲۷
شکل ۲-۴- نمایش جریان کاری در گراف [۱۴] ۳۵
شکل ۲-۵- انواع الگوریتم های زمان بندی[۱۱] ۳۷
شکل ۲-۶- نمایش جریان کاری مبتنی بر DAG ۳۹
شکل ۲-۷- جریان کاری [۲۴] ۳۹
شکل ۲-۸- انواع استراتژی های بهینه سازی ۴۵
شکل ۴-۱- فلوچارت روش پیشنهادی ۶۰
شکل ۴-۲- الگوریتم پیشنهادی ۶۲
شکل ۵-۱- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری inspiral) ۷۰
شکل ۵-۲- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری Epigenomics) ۷۱
شکل ۵-۳- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری Montage) ۷۲
شکل ۵-۴- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری Cybershake) ۷۳
شکل ۵-۵- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری SIPHT) ۷۴

فهرست جداول

جدول ۳-۱- خلاصه تکنیک های مبتنی بر اکتشافی ۵۰
جدول ۳-۲- خلاصه از تکنیک های مبتنی بر فرا اکتشافی ۵۵
جدول ۵-۱- کلاسهای جریان کاری ۶۹
جدول ۵-۲- نتایج آماری زمان تکمیل برای جریان کاری inspiral ۷۰
جدول ۵-۳- نتایج آماری زمان تکمیل برای جریان کاری Epigenomics ۷۱
جدول ۵-۴- نتایج آماری زمان تکمیل برای جریان کاری Montage ۷۱
جدول ۵-۵- نتایج آماری زمان تکمیل برای جریان کاری Cybershake ۷۲
جدول ۵-۶- نتایج آماری زمان تکمیل برای جریان کاری SIPHT ۷۳

فصل اول

مقدمه

۱-۱- مقدمه

در آینده سیستم ناهمگن به وسیله شناور ساختن منابع ناهمگن، به عنوان زیرساختی جهانی برای برنامه‌های کاربردی مبتنی بر محاسبات الکترونیک خواهد شد. سیستم ناهمگن شامل سیستم‌های ابری، سیستم‌های گرید، خوشه‌ها و غیره می‌باشد. برای انجام آزمایش‌های پیچیده، در حالی که بر عملیات جریان کار روی سیستم های ناهمگن نظارت می‌شود، منابع توزیع شده همانند وسایل محاسباتی، ابزارهای علمی و برنامه های کاربردی بایستی سازمان‌دهی شوند [۱]. محاسبات ابری یک پارادایم محاسباتی جدید است که برای حل مسائل جدید و به منظور اجرای برخی کارهای بزرگ، کامپیوترهای مختلف را با هدف تشکیل یک سیستم محاسباتی بزرگ ترکیب می‌کند. یکی از چالش‌های اصلی در محاسبات ابری زمان‌بندی کار و استفاده بهینه از منابع ابری است. اساساً، زمان‌بندی کار یک مساله NP سخت است و به دست آوردن راه‌حل بهینه برای آن، کاری چالش برانگیز است. مساله زمان‌بندی به علت ناهمگونی، پویایی و طبیعت خودکنترلی محیط ابری، کاری پیچیده است. زیرساخت به عنوان سرویس (IaaS)[1]،چارچوب به عنوان سرویس (PaaS)[2] نرم افزار به عنوان سرویس (SaaS)[3] مدل‌های پیاده‌سازی ابر هستند که به کاربران اجازه می‌دهند برنامه‌های کاربردی خودشان را تحت سیاست "پرداخت به ازای استفاده"، روی ابر اجرا کنند. با توجه به در نظر گرفتن بحث هزینه در استفاده از ابر، کاربران مجبور به استفاده بهینه از ابر هستند [۲]. جریان کار، متداولترین مدل برای نمایش برنامه‌های کاربردی است و به طور گسترده در زمینه محاسبات علمی از قبیل: فیزیک، بیوانفورماتیک و اخترشناسی مورد استفاده قرار گرفته است. کارها در جریان کار علمی، متمرکز بر داده یا متمرکز بر محاسبه هستند و لذا جهت فراهم نمودن منابع محاسباتی، نیازمند محیط محاسباتی با کارایی بالا هستند. محاسبات

[1] Infrastructure as a Service
[2] Platform as a Service
[3] Software as a Service

ابری یک چارچوب محاسباتی توزیع شده بزرگ مقیاس است که در آن منابع محاسباتی بر حسب تقاضای برنامه‌های کاربردی مبتنی بر جریان کار، در دسترس قرار می گیرند [۳]. NP کامل بودن زمان‌بندی جریان کار در ابر، امری شناخته شده است و زمان‌بندی جریان کار در محیط چند ابری حتی از آن هم پیچیده‌تر است. این بدان علت است که در محیط‌های چند ابری، سرویس‌ها از طریق چند چارچوب IaaS ابری مستقل ارائه می‌شوند و منابع محاسباتی به صورت یک یا چند سرویس ترکیبی تجمیع خواهند شد. بدین ترتیب انتخاب ترکیب بهینه‌ای از سرویس‌ها از میان چند چارچوب IaaS که نیازهای QoS[1] را برآورده سازند، مساله‌ای چالش برانگیز است. اجرای برنامه‌های کاربردی مبتنی بر جریان کار روی سیستم‌های ناهمگن مزایای زیر را ارائه می‌دهد [۱]:

۱- امکان ساخت برنامه‌های کاربردی پویا با استفاده از منابع ناهمگن.

۲- استفاده کامل از منابع که در دامنه خاصی رخ می‌دهد و بازده و هزینه اجرا را بهبود می‌دهد.

۳- بازه های اجرای دامنه‌های مدیریتی مختلف جهت دستیابی به عملکرد پردازشی خاص.

۲-۱- تعریف مسأله

زمان‌بندی یکی از چالش‌های اولیه برای توازن بار کاری، روی یک ابر ترکیبی است. استراتژی‌های زمان‌بندی مختلف ممکن است نرخ استفاده از منبع، زمان پاسخ، قابلیت اطمینان، هزینه عملیات و هزینه نگهداری را تغییر دهند. مساله زمان‌بندی بهینه، یک مساله NP کامل است. هیچ رویکرد زمان‌بندی که بتواند به یک راهکار بهینه با زمان چندجمله ای به ویژه در زمینه زمان‌بندی کارهای بزرگ دست یابد، ارائه نشده است. کاربران می‌توانند برای اجرای کارها از منابع محاسباتی موجود، به شیوه‌ای کارآمد استفاده کنند. در هر حال به علت افزایش چشم‌گیر پیچیدگی و اندازه برنامه‌های کاربردی امروزی، منابع محاسباتی محدود فعلی پاسخگوی نیازهای کاربران نیست. در نتیجه کاربران

[1] Quality of Service

نیاز به یک محیط محاسباتی مناسب دارند تا فضای ذخیره و منابع محاسباتی لازم برای پردازش برنامه کاربردی پیچیده و بزرگ مقیاس را فراهم کند. از آن جا که دو عامل اصلی در محاسبات ابری حضور دارند، در بهینه‌سازی نیز دو جنبه مد نظر خواهد بود: بهینه‌سازی انجام شده توسط فراهم‌کنندگان و بهینه‌سازی انجام شده توسط کاربران. بهینه‌سازی هزینه انجام شده توسط فراهم کننده ابر، عمدتاً روی حداقل کردن هزینه نگهداری مرکز داده فیزیکی تمرکز دارد. معمولاً این نوع تقلیل هزینه به وسیله کاهش مصرف برق انجام می‌شود. در محاسبات ابری، چالش بهینه‌سازی هزینه SWFS[1]، یک مساله چندمنظوره آگاه از هزینه است که نیاز به بررسی سه جنبه اصلی دارد: (i) کاربران مختلف که برای منابع درون محاسبات ابری یا گرید رقابت می‌کنند تا محدودیت‌های QoS را ارضا کنند، (ii) وابستگی‌های متقابل بین وظایف در جریان کار، (iii) هزینه ارتباطات بالا به علت وابستگی‌های متقابل بین وظایف. در هر حال با در نظر گرفتن تمامی جنبه‌های مربوط به مساله بهینه‌سازی، فرآیند SWFS پیچیده‌تر شده و از نظر زمانی نیز به منابع محاسباتی بیشتری نیاز دارد. در بهینه‌سازی مبتنی بر اکتشاف، چندین رویکرد متفاوت برای جمعیت اولیه پیشنهاد شده‌اند. برخی مطالعات به بررسی تاثیر مولدهای اعداد تصادفی به عنوان بخش مهمی از الگوریتم جست و جوی تصادفی، اختصاص یافته‌اند. برخی محققان برای تولید یکنواخت مفردات در فضای مساله، به بررسی تاثیر مولدهای اعداد تصادفی مختلف و تاثیر آن‌ها روی کارایی کلی فرا اکتشاف‌ها پرداخته‌اند. محققان دریافتند که توجه به جمعیت اولیه می‌تواند روی موفقیت الگوریتم‌های فرا اکتشافی تاثیرگذار باشد. ایده ارائه شده در [۴]، استفاده از برخی الگوریتم‌های فرا اکتشافی و انتخاب یکی از آن‌ها برای اجرا در هر مرحله است. یکی از مسائل موجود در [۴]، مرتبه زمانی بالای الگوریتم به علت عدم توجه به جمعیت اولیه در الگوریتم‌هاست. ایده ارائه شده برای رفع این مساله، استفاده از الگوریتم‌های اکتشافی برای بهبود جمعیت اولیه در این

[1] Scientific Workflow Scheduling

الگوریتم‌ها به منظور دستیابی به نتایج بهتر و یک مرتبه زمانی کوتاهتر است. الگوریتم‌های اکتشافی وابسته به مساله بوده و سعی می‌کنند تا راه‌حل را با استفاده از خصیصه‌های مساله به طور کامل پیدا کنند. راهکار آن‌ها مبتنی بر یادگیری و اکتشاف است که بدین وسیله یک جست و جوی جامع و علمی برای یافتن پاسخ بهینه و سرعت بخشی به فرآیند پاسخ به کار گرفته می‌شود. در هر صورت این الگوریتم‌ها بسیار حریصانه بوده و اغلب در بهینه‌های محلی گیر می‌افتند. علاوه بر این ممکن است در دستیابی به راه‌حل بهینه سراسری، شکست بخورند. الگوریتم‌های اکتشافی [۵] که می‌توانند برای بهبود جمعیت پایه مورد استفاده قرار گیرند می‌توانند به صورت زیر باشند:

- الگوریتم ناهمگن زودترین زمان اتمام (HEFT[1]) که دو اکتشاف جدید برای زمان‌بندی کار ارائه می‌کند. اولین اکتشاف برای گرف جهتدار بدون حلقه (DAG[2]) و دیگری برای زمان بندی کارهای مستقل. به منظور کاهش makespan، هر دو اکتشاف رفتار عملکردی مناسبی دارند.
- استراتژی زمان بندی QoS محدود شده چندگانه از جریان‌های کاری چندگانه (MQMW[3])، سه آیتم را ارائه می‌دهد که عبارتند از: پیش پردازنده، زمان‌بند و اجراکننده. پیش پردازنده، زمان و هزینه جریان کار را محاسبه و سپس کارهای آماده را ارسال می‌کند. زمان‌بند، همه کارها را صف کرده و سپس همه آن‌ها را بر اساس استراتژی آنها مرتب می‌کند. در نهایت، اجرا کننده بهترین سرویس را برای اجرای پیوسته کارها انتخاب می‌کند. در پایان اجرا کننده به پیش پردازنده اطلاع رسانی می‌کند تا وضعت تکمیل گردد.
- الگوریتم زمان‌بندی آگاه از منبع (RASA[4])، مزایای Max-Min وMinMin را ترکیب کرده و معایب آن ها را رفع نموده است.

[1] Heterogeneous-Earliest-Finish-Time
[2] directed acyclic graph
[3] Multiple QoS Constrained Scheduling Strategy of Multi-Workflows
[4] Resource-Aware-Scheduling algorithm

- یک الگوریتم توافق زمان – هزینه (CTC[1])، از ویژگی‌های ابر برای تنظیم جریان‌های کاری هزینه محدود متمرکز بر نمونه، از طریق دستیابی به یک توافق سریع بین هزینه و زمان اجرا با ورودی کاربر، استفاده می‌کند.
- زمان‌بندی جریان کار بهینه (OWS[2])، روی زمان‌بندی جریان‌های کاری ابر تمرکز می‌کند. در ابتدا الگوریتم اکتشاف، همه منابع را لیست می‌کند و سپس، جریان کار را با در نظر گرفتن پارامترهای QoS (زمان اجرا، ابلیت اطمینان، هزینه پولی و...)، به منابع آزاد تخصیص می‌دهد.
- الگوریتم ناهمگن زمان اتمام زودتر مقیاس پذیر (SHEFT[3])، بعد از آن که یک گراف وزن دار و جهتدار بدون حلقه برای نمایش یک جریان کار در ابر مدل‌سازی شد، تمامی منابع را به تعدادی خوشه طبقه‌بندی می‌کند. برنامه خطی عدد صحیح (ILP[4])، دو سطح از توافق سطح خدمات (SLA[5]) دارد که کاربر و فراهم کننده را مد نظر قرار می‌دهد. اول علامت‌گذاری و مدلسازی مساله و دوم فرموله سازی برنامه خطی عدد صحیح.
- مسیرهای نیمه بحرانی در ابر SaaS (SC-PCP[6])، مسیرهای نیمه بحرانی که در جریان‌کار زمان‌بندی شده قبل خاتمه یافته‌اند را به صورت بازگشتی، زمان بندی می‌کند.
- زمان‌بندی متقارن جریان کار مبتنی بر سطح (CLWS[7])، در حین اجرای واقعی، تمامی کارها را بر اساس توافق بین کارها، در نظر می‌گیرد.

[1] compromised-Time-Cost
[2] Optimal Workflow Scheduling
[3] Scalable-Heterogeneous-Earliest-Finish- Time
[4] Integer linear program
[5] Service Level Agreements
[6] SaaS Cloud-Partial Critical Paths
[7] Concurrent Level based Workflow Scheduling

- IC-PCP پیشرفته با تکرار (EIPR[1])، به منظور کاهش اختلاف کارایی منابع ابر با در نظر گرفتن مهلت‌های زمانی جریان کار، هزینه و زمان اتلاف شده‌ی جریان کار را مورد استفاده قرار می‌دهد.

۱-۳- چالش‌های موجود

- زمان بندی چند منظوره.
- دوره‌های زمانی یک ساعته.
- ابهام و عدم قابلیت اطمینان در زمان‌بندی جریان کار.
- رویکرد مدیریت منابع مبتنی بر سرویس.
- استراتژی هزینه مبتنی بر مقدار منابع استفاده شده.

۱-۴- اهداف تحقیق

- بهبود جمعیت اولیه الگوریتم تکاملی با هدف افزایش سرعت اجرا و همگرایی سریعتر الگوریتم.

۱-۵- ساختار پایان‌نامه

سازمان‌دهی مطالب پایان‌نامه بدین شرح است که در فصل دوم ابتدا بیان مفاهیم اولیه را شرح می‌دهیم. در فصل سوم کارهای پیشین را مرور می‌کنیم در فصل چهارم راهکار پیشنهادی را بصورت کامل شرح می‌دهیم و در فصل پنجم نتایج کار را مورد ارزیابی قرار می‌دهیم و نشان می‌دهیم. در فصل ششم نتیجه گیری ارائه می‌شود.

[1] Enhanced IC-PCP with Replication

فصل دوم

مفاهیم اولیه

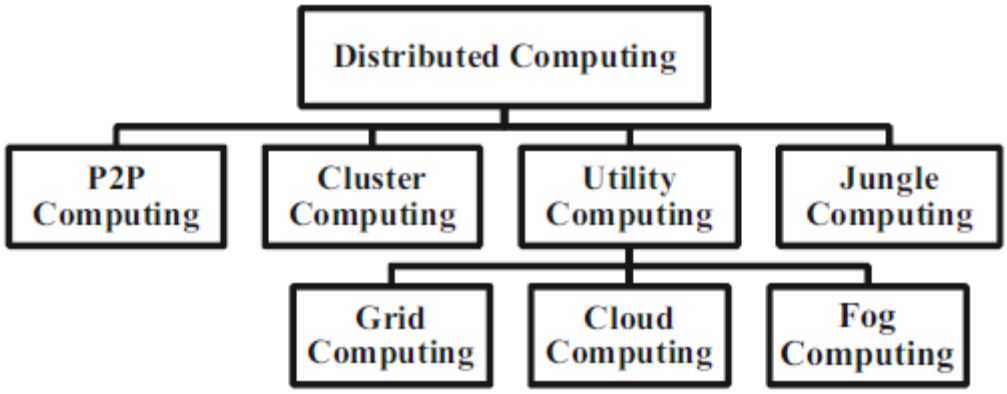
۱-۲- مقدمه

معرفی شبکه‌های کامپیوتری در دهه ۱۹۷۰ منجر به توسعه سیستم‌های توزیع‌شده شد. یک سیستم توزیع‌شده مجموعه‌ای از کامپیوترهای مستقل است که برای کاربر به صورت یک سیستم واحد ظاهر می‌شوند و یک دید سیستمی واحد را ارائه می‌کند. تجمیع هماهنگ این کامپیوترهای توزیع شده، امکان دسترسی به قدرت محاسباتی بزرگی را فراهم می‌کند. تاکنون تکنولوژی‌های کمی در زمینه سیستم‌های توزیع شده ظهور داشته‌اند. شبکه نظیر به نظیر (P2P[1])، یکی از نمونه‌های اولیه از سیستم‌های توزیع شده است. در هر حال سیستم محاسبات توزیع شده است، یکی از مهمترین دسته‌های سیستم‌های توزیع‌شده است که برای کارهای محاسباتی با کارایی بالا مورد استفاده قرار می‌گیرد. در همین راستا، با استفاده از کامپیوترهای شخصی قدرتمندتر و کم هزینه‌تر به همراه شبکه‌های پر سرعت، محاسبات خوشه‌ای به شدت متداول شده است. محاسبات گرید و محاسبات ابری، پارادایم‌های محاسبات توزیع‌شده دیگری هستند که با تکامل اینترنت به ترتیب در اواسط دهه ۱۹۷۰ و ۲۰۰۷، پدیدار شده‌اند. محاسبات ابری در سال‌های اخیر یکی از داغ‌ترین تکنولوژی‌ها بوده است [۶].

اما با توجه به چرخه‌ی، هایپ گارتنر برای ظهور تکنولوژی‌ها، ۲۰۱۳، محاسبات ابری از قله انتظارات عبور کرده و با ۲ تا ۵ سال فاصله از دوره بلوغش به سمت شیب سرخوردگی هدایت شده است. بنابراین تمایل در سیستم‌های توزیع‌شده، تغییر به سمت استفاده از پارادیم‌های محاسباتی جدیدتر است. محاسبات جنگلی به عنوان پارادایم جدیدی روی صحنه آمده‌اند تا با استفاده از محاسبات توزیع‌شده بسیار غیر یکنواخت و متنوع، به کارایی بهتری دست پیدا کنند. محاسبات مه نیز در سال ۲۰۱۲، پارادایم محاسبات ابری را به سمت لبه شبکه توسعه داده تا نوع جدیدی از برنامه‌های کاربردی و

[1] Peer-to-peer

سرویس‌ها را فعال کند. در شکل ۲-۱ طبقه بندی پارادایم های محاسبات توزیع شده نشان داده شده است.



شکل ۲-۱- طبقه بندی محاسبات توزیع شده [۶]

۲-۲- محاسبات ابری

محاسبات ابری، هم به برنامه های کاربردی که به عنوان سرویس روی اینترنت ارائه می شوند و هم به سخت افزار و نرم افزارهای سیستمی موجود روی دیتاسنترهایی که آن سرویس را ارائه می کنند، اشاره دارد. یک خوشه از نرم افزار و سخت افزارهای سیستمی که سرویس هایی را به عموم (احتمالاً در ازای بها) ارائه می کند یک" ابر عمومی" را می سازد. بنابراین محاسبات به عنوان یک کاربرد مفید همانند آب، برق، گاز و... ارائه می شود که فقط در ازای مقدار استفاده شده پول پرداخت می شود. به عنوان مثال ابر توسعه پذیر آمازون، چارچوب Azure مایکروسافت و App Engine گوگل و Salesforce نمونه هایی از ابرهای عمومی هستند که امروزه ارائه شده اند. به هر حال محاسبات ابری شامل "ابر خصوصی" که اشاره به دیتاسنترهای داخل یک سازمان دارد، نمی شود. بنابراین محاسبات ابری می تواند به صورت تجمیع محاسبات به عنوان یک سودمندی و نرم افزار به عنوان سرویس تعریف شود. مجازی سازی منابع یک نیاز کلیدی برای ارائه دهندگان ابر است که به وسیله مالتی پلکس آماری فراهم می شود و به

منظور مقیاس پذیر کردن ابر و همچنین ایجاد توهم منابع نامحدود برای کاربران ابر، مورد نیاز است. در نظر داشته باشید که، محاسبات سودمند مختلفی ارائه شده، بر اساس سطح انتزاع ارائه شده به برنامه نویس و سطح مدیریت منابع، متمایز می شوند. به عنوان نمونه ای از ارائه دهندگان ابر موجود، یک نمونه از EC2 آمازون، بسیار شبیه یک ماشین فیزیکی است و تقریبا کنترل کاملی از پشته نرم افزار را به همراه یک API[1] باریک در اختیار کاربر قرار می دهد. این امر به کاربر انعطاف زیادی را در کدنویسی می دهد اما در عین حال بدان معناست که آمازون تنظیمات مقیاس پذیری اتوماتیک و تحمل شکست کمی دارد. در مقابل، موتور App گوگل، API را سمت کاربر اجرا می کند اما تنظیمات مقیاس پذیری اتوماتیک و تحمل شکست موثری را ارائه می دهد. چارچوب Azure مایکروسافت چیزی بین دو مورد فوق است و برخی امکانات انتخاب زبان و مقیاس پذیری اتوماتیک و تحمل شکست را به کاربر ارائه می دهد. هر کدام از ارائه دهندگان ذکر شده،، تنظیمات متفاوتی برای مجازی سازی محاسبات، ذخیره و ارتباطات دارند [۷]. محاسبات ابری سرویس ها، منابع اشتراکی یا زیرساخت مشترک را از طریق اینترنت و برحسب تقاضا ارائه می دهد. یک ارائه دهنده سرویس، این تسهیلات را ارائه می دهد و در ازای چیزی که کاربر استفاده کرده هزینه دریافت می کند که به آن پرداخت در ازای استفاده گفته می شود [۸]. مشتری می تواند از فضای ذخیره سازی، قابلیت های پردازشی، سرورها، سیستم عامل و محیط های توسعه برنامه کاربردی استفاده کند. کاربر می تواند در یک نمونه(به طور زمانی) مقیاس منابع را بر حسب تقاضا در ابر، کوچکتر یا بزرگتر کند [۹]. همچنین انعطاف لازم برای دسترسی به منابع از طریق وسایل مختلف فراهم می شود. در ابر، کاربران با استفاده از مجازی سازی منابع، می توانند برنامه های کاربردی خودشان را مدیریت کرده، توسعه داده یا مستقر کنند. چندین محیط محاسبات ابری وجود دارند، اما اصولا به سه دسته طبقه بندی می شوند که در ادامه شرح داده می شود [۱۰].

[1] application programming interface

ابر خصوصی (ابر داخلی): این نوع ابرها به علت ارائه مزایای خاص به سازمان ها، درون سازمان قرار دارند. سرمایه گذاری اولیه در ساخت، مدیریت و خرید ابرهای خصوصی وجود دارد. این نوع ابرها، میانگین بهره وری سرور را بهبود می بخشند.

ابر عمومی: این نوع ابر، محیطی برای عموم است. اشخاص ثالث یا ارائه دهندگان، این نوع ابر را مدیریت کرده و سرویس ها را به مشتریان ارائه می کنند. این نوع ابر در مدیریت، توسعه و استقرار برنامه های کاربردی به طور گسترده مورد استفاده قرار گرفته است. ابرهای عمومی بسیار مقیاس پذیر و قابل اطمینان هستند اما بحث امنیت یکی از نگرانی های اصلی در این نوع ابر است.

ابر ترکیبی: این نوع ابر ترکیبی از محیط های ابر خصوصی و عمومی است. نمونه هایی از این ابر در ادامه آورده شده است:

- ابر محاسباتی توسعه پذیر آمازون(EC2): ظرفیت محاسباتی متغیر(چرخه های Cpu) را به کاربران ارائه می دهد [۱۱].
- سرویس ذخیره سازی ساده آمازون (۳S): تسهیلات بازیابی و مدیریت حجم زیادی از داده ها را در هر لحظه و از هر جا به وسیله اینترنت فراهم می کند. این سرویس مبتنی بر اجاره ارائه می شود [۱۱].
- سرویس های CRM توسط salesforce.com ارائه می شوند و قادر به مدیریت اطلاعات کاربر بدون نصب هیچ نرم افزار خاصی هستند.

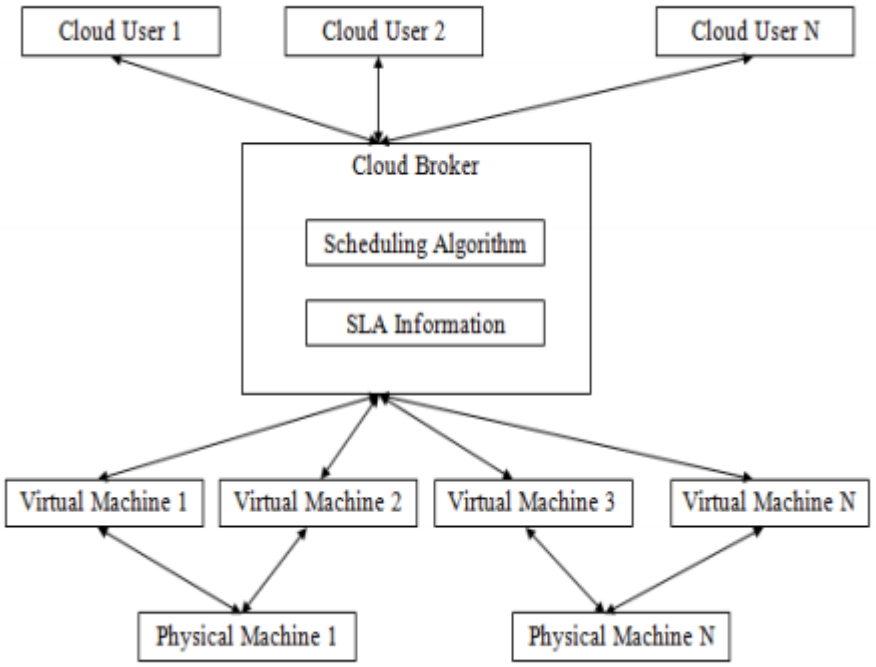
برخی از ویژگی های محاسبات ابری عبارتند از [۱۲].

- واسط برنامه نویسی برنامه کاربردی (APIs): به منظور دسترسی به سرویس های ابر ارائه شده است.

- قابلیت اطمینان و دسترس پذیری: احتمال شکست زیرساخت حداقل شده و لذا بسیار قابل اطمینان و در دسترس هستند.
- اشتراک گذاری چندگانه: چندین کاربر به وسیله به اشتراک گذاشتن زیرساخت و کاهش هزینه می توانند به صورت کارآمدتری کار کنند.
- مقیاس پذیری: منابع بر حسب نیازمندهای تجاری کاهش یا افزایش می یابند.

در ابر چهار موجودیت اصلی وجود دارند. کاربر، کارگزار، ماشین های مجازی و ماشین های فیزیکی [۱۳]. کاربران ابر، مصرف کنندگان اصلی ابر هستند و می توانند درخواست های سرویس خود را از همه جای جهان ارسال کنند. یک دیتاسنتر، از ماشین های فیزیکی تشکیل شده است. با استفاده از تکنولوژی مجازی سازی، ماشین های مجازی مافوق ماشین های فیزیکی شکل می گیرند. کارگزار به عنوان واسطی بین کاربران ابر و دیتاسنترهای ابر عمل می کند. کارگزار مسئول تخصیص منابع ابر به برنامه های کاربردی مبتنی بر جریان کار کاربر است.کارگزار با استفاده از الگوریتم زمان بندی و SLA[1] (توافق سطح خدمات)، ماشین های مجازی را به برنامه های کاربردی مبتنی بر جریان کار تخصیص می دهد. SLA یک توافق مستند بین ارائه دهنده سرویس و کاربر است. شکل ۲-۲ بیان کننده نقش یک کارگزار ابر در یک محیط ابری است.

[1] Service Level Agreements



شکل ۲-۲- قش کارگزار در زمان بندی ابر[۱۴]

الگوریتم های زمان بندی منافعی را برای کاربران ابر و ارائه دهندگان سرویس فراهم می کنند. از یک سو، الگوریتم های زمان بندی می توانند به شیوه ای طراحی شوند که محدودیت های QoS (کیفیت خدمات) کاربران ابر را ارضا کنند و از سوی دیگر می توانند به گونه ای طراحی شوند که توازن بار را بین ماشین های مجازی برقرار کنند و در نتیجه منابع را در سمت ارائه دهندگان سرویس بهبود بخشند.

۲-۳- ویژگی های ضروری

- **سرویس خودگردان بر مبنای تقاضا**

یک مصرف کننده می تواند به طور یک طرفه قابلیت های محاسباتی مانند زمان سرور و فضای ذخیره ی شبکه را بر حسب نیاز به طور اتوماتیک و بدون نیاز به تعامل انسان با هر ارائه دهندگان سرویس، تدارک ببیند.

- **دسترسی شبکه گسترده**

قابلیت ها روی شبکه موجودند و از طریق مکانیزم های استاندارد در دسترس قرار می گیرند که به وسیله چارچوب های کلاینت ناهمگن نازک یا ضخیم به عنوان مثال تلفن های موبایل، تبلت ها، لپ تاپ ها و ایستگاه های کاری، مورد استفاده قرار می گیرند.

- **واکشی منابع**

منابع محاسباتی ارائه دهنده واکشی شده تا با استفاده از یک مدل چند مستاجری به چندین مصرف کننده خدمات ارائه دهند و با منابع فیزیکی و مجازی مختلفی که به طور پویا بر اساس تقاضای مصرف کننده تخصیص داده یا بازتخصیص می شوند. یک حس استقلال مکانی وجود دارد که در آن مصرف کننده به طورکلی هیچ دانش یا کنترلی روی مکان دقیق منابع ارائه شده ندارد اما ممکن است قادر به تشخیص مکان در سطح بالاتری از انتزاع (مانند کشور، ناحیه یا مرکز داده) باشد. نمونه هایی از منابع عبارتند از : فضای ذخیره، پردازش، حافظه و پهنای باند شبکه.

- **توسعه پذیری سریع**

قابلیت ها می توانند به صورت توسعه پذیر و در برخی موارد به صورت اتوماتیک فراهم و آزاد شوند تا بدینوسیله بر حسب تقاضا به سمت درون یا بیرون تغییر مقیاس دهند. قابلیت های موجود، در سمت مصرف کننده اغلب به نظر نامحدود می رسند و می توانند در هر اندازه و در هر زمان تخصیص داده شوند.

- **سرویس اندازه گیری شده**

سرویس های ابری با استفاده از قابلیت های اندازه گیری متناسب با نوع سرویس (به عنوان مثال فضای ذخیره، پردازش، پهنای باند و حسابهای کاربری فعال) در برخی سطوح انتزاع به صورت اتوماتیک مصرف

منابع را کنترل و بهینه سازی می کنند. مصرف منبع می تواند کنترل، نظارت و گزارش شود و شفافیت را برای ارائه کننده و مصرف کننده سرویس مورد استفاده، فراهم کند [۱۵].

۴-۲- مزایای کلیدی محاسبات ابری

بسیاری از ایده های اولیه در محاسبات ابری جدید نیستند (در واقع به سال ۱۹۶۵ بر می گردد که اتحادیه غربی نقش آتی شرکت را به عنوان "سودمندی اطلاعاتی" به عنوان بخشی از طرح های استراتژیک شرکت در سطح ملی تصویر کرد) که منجر شد تا ناظرین متعددی مانند لری الیسون مدیرعامل اوراکل کل مفهوم را یک محصول ناامید کننده اعلام کند. با وجود اینکه در واقع برخی از ایده های فوق برای مدت طولانی وجود داشتند، با این حال استدلال می کنیم که همپوشانی آن ها در محیط امروزی که اطلاعات، مستقل از وسلیه و مکان، قابل دسترسی هستند، تغییر اساسی در محاسباتی که می شناختیم ایجاد کرده است. محاسبات ابری به طور خاص مزایای کلیدی زیر را ارائه کرده است [۱۶]:

۱. هزینه ورود را برای شرکت های کوچکی که در تلاش برای بهره برداری از تحلیل های تجاری متمرکز بر محاسبه هستند، کاهش می دهد که تاکنون فقط برای شرکت های بزرگ در دسترس بوده است. این تمرین های محاسباتی.، معمولا قدرت محاسباتی بالا را برای مدت زمان کمی در بر می گیرند و محاسبات ابری چنین تدارکات پویایی از منابع را فراهم می سازد. محاسبات ابری همچنین فرصت بزرگی را برای بسیاری از کشورهای جهان سوم که تاکنون در انقلاب IT[1] عقب مانده بودند، فراهم می سازد. همان طور که بعدا بحث می کنیم، ارائه کنندگان محاسبات ابری از مزایای چارچوب ابری برای فعال کردن سرویس های IT در کشورهایی که به طور سنتی فاقد منابع لازم برای توسعه گسترده سرویس های IT هستند، بهره برداری می کنند.

[1] Information Technology

۲. بدون نیاز به سرمایه اولیه، تقریباً دسترسی سریعتری را به منابع سخت افزاری برای کاربران فراهم می کند که در نتیجه زمان ارائه به بازار را برای بسیاری از کسب و کارها سرعت می بخشد. با در نظر گرفتن IT به عنوان هزینه عملیاتی (در زبان صنعت 'Op-ex' متضاد 'Cap-ex' در نظر گرفته می شود) باعث کاهش چشمگیر هزینه های پیش رو در محاسبات شرکت، می گردد. به عنوان مثال، بسیاری از استارت آپ های نوید بخش مبتنی بر اینترنت جدید، مانند 37 Signals، Jungle Disk، Gigavox و SmugMug با سرمایه گذاری در فناوری اطلاعات محقق شده اند که تنها کمی با تعداد مورد نیاز در سال های قبل فاصله دارد. ابر یک زیرساخت انطباق پذیر است که می تواند توسط کاربران مختلف به اشتراک گذاشته شود و هر کدام از آن ها ممکن است آن را به روش متفاوتی به کار گیرند. کاربران کاملا از هم جدا شده اند و انعطاف پذیری زیرساخت موجب می شود تا همان طور که کاربران به سیستم متصل می شوند، بار محاسباتی به سرعت متوازن شود (فرآیند تنظیمات زیرساخت به گونه ای استاندارد شده که اضافه شدن قابلیت های محاسباتی به سادگی افزودن یک واحد ساختاری به گرید موجود است). زیبایی کار در آن است که با افزایش تعداد کاربران، بار تقاضا شده روی سیستم به صورت اتفاقی متوازن می شود و حتی هنگامی که مقیاس اقتصادی آن توسعه می یابد.

۳. محاسبات ابری می تواند موانع IT برای نوآوری را کاهش دهد به طوری که این امر می تواند به وسیله بسیاری از استارت آپ های نوید بخش، از برنامه های کاربردی آنلاین گسترده همانند فیسبوک و یوتیوب گرفته تا برنامه های کاربردی متمرکزتر مانند TripIt (برای مدیریت سفر فرد) یا Mint (برای مدیریت امور مالی فرد)، اثبات شود.

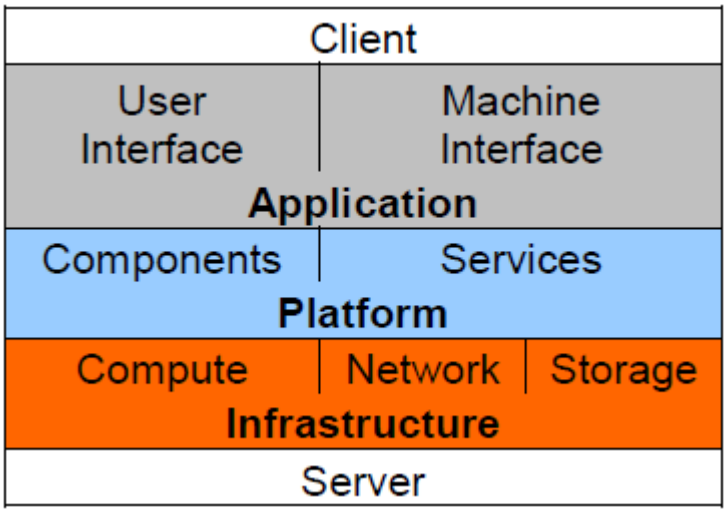
۴. محاسبات ابری باعث می شود که شرکت ها بتوانند سرویس های خود را، که به طور فزاینده ای به دقت اطلاعات وابسته می شوند، برحسب تقاضای مشتریان توسعه پذیر کنند. از آن جا که منابع

محاسباتی از طریق نرم افزار مدیریت می شوند، هنگامی که نیازمندی های جدیدی به وجود می آیند می توانند به سرعت به کار گرفته شوند. در حقیقت هدف محاسبات ابری، کاهش یا افزایش مقیاس منابع از طریق API های نرم افزاری بر اساس بار کلاینت با کمترین تعامل با ارائه دهنده سرویس است.

۵. محاسبات ابری همچنین دسته جدیدی از برنامه های کاربردی را امکان پذیر ساخته و سرویس هایی که قبلا امکان پذیر نبودند را ارائه می کند. نمونه های آن عبارتند از (a) برنامه های کاربردی وتعاملی موبایل که آگاه از زمینه، محیط و مکان هستند و به صورت بلادرنگ به اطلاعات ارائه شده توسط کاربران انسانی، حسگرهای غیرانسانی (به عنوان مثال حسگرهای رطوبت و فشار داخل کانتینرهای حمل و نقل دریایی) یا حتی سرویس های اطلاعاتی مستقل (به عنوان مثال داده های آب و هوایی جهانی) پاسخ می دهند. (b) پردازش دسته ای موازی که به کاربران اجازه می دهد از مزایای قدرت محاسباتی زیاد برای تحلیل داده های در حد ترابایت در دوره های زمانی تقریبا کوتاهی بهره مند شوند، در حالی که انتزاع برنامه نویسی گوگل یعنی MapReduce یا همتای منبع باز آن Hadoop، فرآیند پیچیده ی اجرای موازی یک برنامه کاربردی روی صدها سرور را برای برنامه نویسان شفاف می سازد. (c) تحلیل های تجاری که حجم زیادی از منابع کامپیوتری را برای درک مشتریان، عادت های خرید، زنجیره تامین و... از بین مقادیر زیادی از داده مورد استفاده قرار می دهند.(d) گسترش برنامه های کاربردی دسکتاپ که متمرکز بر محاسبه بوده و می تواند تله های داده را به ابر ارسال و تنها فرآیند رندرکردن داده را در سمت جلویی نگه دارد که با افزایش پهنای باند شبکه باعث کاهش تاخیر می شود [۱۶].

۲-۵- لایه های معماری محاسبات ابری

همان طور که در شکل ۲-۳ نشان داده شده معمولا محاسبات ابری به سه سطح از ارائه سرویس طبقه بندی می شود: نرم افزار به عنوان سرویس (SaaS)، چارچوب به عنوان سرویس (PaaS) و زیرساخت به عنوان سرویس (IaaS). این لایه ها مدیریت و مجازی سازی سطوح مختلف پشته پاسخ را پشتیبانی می کنند [۱۷].



شکل ۲-۳- پشته محاسبات ابری [۱۷]

- **نرم افزار به عنوان سرویس**

این نوع ابر، ایده ای است که در آن یک نفر یک مجموعه ی نرم افزاری میزبانی شده (اجرا شده روی یک چارچوب و زیرساخت) را به شما ارائه می دهد که شما مالک آن نیستید اما برای استفاده از برخی عناصر آن (به وسیله کاربر یا نوع دیگری از استفاده) هزینه پرداخت می کنید. در این جا شما مجبور به برنامه نویسی یا توسعه نیستید اما ممکن است لازم باشد وارد شده و نرم افزار را پیکربندی کنید(بسیار منعطف، قابل تنظیم و گاهی با قابلیت شخصی سازی). شما تنها برای آنچه که استفاده کرده اید هزینه می پردازید. یک ارائه دهنده SaaS معمولا یک برنامه کاربردی داده شده را میزبانی و مدیریت می کند

و آن را برای چند مستاجر و کاربر روی وب دسترس پذیر می سازد. برخی از ارائه دهندگان SaaS روی پیشنهادهای سرویس ارائه دهنده ابر نوع SaaS یا PaaS دیگری اجرا می شوند. نمونه های معروف SaaS عبارتند از CRM بر حسب تقاضای اوراکل، Salesforce.com و Netsuite.

- **چارچوب به عنوان سرویس**

این نوع ابر، ایده ای است که در آن یک نفر سخت افزار(همانندIaaS) را همراه با مقادیر مورد نیاز از نرم‌افزار کاربردی ارائه می کند- همانند تجمیع مجموعه مشترکی از توابع برنامه نویسی یا پایگاه داده ها به عنوان یک فونداسیون که شما می توانید برنامه کاربردی خودتان را روی آن بسازید. چارچوب به عنوان سرویس(PaaS) یک چارچوب توسعه و استقرار برنامه کاربردی است که به صورت یک سرویس، از طریق وب به توسعه دهندگان ارائه می شود. این سرویس توسعه و استقرار برنامه های کاربردی را بدون هزینه و پیچیدگی خرید و مدیریت زیرساخت‌های مورد نیاز تسهیل می کند و همه تسهیلات مورد نیاز برای تکمیل چرخه حیات ساخت و تحویل برنامه های کاربردی و سرویس های تحت وب را تماماً از طریق اینترنت فراهم می کند. این چارچوب، متشکل از نرم‌افزار زیرساختی و معمولاً یک پایگاه داده، میان افزار و ابزارهای توسعه است. معمولاً یک معماری محاسبات گرید خوشه‌ای و مجازی شده، مبنای نرم‌افزار زیرساختی است. برخی پیشنهاد های PaaS زبان برنامه نویسی یا API مخصوص دارند. به عنوان مثال AppEngine گوگل یک پیشنهاد PaaS است که توسعه دهندگان با زبان جاوا یا پیتون مینویسند. EngineYard، Ruby on Rails است. گاهی اوقات ارائه‌دهندگان PaaS زبان های اختصاصی دارند مانند force.com ازSalesforce.com و Coghead که هم اکنون متعلق به SAP است.

- **زیرساخت به عنوان سرویس**

زیرساخت به عنوان سرویس(IaaS) عبارت است از ارائه سخت افزار(سرور، فضای ذخیره و شبکه) و نرم افزار های مربوطه(سیستم عامل ها، تکنولوژی مجازی سازی، سیستم فایل) به عنوان سرویس. این نوع ابر تکاملی از میزبانی سنتی است که نیاز به تعهد بلندمدت نداشته و به کاربران اجازه میدهد منابع را بر حسب تقاضا فراهم کنند. ارائه دهنده IaaS برخلاف سرویس‌های PaaS، به استثناء حفظ عملکرد دیتا سنتر، مدیریت کمتری انجام می دهد. کاربران بایستی مدیریت و استقرار سرویس های نرم‌افزاری خودشان را به محض ورود به دیتاسنتر انجام دهند. سرویس های وب آمازون ابر،محاسبات توسعه پذیرEC2 و سرویس ذخیره امن S3 نمونه هایی ازIaaS هستند.

۶-۲- مدل کارگزاری

- ابر عمومی:

در ابر عمومی، چندین مشتری منابع محاسباتی فراهم شده توسط یک ارائه دهنده سرویس واحد را به اشتراک می‌گذارند. کاربران می توانند به سرعت به این منابع دست پیدا کنند و تنها برای منابع عملیاتی پول می پردازند. اگرچه ابر عمومی، مزایای قانع کننده ای دارد اما در آن خطر پنهان امنیت و انطباق مقررات و کیفیت خدمات(QoS) وجود دارد.

- ابر خصوصی:

در ابر خصوصی، منابع محاسباتی به وسیله یک شرکت خصوصی استفاده وکنترل می‌شود. این نوع ابر معمولاً در دیتا سنتر یک شرکت، کارگزاری میشود و به وسیله کارکنان داخلی یا ارائه دهنده سرویس مدیریت می‌شود. مزایای اصلی این مدل آن است که امنیت، انطباق و QoS تحت کنترل شرکت است.

- ابر ترکیبی:

سومین نوع، ابر ترکیبی است که ترکیب متداولی از ابر عمومی و خصوصی است. این نوع ابر شرکت را قادر می سازد تا بارکاری ثابت را در ابر خصوصی اجرا کند و هنگامی که نقطه اوج بارکاری رخ می دهد منابع محاسباتی بیشتر را از ابر عمومی درخواست کند سپس هنگامی که دیگر نیازی نبود آن را بازگرداند.

- ابر انجمنی:

چندین سازمان به طور مشترک یک زیرساخت ابر و همچنین سیاست ها، الزامات، ارزش ها و نگرانی ها را ایجاد و به اشتراک می گذارند. ابر انجمنی با درجه ای از توسعه پذیری اقتصادی و تعادل دموکراتیک شکل می‌گیرد. زیرساخت ابر می تواند به وسیله ی یک ارائه دهنده شخص ثالث یا داخل یکی از سازمان های درون انجمن میزبانی شود [۱۸].

۷-۲- چالش های محاسبات ابری

انطباق فعلی محاسبات ابری با چالش های زیادی همراه است زیرا کاربران هنوز در مورد صحت و اعتبار آن تردید دارند. بر اساس نظر سنجی انجام شده توسط IDC[1] در سال ۲۰۰۸، چالشهای عمده ای که مانع استفاده از محاسبات ابری در سازمان ها می شوند عبارتند از [۱۹]:

A. امنیت: واضح است که مساله امنیت، نقش مهمی را در ممانعت از پذیرش محاسبات ابری ایفا کرده است. بدون شک قرار دادن داده ها و اجرای نرم افزار روی هارددیسک شخص دیگر با استفاده از پردازنده شخص دیگر برای بسیاری از مردم دلهره آور است. مسائل امنیتی شناخته شده از قبیل اتلاف داده، فیشینگ، بات نت ها(از راه دور روی مجموعه ای از ماشین ها اجرا

[1] International Data Corporation

می شود)، تهدیداتی جدی برای داده و نرم‌افزار سازمان می باشند. علاوه بر این، مدل چند مستاجری و منابع کامپیوتری جمع آوری شده، چالش های امنیتی جدیدی در محاسبات ابری ایجاد کرده که نیاز به تکنیک های جدید برای مقابله با آن وجود دارد. به عنوان مثال هکرها می‌توانند از ابر برای سازماندهی بات نت استفاده کنند زیرا ابر، سرویس های زیرساختی قابل اطمینان تر و با هزینه نسبتاً ارزان تری را برای شروع یک حمله در اختیار آنان قرار می دهد.

B. مدل هزینه: مصرف کنندگان ابر، بایستی بین محاسبات، ارتباطات و یکپارچگی سبک و سنگین کنند. با این که مهاجرت به ابر می تواند هزینه های زیر ساختی را به طور چشمگیری کاهش دهد، از طرف دیگر هزینه‌های ارتباطات داده ای را افزایش می‌دهد. به عنوان مثال هزینه انتقال داده های سازمان به و از ابر عمومی و انجمنی و هزینه به ازای استفاده از هر واحد منبع محاسباتی، به نظر بالاتر می‌رسد. این مساله.، به ویژه در حالتی که مصرف کننده از مدل پیاده سازی ابر ترکیبی استفاده کند مهمتر است، زیرا در این مدل.، داده‌های سازمان بین تعدادی ابر عمومی.، خصوصی(ساختار IT داخلی) و انجمنی، توزیع می شوند. به طور مستقیم نیاز به محاسبه بر حسب تقاضا تنها در مورد کارهای متمرکز بر Cpu احساس می شود.

C. مدل مطالبه هزینه: واکشی منبع به صورت توسعه پذیر، تجزیه و تحلیل هزینه را نسبت به مراکز داده معمولی، که اغلب هزینه‌هایشان را بر اساس مصرف محاسبات ایستا محاسبه می‌کنند، پیچیده تر کرده است. علاوه بر آن به جای یک سرور فیزیکی پایه‌ای، یک ماشین مجازی واحد تجزیه و تحلیل هزینه شده است. برای ارائه کنندگان ابرSaaS .، هزینه توسعه چند مستاجری می تواند بسیار قابل توجه باشد. این هزینه ها شامل مواردی است از قبیل: توسعه و طراحی مجدد نرم افزار که در اصل برای تک مستاجری طراحی شده است، هزینه

ارائه خصیصه هایی که امکان سفارشی سازی را فراهم می کنند، افزایش کارایی و امنیت برای دسترسی همزمان کاربر و نیز مقابله با پیچیدگی های ناشی از تغییرات فوق. به همین ترتیب ارائه دهندگان ابر SaaS، نیاز به برقراری توازن بین تدارک چند مستاجری و صرفه جویی هزینه ناشی از آن از قبیل کاهش سربار از طریق استهلاک، کاهش تعداد مجوز های نرم افزاری روی سایت و... دارند. از این رو، یک مدل مطالبه هزینه استراتژیک و قابل اعتماد برای سودآوری و پایداری ارائه دهندگان SaaS ضروری است.

D. توافق سطح سرویس: با وجود آنکه مصرف‌کننده ابر هیچ کنترلی روی منابع محاسباتی زیربنایی ندارد، اما هنگام انتقال عملکردهای تجاری اصلی خود به ابر، بایستی کیفیت، دسترس پذیری، قابلیت اطمینان و کارایی این منابع را تضمین کند. به عبارت دیگر برای مصرف کنندگان، گرفتن ضمانت از ارائه دهنده در هنگام تحویل سرویس، امری حیاتی است. به طور معمول این امر از طریق توافق سطح سرویس (SLAs) که بین ارائه دهندگان و مصرف کنندگان انجام می شود، تامین می گردد. اولین مسئله، تعریف خصوصیات SLA به گونه‌ای است که سطح مناسبی از جزئیات را داشته باشد. در واقع یک توازن بین وضوح و پیچیدگی به طوری که بتواند اغلب انتظارات مصرف کننده را پوشش دهد. از طرفی وزن دهی، اعتبار سنجی، ارزیابی و پیاده سازی آن برای مکانیزم تخصیص منابع روی ابر نسبتاً ساده باشد. علاوه بر آن ابر های مختلف که (IaaS، PaaS و SaaS) را ارائه می دهند نیاز به خصوصیات SLA اضافی متفاوتی خواهند داشت. این امر به نوبه خود یکسری مسائل پیاده سازی را برای ارائه دهندگان ابر ایجاد می کند. علاوه بر آن مکانیزمهای SLA پیشرفته، نیازمند ادغام دائمی بازخور کاربر و ویژگی های سفارشی سازی در چارچوب ارزیابی SLA هستند

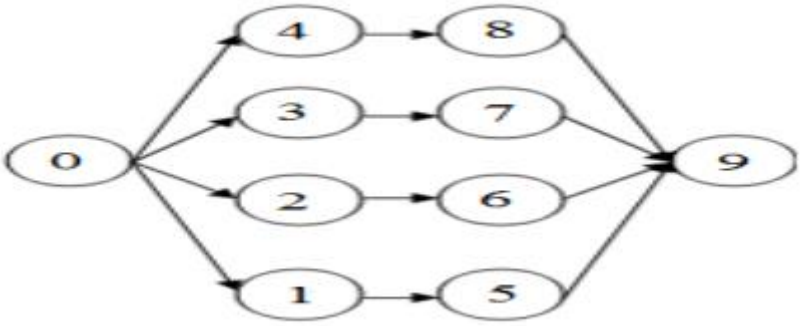
E. چه چیزی مهاجرت کن: بر اساس یک نظرسنجی(سایز نمونه = ۲۴۴) که توسط IDC در سال ۲۰۰۸ انجام گرفته، هفت سیستم/ برنامه کاربردی IT که به ابر مهاجرت کرده اند عبارتند از: برنامه های کاربردی مدیریتی(۲۵٪) برنامه های کاربردی مبتنی بر همکاری(۲۵,۴ ٪) برنامه های کاربردی شخصی(۲۵٪) برنامه های کاربردی تجاری(۲۳,۴٪) توسعه و استقرار برنامه های کاربردی(۱۶,۸٪) ظرفیت سرور(۱۵,۶٪) و ظرفیت ذخیره(۱۵,۵٪). این نتایج نشان می دهد که سازمان ها هنوز نگرانی های امنیتی/ حریم خصوصی، برای انتقال داده هایشان به روی ابر دارند. در حال حاضر عملکردهای جانبی از قبیل برنامه های کاربردی شخصی و مدیریت IT راحت ترین سیستم های IT برای انتقال هستند. سازمان ها در بکارگیری IaaS نسبت به SaaS محافظه کارترهستند. بخشی از آن به این دلیل است که اغلب، توابع جانبی به ابر منتقل می شوند و توابع اصلی درون سازمان باقی می مانند. همچنین نظرسنجی نشان می‌دهد که در طول سه سال ۳۱,۵٪ سازمان ها، ظرفیت ذخیره سازی خود را به ابر منتقل خواهند کرد. با این حال این عدد در مقایسه با برنامه های مبتنی بر همکاری۴۶,۳٪ در آن زمان تقریبا کم است.

F. مسائل مربوط به قابلیت همکاری در ابر: در حال حاضر هر ابر، برای چگونگی تعامل کلاینت ها، برنامه های کاربردی و کاربران با ابر، راهکار ویژه خود را دارد که باعث پدیده(ابرمبهم) میشود. این امر با اجبار ارائه دهنده به قفل سازی، به شدت توسعه اکوسیستم ابر را محدود می کند و لذا توانایی کاربران را برای انتخاب ارائه دهندگان جایگزین که همزمان سرویس هایی را برای بهینه سازی منابع در سطوح مختلف یک سازمان ارائه می کنند، محدود می سازد. مهمتر از همهAPI های اختصاصی در ابر، باعث می شود که ادغام سرویس های ابری با سیستم های قدیمی موجود در سازمان(به عنوان مثال مرکز داده پیش فرض برای مدل سازی

بسیار تعاملی برنامه های کاربردی یک شرکت دارویی) بسیار دشوار باشد. هدف اصلی در قابلیت همکاری آن است که حرکت یکپارچه داده شناور را بین ابر ها و ما بین ابر و برنامه های کاربردی محلی، تحقق بخشد. سطوحی وجود دارد که قابلیت همکاری در ابر ضروری به نظر می‌رسد. اول آنکه سازمان برای بهینه سازی دارایی‌های IT و منابع محاسباتی، اغلب نیاز دارد تا دارایی‌هایIT و قابلیت های شایسته اصلی را درون خود نگه دارد در حالی که توابع و فعالیت های جانبی(به عنوان مثال سیستم منابع انسانی) را روی ابر برون سپاری کند. دوم آنکه اغلب اوقات یک سازمان به منظور بهینه سازی ممکن است توابع جانبی خود را به سرویس های ارائه شده توسط ارائه دهندگان مختلف برون‌سپاری کند. به نظر می‌رسد که استانداردسازی یک راه حل مناسب برای رفع مسائل قابلیت همکاری است. با این حال همانطور که محاسبات ابری شروع به افول کرده مساله قابلیت همکاری در دستور کار فوری ارائه‌دهندگان ابر اصلی قرار نگرفته است.

۸-۲- جریان های کار

به طور کلی برنامه های کاربردی مبتنی بر جریان کار به صورت یک دنباله اجرا می شوند. زیرا به صورت مجموعه ای از کارهای مختلف برای دستیابی به یک هدف خاص در نظر گرفته می شوند [۱۴]. کارها بر اساس وابستگی‌های داده ای اجرا می‌شوند. این کارها ارتباطات پدر/ فرزندی دارند. کار پدر بایستی قبل از کار فرزند اجرا گردد. جریان کار معمولا به صورت یک گراف جهت دار (DAG) بدون حلقه نمایش داده میشود. این گراف به صورت G (v,e) نمایش داده می‌شود. در اینجاv بیانگر تعداد گره ها و e بیانگر اطلاعات مربوط به وابستگی های داده بین کارها است. دیاگرام زیر نشان دهنده یک جریان کاری است.



شکل ۲-۴- نمایش جریان کاری در گراف [۱۴]

شکل ۲-۴ وابستگی های بین کارهای درون یک گراف جریان کاریG را نشان می دهد. کارهای فرزند ۱ و ۲ و ۳ و ۴ بعد از کار پدر ۵ اجرا شده‌اند. گره فرزند، ورودی خود را از خروجی گره پدر می گیرد. گره ۵ به عنوان یک گره ورودی و گره ۹ به عنوان یک گره خروجی عمل می‌کنند. بعد از تکمیل کارهای ۵ و ۶ و ۷ و ۸ کار ۹ اجرا می‌شود. در یک گراف، یک کار ورودی کاری است که پدر ندارد و یک کار خروجی.،کاری است که فرزند ندارد. دقیقاً یک کار ورودی و یک کار خروجی از الزامات الگوریتم های زمان بندی کار است. Makespan بیانگر کارایی یک جریان کاری بوده و به صورت، اختلاف زمان شروع از زمان خاتمه در یک جریان کاری محاسبه میگردد. هر گره و هر یال دریک گراف، با یک وزن خاص نمایش داده می‌شوند. باید توجه داشت که اعمال تکنیک های مختلف برای محاسبه مقادیر گره ها و یال ها تاثیر قابل توجهی روی زمان بندی خواهد داشت. برخی تکنیک های محاسبه وزن به صورت زیر هستند [۲۰]:

مقدار میانگین[1](ME)، این روش میانگین هزینه اجرای هر کار روی هر ماشین را برای محاسبه وزن یک گره در نظر می‌گیرد. برای محاسبه وزن یک یال میانگین هزینه ارتباطات بین دو کار در نظر گرفته می شود.

مقدار میانه[2](M)، مشابه روش مقدار میانگین است با این تفاوت که به جای میانگین مقدار میانه را در نظر می گیرد.

بهترین مقدار[3](B)، به منظور یافتن وزن یک گره،کمترین مقدار از هزینه اجرا را در نظر می‌گیرد. مقدار یک یال به صورت هزینه انتقال از دو ماشین مختلف که حداقل هزینه اجرا از کارهای مرتبط را داشته باشد محاسبه می گردد.

بدترین مقدار[4](W)، به منظور یافتن وزن یک گره، بدترین مقدار از هزینه اجرا را در نظر می‌گیرد. مقدار یک یال به صورت هزینه انتقال از دو ماشین مختلف که حداکثر هزینه اجرا از کارهای مرتبط را داشته باشد محاسبه می گردد.

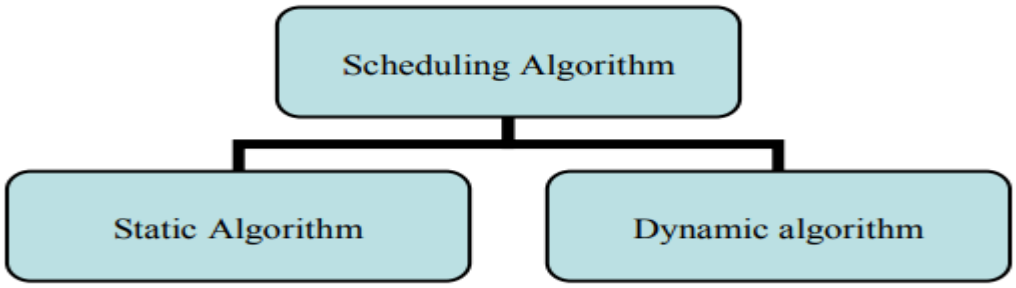
بهترین مقدار ساده[5](SB)، بهترین مقدار از هزینه اجرا و ارتباطات را در نظر می‌گیرد.

بدترین مقدار ساده[6](SW)، بدترین مقدار از هزینه اجرا و ارتباطات را در نظر می‌گیرد.

[1] Median value
[2] Mean
[3] Best
[4] Worst
[5] Simple best
[6] Simple worst

۹-۲- مفهوم زمان بندی

زمان بندی در بستر ابر، به معنای انتخاب مناسب ترین منبع برای اجرای یک کار یا اختصاص ماشین ها به کار ها به گونه ای است که زمان تکمیل کار(makespan) حداقل گردد. به طور کلی در الگوریتم های زمان بندی لیست کارها به وسیله دادن اولویت به هر کار ساخته می شود. کارها بر اساس اولویت انتخاب شده و به یک پردازنده تخصیص می‌یابند که یک تابع هدف پیش‌فرض را اجرا می کند. دو نوع الگوریتم زمان بندی وجود دارند: اولین نوع استاتیک است که اطلاعات مربوط به تخمین زمان اجرای کار، ساختار کار و نگاشت منابع را قبل از اجرا در اختیار دارد. دومین نوع الگوریتم های پویا هستند که اطلاعات لازم را پیش از اجرای کار و در وضعیت آماده تخمین می زنند [۱۱].



شکل ۲-۵- انواع الگوریتم های زمان بندی[۱۱]

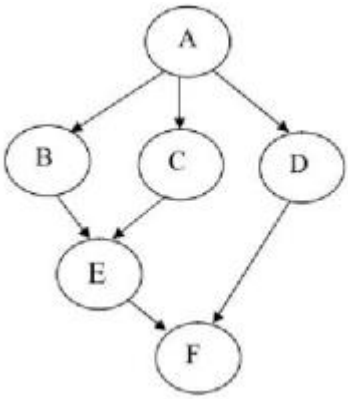
لیست جدیدی از الگوریتم های زمان بندی برای محیط های ناهمگن پیشنهاد شده اند. اغلب کارها در زمینه زمان بندی محدود به برنامه کاربردی مبتنی بر یک جریان کاری هستند. مواردی وجود دارند که ما نیازمند زمان بندی برای جریان‌های کار چندگانه هستیم [۲۱]. نویسنده برای حل این مساله دو رویکرد را ارائه می کند. رویکرد اول، یک رویکرد ترکیبی است که در آن الگوریتم.، چند جریان کاری را ادغام و سپس زمان بندی را اجرا می کند. رویکرد دوم رویکرد منصفانه است که در آن بعد از تکمیل یک کار، مقادیر مجددا محاسبه می شوند سپس برای زمان بندی تصمیم گرفته می شود. جریان های

کاری چندگانه به وسیله کاربران مختلف و در زمانهای مختلف ارسال می شوند و به صورت آنلاین مدیریت می شوند [۲۲].

زمان بندی ایستا و پویا : زمان بندی ایستا، اجازه می دهد تا داده های مورد نیاز از قبل واکشی شوند و مراحل انتهایی از اجرای کار را روی کانال قرار می دهد. زمان بندی ایستا سربار زمان اجرای حداقل دارد. در مورد زمان بندی پویا، اطلاعات مربوط به اجزای کار یا وظیفه از قبل مشخص نیست بنابراین زمان اجرای یک وظیفه ممکن است مشخص نباشد و تخصیص وظایف تنها در زمان اجرای برنامه کاربردی انجام می گیرد [۲۳].

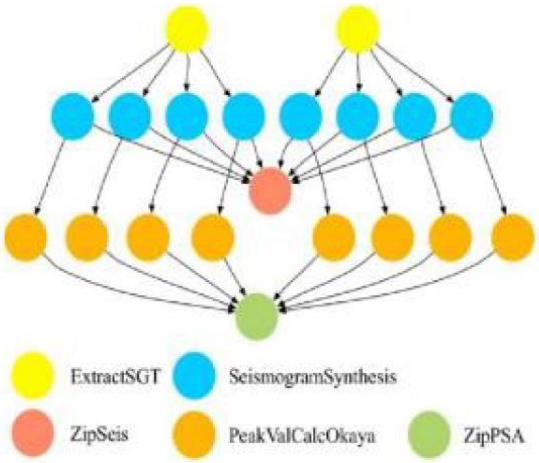
۱۰-۲- زمان بندی جریان های کاری

زمان بندی عبارت است از شیوه تخصیص وظایف به منابع محاسباتی جهت اجرای آن ها. در محیط های توزیع شده همانند ابر، کارهای مختلف به منابع محاسباتی مجازی متفاوتی زمان بندی می شوند و آن کارها به صورت از راه دور اجرا می شوند. مساله زمان بندی می تواند به دو دسته که عبارتند از مساله زمان بندی و مساله تصمیم گیری بر مبنای هدف، طبقه بندی شود. مساله بهینه سازی، بهترین جواب را از بین همه گراف های DAG جریان کاری پیدا می کند و دارای ساختار تکرار است که در آن وظایف جریان کاری تکرار می‌شوند. تکرار می تواند حلقه یا چرخه نامیده شود. ساختارهای تکرار در برنامه های کاربردی علمی متداول هستند.



شکل ۲- ۶- نمایش جریان کاری مبتنی بر DAG

این نوع از ساختار جریان کاری می‌تواند به عنوان بلوک های ساختاری برای ساخت جریان های کاری بزرگ و پیچیده مورد استفاده قرار گیرد. به طور کلی جریان‌های کاری به دو دسته ساده و علمی طبقه بندی می‌شوند. شکل ۲-۷ یک نمونه جریان کاری علمیCybershake و شکل۲-۶ به جریان کاری ساده را نشان می دهد.



شکل ۲-۷- جریان کاری [۲۴]

۲-۱۱- زمان بندی جریان کار در ابر

دو بخش اصلی برای اجرای جریان کاری روی محیط ابری وجود دارند که عبارتند از:

- تدارک منبع
- زمان بندی کار

تخصیص منابع سیستم به کارهای مختلف، تحت عنوان زمان بندی شناخته می شود. سیستم، صف های اولویت را برای کارهای در حال انتظار نگهداری می کند و تصمیم می گیرد که کدام کار اجرا شود و چه مقدار زمان به آن اختصاص یابد به طوری که همه کارها در یک زمان تکمیل می شوند. زمان بندی کار، عبارت است از تخصیص کارها به منابع مناسب به گونه‌ای که حداکثر استفاده از منابع، بهبود بهره وری، افزایش کارایی و QoS و حداقل کردن زمان اجرا محقق شوند. الگوریتم های زمان بندی که برای محیط های خوشه‌ای و گرید توسعه پیدا کرده‌اند اصولا روی زمان بندی کار تمرکز دارند و چون تمام منابع مورد نیاز در اصل ایستا هستند نیاز به تدارک پویا وجود ندارد. اما در مورد محیط‌های ابر منابع مورد نیاز در اصل پویا هستند و لذا بایستی هم تدارک منبع و هم زمان بندی کارها، مد نظر قرار گیرند تا یک زمان بندی کارآمد برای اجرای جریان های کار در محیط ابری تولید شود.

۱۲-۲- اهداف زمان بندی جریان کار

اهداف زمان بندی که در الگوریتم های زمان بندی جریان کاری مختلف در نظر گرفته می شوند عبارتند از [۲۳]:

زمان اجرا(ET)[1].، زمان صرف شده توسط ماشین مجازی برای اجرای کار یا وظیفه.

[1] Execution Time

Makespan (Ms)، به طول کلی زمان بندی، از هنگام ارسال کار تا زمان تکمیل گفته می شود. اکثر الگوریتمهای مرور شده، نگران زمان اجرای جریان کار یا Makespan هستند. این آیتم نیز، همانند هزینه به عنوان بخشی از اهداف زمان بندی در نظر گرفته شده و سعی در حداقل کردن مقدار آن یا تعریف یک حد زمانی یا مهلت زمانی برای اجرای جریان کار بر اساس آن دارد.

مصرف انرژی: مصرف توان یا انرژی منابع در طول سرویس دهی.

هزینه/ بودجه: منظور هزینه استفاده از منابع است. الگوریتم هایی که برای چارچوب ابری طراحی می شوند باید هزینه اجاره زیرساخت را در نظر بگیرند. اگر نتوانند این کار را انجام دهند، هزینه اجاره ماشین های مجازی، انتقال داده و استفاده از فضای ذخیره ابر میتواند به طور قابل توجهی بالا رود. این هدف به وسیله سعی در حداقل کردن مقادیر یا با کمتر کردن هزینه صرف شده برای منابع(یعنی بودجه) در الگوریتم ها گنجانده شده است. تمامی الگوریتم ها بین هزینه و دیگر اهداف مرتبط با کارایی یا نیازمندی‌های غیرعملکردی همانند امنیت، قابلیت اطمینان و مصرف انرژی، توازن برقرار می‌کنند. به عنوان مثال یکی از متداولترین نیازهای QoS، حداقل کردن هزینه کلی و درعین حال ارضای یک محدودیت مهلت تعریف شده توسط کاربر است.

مهلت/ زمان: محدودیت زمانی برای اجرای جریان کار.

قابلیت اطمینان: احتمال اجرای موفقیت آمیز کار.

مقیاس پذیری: توانایی رشد خودکار سیستم بر اساس افزایش تقاضا.

[1] Makespan

بهره برداری از منابع(RU): بهره برداری حداکثری از منابع محاسباتی.

هزینه ارتباطات: هزینه انتقال داده بین دو گره.

توازن بار: توزیع یکنواخت جریان های کاری بین چندین منبع محاسباتی.

تحمل پذیری خطا: اطمینان از دسترس پذیری و قابلیت اطمینان برای سرویس های ضروری.

۱۳-۲- استراتژی بهینه سازی

طبقه بندی استراتژی بهینه سازی در شکل ۲-۸ نشان داده شده است. بر اساس تعریف Casavant و Kuhl، الگوریتم های زمان بندی می توانند به دو دسته بهینه و زیر بهینه تقسیم شوند. به علت اینکه مساله مورد بحث از نوع NP کامل است پیدا کردن راه حل های بهینه حتی در نسخه کوچک مقیاس مساله هزینه بر است و این استراتژی را در اکثر شرایط غیر عملی می سازد. علاوه بر این بهینه کردن راه حل به وسیله مفروضات ایجاد شده توسط زمان بند که مرتبط با وضعیت سیستم،نیازمندی های منابع و خصوصیات محاسباتی کارها است، محدود شده است. بر این مبنا پیدا کردن راه حل بهینه برای جریان های کاری بزرگ مقیاس که تحت کارایی های متنوع اجرا می شوند، ممکن است راضی‌کننده نباشد. با این حال این استراتژی ممکن است برای جریان های کاری کوچک با وظایف درشت دانه،که از نظر محاسباتی متمرکز هستند و انتظار می‌رود برای مدت طولانی اجرا شوند، جذاب تر باشد. چند روش برای یافتن زمان بندی های بهینه وجود دارند. به طور خاص Casavant و Kuhl [۲۵]، چهار استراتژی را برای مساله زمان بندی چند پردازنده عمومی، شناسایی کرده‌اند: جستجو و شمارش فضای پاسخ.، نظریه گراف، برنامه نویسی ریاضی و نظریه صف. روش های شمارش فضا و مدل های ریاضی به مساله ما بیشتر مرتبط هستند. در بین الگوریتم های مرور شده، برنامه های خطی عدد صحیح

ترکیبی(MILPs)، برای بهینه سازی سطح جریان کاری مورد استفاده قرار گرفته است. همان استراتژی و برنامه نویسی پویا برای یافتن زمان بندی های بهینه برای زیر مجموعه ای از وظایف جریان کار یا نسخه ساده شده مساله مورد استفاده قرار گرفته اند هرچند که این بهینه سازی نیمه سراسری منجر به یک راه حل بهینه نمی شود. اکثر الگوریتم ها روی تولید پاسخ های تقریبی یا نزدیک به بهینه تمرکز کرده اند. ما برای طبقه زیر بهینه، ۳ روش مختلف که توسط الگوریتم های مطالعه شده مورد استفاده قرار گرفته اند را شناسایی کرده ایم. دو دسته اول رویکرد های اکتشافی و فرا اکتشافی هستند که توسط Yu و همکاران [۲۶] تعریف شده‌اند. ما به این مدل، یک دسته سوم ترکیبی را اضافه کرده ایم که شامل الگوریتم هایی می شود که استراتژی‌های مختلف را ترکیب کرده اند.

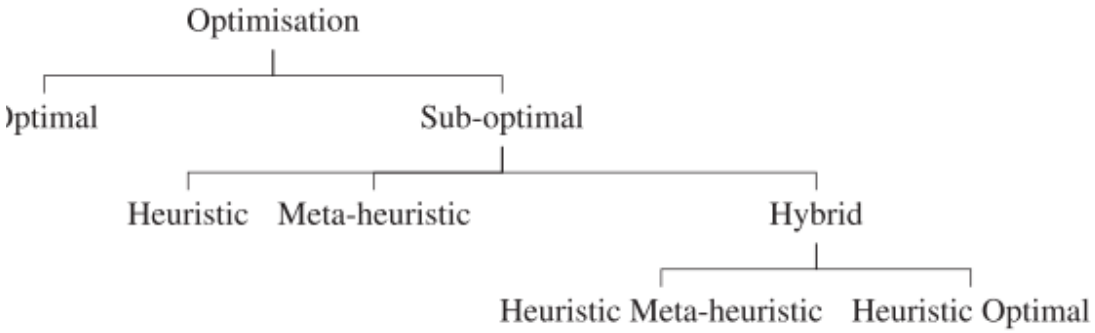
اکتشافی ها:

به طور کلی یک اکتشاف، مجموعه‌ای از قواعد است که هدفش یافتن پاسخ برای یک مساله خاص است. این قواعد، خاص آن مساله هستند و به گونه ای طراحی شده اند که پاسخ تقریبی در یک بازه زمانی قابل قبول پیدا می شود. برای سناریوی زمان بندی مطرح شده در اینجا، یک رویکرد اکتشافی، دانش مربوط به خصیصه های ابر و همینطور خصیصه های برنامه کاربردی را برای یافتن یک زمان بندی که محدودیت های QoS تعریف شده توسط کاربر را ارضاء کند، به کار میبرد. مزیت اصلی الگوریتم های زمان بندی مبتنی بر اکتشاف، کارایی موثر آنها است. آنها تمایل به یافتن راه حل های رضایت بخش در یک زمان کافی دارند. همچنین این الگوریتم ها پیاده سازی آسان تری داشته و نسبت به روش‌های مبتنی بر فرااکتشاف قابل پیش بینی تر هستند.

فرا اکتشاف ها:

در حالیکه اکتشاف ها برای عملکرد بهتر روی یک مساله خاص طراحی شده اند، فرا اکتشاف ها الگوریتم های عمومی تری هستند که برای حل مسائل بهینه سازی طراحی شده اند. آنها استراتژی های سطح بالایی هستند که اکتشاف های خاص مساله را، برای یافتن یک راه حل نزدیک به بهینه، روی مساله اعمال می کنند. رویکردهای فرا اکتشافی در مقایسه با الگوریتم های مبتنی بر اکتشاف از نظر محاسباتی متمرکزتر و زمان بیشتری برای اجرا نیاز دارند. همینطور این رویکردها تمایل دارند تا هنگام جستجوی هدایت شده در بین جواب های مختلف، زمان بندی های مطلوب تری را پیدا کنند. استفاده از فرا اکتشاف برای حل مساله زمان بندی جریان کار در محیط ابری چالش هایی دارد از قبیل، مدل سازی نظری تعداد نامحدود منابع، تعریف عملگر هایی برای پیشگیری از جستجوی جواب های نامعتبر(مثلا نقض وابستگی داده) جهت تسهیل همگرایی و برش فضای حالت با استفاده از اکتشاف هایی روی مدل منبع ابری پایه.

ترکیبی: الگوریتم هایی که رویکرد ترکیب را به کار می‌برند، ممکن است از روش‌های فرا اکتشافی برای بهینه سازی زمان بندی گروهی از وظایف جریان کار استفاده کنند. گزینه دیگر، یافتن راه حل های بهینه برای نسخه های ساده شده و/ یا کوچکتر مساله و ترکیب آنها با استفاده از اکتشاف ها است. در این روش الگوریتم ها ممکن است بتوانند نسبت به روش‌های مبتنی بر اکتشاف تصمیمات بهینه سازی بهتری بگیرند و در عین حال به علت در نظر گرفتن فضای مساله کوچکتر، زمان محاسبه را کاهش دهند.



شکل ۲-۸- انواع استراتژی های بهینه سازی

فصل سوم

کارهای پیشین

در این بخش می‌خواهیم به مطالعه کارهایی که در این زمینه موجودند بپردازیم. اکثر الگوریتم‌ها، روی تولید راه‌حل‌های تقریبی یا نزدیک به بهینه تمرکز کرده‌اند. برای طبقه زیر بهینه، ما دو روش متفاوت که توسط الگوریتم‌های مطالعه شده مورد استفاده قرار گرفته‌اند شناسایی کرده‌ایم. دو دسته اول عبارتند از رویکردهای اکتشافی و فرا اکتشافی.

۱-۳- تکنیک های مبتنی بر اکتشافی

به طور کلی یک اکتشاف، مجموعه‌ای از قواعد است که هدفش یافتن پاسخ برای یک مساله خاص است. این قواعد، خاص آن مساله هستند و به گونه ای طراحی شده اند که پاسخ تقریبی در یک بازه زمانی قابل قبول پیدا می شود. برای سناریوی زمان بندی مطرح شده در اینجا، یک رویکرد اکتشافی، دانش مربوط به خصیصه های ابر و همینطور خصیصه های برنامه کاربردی را برای یافتن یک زمان بندی که محدودیت های QoS تعریف شده توسط کاربر را ارضاء کند، به کار میبرد. مزیت اصلی الگوریتم های زمان بندی مبتنی بر اکتشاف، کارایی موثر آنها است. آنها تمایل به یافتن راه حل های رضایت بخش در یک زمان کافی دارند. همچنین این الگوریتم ها پیاده سازی آسان تری داشته و نسبت به روش‌های مبتنی بر فرااکتشاف قابل پیش بینی تر هستند [۲۳].

در [۲۷]، نویسندگان به مساله زمان بندی جریان کار علمی در محیط های ابری تجاری با تدارک پویا، پرداخته‌اند. در این مقاله، زمان بندی جریان کار برای دستیابی به هزینه کمتر همراه با زمان پاسخ مناسب در محیطهای ابری مورد استفاده قرار گرفته است. آنها الگوریتم های جدید مهلت محدود متناسب (PDC[1]) و مسیر بحرانی مهلت محدود (DCCP[2]) را معرفی کرده اند. PDC، به وسیله حداکثر

[1] Proportional Deadline Constrained
[2] Deadline Constrained Critical Path

کردن موازی سازی در جریان کاری از طریق جداسازی آن به سطوح منطقی و سپس تقسیم متناسب مهلت زمانی کلی جریان کاری بین آن سطوح، عمل می کند. الگوریتم DCCP مشابه PDC است، با این تفاوت که یک مسیر بحرانی محدود شده در جریان کاری معین می کند تا بدین وسیله، کارهایی را که در یک نمونه با هم در ارتباط هستند، در همسایگی قرار دهد. از نظر کارایی هزینه، به طور کلی الگوریتمهای DCCP و PDC، پایین ترین هزینه محاسبه را روی تمام جریان های کاری و تنظیمات نمونه به دست آورده‌اند.

در [۲۸]، برای زمان بندی برنامه های کاربردی خاص در محیط ابری، یک الگوریتم حداقل سازی هزینه پویا و اکتشاف مهلت محدود به نامJIT-C ارائه شده است. به منظور پایین نگه داشتن هزینه اجرا، منابع قبل از آنکه نیاز باشد فراهم می شوند. در این الگوریتم، هدف مهلت زمانی، از طریق نظارت پیوسته روی وظایف در حال اجرا و تصمیمات زمان بندی پویا با هزینه موثر برای وظایف بعدی به طوری که محدودیت مهلت زمانی نقض نشود، برآورده می شود. نتایج شبیه سازی روی چهار جریان کاری شناخته شده، نشان می‌دهد که در مقایسه با سایر اکتشاف های معروف دیگر مانندIC-PCP ، RCT و RTC، الگوریتم ارائه شده، برای مهلت زمانی، نرخ برخورد بالاتری را نشان می دهد. علاوه بر آن، این الگوریتم، از زمان سکون موجود به همراه مهلت زمانی ریلکس شده، استفاده می کند تا زمان بندی ارزانتر با هزینه اجرای کمتری تولید کند. در مقایسه با RTC که الگوریتم پایه با بهترین عملکرد است، الگوریتم JIT-C ارائه شده، زمان بندی هایی با متوسط۳۴٪ هزینه کمتر تولید می کند.

در [۲۹]، یک سیستم زمان بندی جریان کاری چهار لایه ارائه شده است. سیاست زمان بندی جدید CWSA، برای زمان بندی برنامه های کاربردی مبتنی بر جریان کار در محیط محاسبات ابری ارائه شده است. یک تجزیه و تحلیل روی معیارهای کارایی مختلف انجام گرفته است. شبیه سازی گسترده‌ای برای

ارزیابی کارایی آن ارائه شده سپس عملکرد CWSA با سیاست های زمان بندی مختلف مقایسه شده تا کارایی و استحکام راه حل پیشنهادی برجسته گردد. نتایج نشان داده کهCWSA ، نسبت به دیگر سیاست های زمان بندی بهتر عمل می کند. مهمتر آن که CWSA نشان داده به وسیله کاهش زمان های بیکاری در گره های منبع ابر منابع محاسباتی را به طور مناسبی به کار می‌گیرد.

در [۳۰]، نویسندگان یک استراتژی زمان بندی منحصر به فرد را برای جریان های کاری در محیط ابری پیشنهاد کرده‌اند تا به مسائل مربوط به هزینه و زمان رسیدگی کند. این الگوریتم، زمان اجرای وظایف در مراحل مختلف را، بر مبنای ارتفاع وظایف به وسیله تشکیل گروههای وظیفه‌ای حداقل می کند. زمان تکمیل در هر ارتفاع به حداقل سطح ممکن کاهش می‌یابد بنابراین تخصیص منابع اضافی نیز زمان خیلی کمتری می گیرد و لذا زمان تکمیل سراسری جریان کار نیز کاهش می یابد. الگوریتم پیشنهادی زمان خیلی کمی برای تخصیص وظایف یک جریان کاری به ماشین‌های مجازی نیاز دارد. زمان Makespanاین الگوریتم، نسبت به سایر الگوریتم های موجود به طور قابل توجهی کاهش یافته و همچنین از طریق گرفتن یک منبع اضافه تنها زمانی که نیاز باشد، هزینه کلی اجرای یک وظیفه راکاهش میدهد.

در [۳۱]، یک الگوریتم زمان بندی با پیچیدگی زمانی درجه دو ارائه شده که دو مشکل جدی زمان بندی هزینه و زمان را برای جریان کار مبتنی برQoS بررسی می کند و زمان بندی محدود به بودجه و مهلت نامیده (DBCS[1]) می شود. الگوریتم DBCS، بر اساس بودجه و زمان مشخص شده توسط مشتری،راه حل ممکن را با ارضای هر دو محدودیت با همان نسبت موفقیت سایر الگوریتمهای مرتبط دیگر، می یابد.جدول ۳-۱ خلاصه از تکنیک های مبتنی بر اکتشافی را نشان می دهد.

[1] Deadline-Budget Constrained Scheduling

جدول ۳-۱- خلاصه تکنیک های مبتنی بر اکتشافی

#	عنوان	روش	مزیت
۱	Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources	در [۲۷]، زمان بندی جریان کار برای دستیابی به هزینه کمتر همراه با زمان پاسخ مناسب در محیطهای ابری مورد استفاده قرار گرفته است. آنها الگوریتم های جدید مهلت محدود متناسب (PDC) و مسیر بحرانی مهلت محدود (DCCP) را معرفی کرده اند	از نظر کارایی هزینه، به طور کلی الگوریتمهای DCCP و PDC، پایین ترین هزینه محاسبه را روی تمام جریان های کاری و تنظیمات نمونه به دست آورده‌اند.
۲	A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment	در [۲۸]، برای زمان بندی برنامه های کاربردی خاص در محیط ابری، یک الگوریتم حداقل سازی هزینه پویا و اکتشاف مهلت محدود به نامJIT-C ارائه شده است	در مقایسه با RTC الگوریتم ارائه شده، برای مهلت زمانی، نرخ برخورد بالاتری را نشان می دهد. که الگوریتم پایه با بهترین عملکرد است، الگوریتم JIT-C ارائه شده، زمان بندی هایی با متوسط۳۴٪ هزینه کمتر تولید می کند.
۳	Workflow scheduling in multi-tenant cloud computing environments	در [۲۹]، یک سیستم زمان بندی جریان کاری چهار لایه ارائه شده است. سیاست زمان بندی جدیدCWSA ، برای زمان بندی برنامه های کاربردی مبتنی بر جریان کار در محیط محاسبات ابری ارائه شده است.	CWSA نشان داده به وسیله کاهش زمان های بیکاری در گره های منبع ابر منابع محاسباتی را به طور مناسبی به کار می‌گیرد.
۴	A novel method for scheduling workflows in cloud computing environment	در [۳۰]، نویسندگان یک استراتژی زمان بندی منحصر به فرد را برای جریان های کاری در محیط ابری پیشنهاد کرده‌اند تا به مسائل مربوط به هزینه و زمان رسیدگی کند.	زمان Makespan این الگوریتم، نسبت به سایر الگوریتم های موجود به طور قابل توجهی کاهش یافته و همچنین از طریق گرفتن یک منبع اضافه تنها زمانی که نیاز باشد، هزینه کلی اجرای یک وظیفه راکاهش میدهد.
۵	Low-time complexity budget–deadline constrained workflow scheduling on heterogeneous resources	در [۳۱]، یک الگوریتم زمان بندی با پیچیدگی زمانی درجه دو ارائه شده که دو مشکل جدی زمان بندی هزینه و زمان را برای جریان کار مبتنی برQoS بررسی می کند	الگوریتم DBCS، بر اساس بودجه و زمان مشخص شده توسط مشتری،راه حل ممکن را با ارضای هر دو محدودیت با همان نسبت موفقیت سایر الگوریتمهای مرتبط دیگر، می یابد.

۲-۳- تکنیک های مبتنی بر فرا اکتشافی

در حالیکه اکتشاف ها برای عملکرد بهتر روی یک مساله خاص طراحی شده اند، فرا اکتشاف ها الگوریتم های عمومی تری هستند که برای حل مسائل بهینه سازی طراحی شده اند. آنها استراتژی های سطح بالایی هستند که اکتشاف های خاص مساله را، برای یافتن یک راه حل نزدیک به بهینه، روی مساله اعمال می کنند. رویکردهای فرا اکتشافی در مقایسه با الگوریتم های مبتنی بر اکتشاف از نظر محاسباتی متمرکزتر و زمان بیشتری برای اجرا نیاز دارند. همینطور این رویکردها تمایل دارند تا هنگام جستجوی هدایت شده در بین جواب های مختلف، زمان بندی های مطلوب تری را پیدا کنند. استفاده از فرا اکتشاف برای حل مساله زمان بندی جریان کار در محیط ابری چالش هایی دارد از قبیل، مدل سازی نظری تعداد نامحدود منابع، تعریف عملگر هایی برای پیشگیری از جستجوی جواب های نامعتبر(مثلا نقض وابستگی داده) جهت تسهیل همگرایی و برش فضای حالت با استفاده از اکتشاف هایی روی مدل منبع ابری پایه [۳۲].

در[۳۳]، الگوریتمGA-ETI ارائه شده است که یک زمان بندی برای برنامه های کاربردی علمی روی سیستم های ابری است و به طور همزمان هزینه های مالی و زمان اجرای Makespan را بهینه می کند. GA-ETI، الگوریتم GA اصلی را از طریق اصلاح سفارشی/ هدفمند، به عملگرهای تقاطع و جهش خود، توسعه داده استGA-ETI .، عملگر تقاطع پیشرفته ای را برای ترکیب خوشه هایی از ژن به جای کروموزوم های تقسیم شده تصادفی، به کار می‌برد. همچنین جهش های افزایشی/ کاهشی را برای حذف/ اضافه ماشین های مجازی از یک کروموزوم داده شده به کار می برد. هردوی این اصلاحات تصادفی بودن ذاتی را در مقایسه با GA اصلی کاهش می‌دهند. GA-ETI، با استفاده از پنج جریان کاری که برای نشان دادن انواع مسائل علمی اخیر به کار می روند، مورد آزمایش قرار گرفته و برتری خود را

در مقابل سه زمان بند به روز و معروف (HEFT، Provenance و FSV) در این زمینه نشان داده است. GA-ETI، در مقایسه با راه حل های ارائه شده توسط HEFT ، زمان Makespan و هزینه مالی کمتری را نشان داده است.

در[۳]، نویسندگان یک الگوریتم زمان بندی چندمنظوره برای زمان بندی جریان کار در محیط های ابری چندگانه ارائه کرده‌اند. هدف حداقل کردن Makespan و هزینه جریان کاری، تحت محدودیت قابلیت اطمینان است. به علت ضرائب شکست متفاوت در چارچوب های ابری، برای کاربران ابر بسیار مهم است که به منظور قابلیت اطمینان کلی، به نگاشت انواع ماشین های مجازی به وظایف در جریان کاری، توجه داشته باشند. برای حل مساله اشاره شده در بالا، الگوریتم MOS بر اساس بهینه سازی ازدحام ذرات پیشنهاد شده که مکان اجرای وظایف و ترتیب وظایف مربوط به انتقال داده را همزمان در نظر می گیرد. نتایج شبیه سازی بر مبنای ساختارهای جریان کار علمی دنیای واقعی نشان داده که الگوریتم MOS روی همه پارامترهای کارایی چندمنظوره، نسبت به الگوریتم های CMOHEFT و RANDOM، عملکرد بهتری دارد .

در[۳۳]، نویسندگان، مساله زمان بندی ایستا را برای جریان های کاری دنیای واقعی روی سیستم های ابری در نظر می‌گیرند. آنها سعی در حداقل کردن هزینه اجرا دارند در حالی که محدودیتMakespan را ارضا کند. الگوریتم آنها می‌تواند به صورت انطباق پذیر منابع محاسباتی اشغال شده در طول فرایند زمان بندی را به صورت تکراری تنظیم کند. PSO در هر تکرار به کار گرفته می شود و هنگامی که تعداد ماشین های مجازی استفاده شده پایدار شد، حلقه متوقف می شود. این الگوریتم از وابستگی به واکشی منابع اولیه رها شده است. علاوه بر این روش صورت حساب جزئی برای مطابقت با روند بازار در آن اعمال می شود.

در [۲]، نسخه پیشرفتهWFSACO، برای زمان بندی جریان کار با استفاده از ACO ارائه شده است که Makespan را کاهش می دهد و همزمان از پیچیدگیWFSACO میکاهد. در ابتدا وظایف در هر سطح از جریان کاری، بر مبنای طول وظیفه و تعداد فرزند مرتب می‌شوند. وظایف مرتب شده به وسیله ACO به منابع محاسباتی نگاشت می شوند. برای حداقل کردن زمان Makespan، سطوح فرومون و داده های اکتشافی برای هر ماشین، برمبنای احتمال انتقال، مطابق با فرض رویکرد پیشنهادی، به روز رسانی می شوند. مقایسه نتایج تجربی رویکرد پیشنهادی با الگوریتم های موجود دیگر انجام می‌شود که با پارامتر هایی نظیر مدت زمان بندی و هزینه منابع، کار می کنند.

در[۳۴]، روش فرا اکتشافی جدیدی مبتنی بر بهینه سازی ازدحام ذرات گسسته دودویی گربه (DBCSO[1]) ارائه شده است که سعی دارد در هنگام اجرای برنامه های کاربردی مبتنی بر جریان کار روی محیط ابری، یک زمان Makespan بهینه را به دست آورد. نتایج شبیه سازی نشان می دهد که روش DBCSO زمان Makespan بهینه ای را ارائه می دهد و همچنین برای تعداد زیاد وظایف نسبت به بهینه سازی ازدحام ذرات استاندارد(PSO[2]) و بهینه سازی ازدحام ذرات باینری(BPSO[3])، عملکرد بهتری دارد.

در [۳۵]، نویسندگان الگوریتم برنامه نویسی بیان ژن چند منظوره مبتنی بر شاخص(IBM-GEP[4]) را ارائه کرده اند. در IBM-GEP، وظایف با یک روش مرحله به مرحله به منابع تخصیص داده میشوند. برای هر جفت "منبع و وظیفه "، یک مقدار اولویت توسط یک تابع اولویت محاسبه می شود. در هر مرحله، زوجی که بالاترین اولویت را داشته باشد، انتخاب می‌شود. برای جستجوی بهترین تابع اولویت،

[1] Discrete binary cat swarm optimization
[2] particle swarm optimization
[3] binary particle swarm optimization
[4] Indicator based Multi-objective Gene Expression Programming

در ابتدا چندین اکتشاف سطح پایین تعریف می شوند و به عنوان بلوک های ساختاری برای ساخت اکتشاف های نهایی، مورد استفاده قرار می‌گیرند. سپس تکنیک بهینه سازی چند منظوره مبتنی بر شاخص را یک با GP متنوع که اخیراً منتشر شده (SL-GEP نامیده می شود)، ادغام می کنند تا به جستجوی اکتشاف هایی با توازن های متفاوتی بین هزینه اجرا و زمان اجرای جریان کار بپردازد.

در [۳۶]، نویسندگان یک رویکرد الگوریتم ژنتیک جدیدPEFT ،را برای کاهش بیشتر زمان اجرای این مدل ارائه کرده‌اند. یک استراتژی توسعه داده شده تا به GA اجازه دهد، روی بهینه کردن هدف کروموزوم برای تولید مناسب ترین فرزند جهش یافته تمرکز کند. بعد از به دست آوردن یک راهکار عملی، الگوریتم ژنتیک روی بهینه کردن زمان اجرا تمرکز می کند. نتایج نشان داده که PEFTGA از نظر Makespan عملکرد بهتری برای زمان بندی وظایف روی ماشین های مجازی دارد. زمان تکمیل(Makespan) الگوریتم PEFTGA ارائه شده، کاهش ۲۵ درصدی را در مقایسه با GA استاندارد نشان می‌دهد.

در [۳۷]، نویسندگان یک الگوریتم زمان بندی جریان کار چند منظوره ممتاز تطبیقی(APMWSA[1]) برای محیط های ابری ارائه کرده اند. این روش اهدافی مانند Makespan و هزینه را بهینه می کند. اکثر کارهای موجود، یک تابع تناسب واحد را به کار می برند. آنها محاسبه روی تابع تناسب را روی هر دو معیار Makespan و هزینه انجام می دهند. مشاهده شده که APMWSA، نتایج بهتری نسبت به برخی الگوریتم های زمان بندی جریان کار موجود به دست آورده است. این روش همچنین بار کاری روی منابع محاسباتی را به وسیله توزیع وظایف روی منابع موجود، متوازن می کند. بدنبال عدم وجود یک رویکرد زمان بندی پایه برای زمان بندی جریان کار در محیط ابری، الگوریتم بهینه سازی ازدحام ذرات ترکیبی همه منظوره (HPSO[2]) در [۳۸] ارائه شده است. این الگوریتم، ترکیبی از الگوریتم بهینه

[1] adaptive privileged multi-objective workflow scheduling algorithm
[2] Hybrid Particle Swarm Optimization

سازی ازدحام ذرات همه منظوره و اکتشاف مرتبط با شاخص است. اجرای این الگوریتم به وسیله سه هدف که عبارتند از makespan، مصرف انرژی و هزینه کلی، تحت محدودیت های مهلت و بودجه ارزیابی شده است.

در [۳۹]، الگوریتم جهش ترکیبی قورباغه تکمیل شده (ASFLA[1]) ارائه شده که یک روش ساخت یافته برای زمان‌بندی منابع و زمان بندی جریان کار در محیط ابری از نوع زیرساخت به عنوان سرویس(IaaS) است. نتایج آزمایش نشان می دهد که ASFLA، در کاهش هزینه کارایی کلی برای جریان‌های کاری در نظر گرفته شده، نسبت به روشهای دیگر بهتر عمل می کند. جدول ۳-۲ خلاصه از تکنیک های مبتنی بر فرا اکتشافی را نشان می دهد.

جدول ۳-۲- خلاصه از تکنیک های مبتنی بر فرا اکتشافی

#	عنوان	روش	مزیت
۱	GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments	در[۳۳]، الگوریتمGA-ETI ارائه شده است که یک زمان بندی برای برنامه های کاربردی علمی روی سیستم های ابری است و به طور همزمان هزینه های مالی و زمان اجرای Makespan را بهینه می کند.	GA-ETI، در مقایسه با راه حل های ارائه شده توسط HEFT ، زمان Makespanو هزینه مالی کمتری را نشان داده است.
۲	An adaptive PSO-based real-time workflow scheduling algorithm in cloud systems	در[۳۳]، نویسندگان، مساله زمان بندی ایستا را برای جریان های کاری دنیای واقعی روی سیستم های ابری در نظر می‌گیرند.PSO در هر تکرار به کار گرفته می شود	آنها سعی در حداقل کردن هزینه اجرا دارند در حالی که محدودیتMakespan را ارضا کند.
۳	Multi-objective scheduling for scientific workflow in multicloud environment	در[۳]، نویسندگان یک الگوریتم زمان بندی چندمنظوره برای زمان بندی جریان کار در محیط های ابری چندگانه ارائه کرده‌اند. هدف	الگوریتم MOS روی همه پارامترهای کارایی چندمنظوره، نسبت به الگوریتم های

[1] Augmented Shuffled Frog Leaping Algorithm

		حداقل کردن Makespanو هزینه جریان کاری، تحت محدودیت قابلیت اطمینان است.	CMOHEFTو RANDOM، عملکرد بهتری دارد .
	Discrete binary cat swarm optimization for scheduling workflow applications in cloud systems	در[۳۴]، روش فرا اکتشافی جدیدی مبتنی بر بهینه سازی ازدحام ذرات گسسته دودویی گربه (DBCSO) ارائه شده است	روش DBCSOزمان Makespan بهینه ای را ارائه می دهد و همچنین برای تعداد زیاد وظایف نسبت به بهینه سازی ازدحام ذرات استاندارد(PSO) و بهینه سازی ازدحام ذرات باینری(BPSO)، عملکرد بهتری دارد .
۶	Indicator-based multi-objective genetic programming for workflow scheduling problem	در [۳۵]، نویسندگان الگوریتم برنامه نویسی بیان ژن چند منظوره مبتنی بر شاخص(IBM-GEP) را ارائه کرده اند.	به جستجوی اکتشاف هایی با توازن های متفاوتی بین هزینه اجرا و زمان اجرای جریان کار می پردازد.
۷	Time Effective Workflow Scheduling using Genetic Algorithm in Cloud Computing	در [۳۶]، نویسندگان یک رویکرد الگوریتم ژنتیک جدید PEFT ،را برای کاهش بیشتر زمان اجرای این مدل ارائه کرده‌اند.	زمان تکمیل(Makespan) الگوریتم PEFTGA ارائه شده، کاهش ۲۵ درصدی را در مقایسه با GA استاندارد نشان می‌دهد.
۹	A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling	الگوریتم بهینه سازی ازدحام ذرات ترکیبی همه منظوره (HPSO) در [۳۸]ارائه شده است.	اجرای این الگوریتم به وسیله سه هدف که عبارتند از makespan، مصرف انرژی و هزینه کلی، تحت محدودیت های مهلت و بودجه ارزیابی شده است.
۱۰	Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm	در [۳۹]، الگوریتم جهش ترکیبی قورباغه تکمیل شده (ASFLA) ارائه شده که یک روش ساخت یافته برای زمان‌بندی منابع و زمان بندی جریان کار در محیط ابری از نوع زیرساخت به عنوان سرویس(IaaS) است.	ASFLA، در کاهش هزینه کارایی کلی برای جریان‌های کاری در نظر گرفته شده، نسبت به روشهای دیگر بهتر عمل می کند.

فصل چهارم

راهکار پیشنهادی

۱-۴- مقدمه

رشته های علمی مختلف، از جریان های کاری برای تجزیه و تحلیل مقادیر بزرگی از داده و اجرای آزمایش ها و شبیه سازی های پیچیده به طور کارآمد استفاده می‌کنند. فرآیند می‌تواند به وسیله تقسیم شدن به زیر فرآیند های کوچک تر و ساده تر (یعنی وظایف)، به وسیله جریان کاری مدل سازی شود. سپس این وظایف برای اجرای سریع تر و کارآمد تر، روی چندین منبع محاسباتی توزیع می شوند. زمانبندی وظایف در جریان های کاری روی چارچوب های توزیع شده، طی سال ها به طور گسترده مورد مطالعه قرار گرفته است [۲۳]. چالش های بهینه سازی هزینهSWFS ، در محاسبات ابری یک مسئله آگاه به هزینه چندمنظوره است که نیازمند توجه به سه جنبه اصلی دارد [۴۰]:

(i) کاربران مختلف که معمولا برای منابع موجود در محاسبات ابری جهت ارضای محدودیت های QoS رقابت می کنند.

(ii) وابستگی های متقابل بین وظایف در جریان کاری

(iii) هزینه ارتباطات بالا به علت وابستگی های متقابل بین وظایف.

به هر حال در نظر گرفتن تمامی جنبه های مربوط به بهینه سازی هزینه، فرایند SWFS را پیچیده تر و همچنین از نظر زمان محاسبات نیازمند مقادیر بیشتری از منابع محاسباتی نموده است.

هدف اصلی الگوریتم های زمانبندی بدست آوردن عملکرد محاسباتی بالا و بهترین توان عملیاتی سیستم است. در مسائل چند هدفه مانند ابر چندین راهکار وجود دارد:

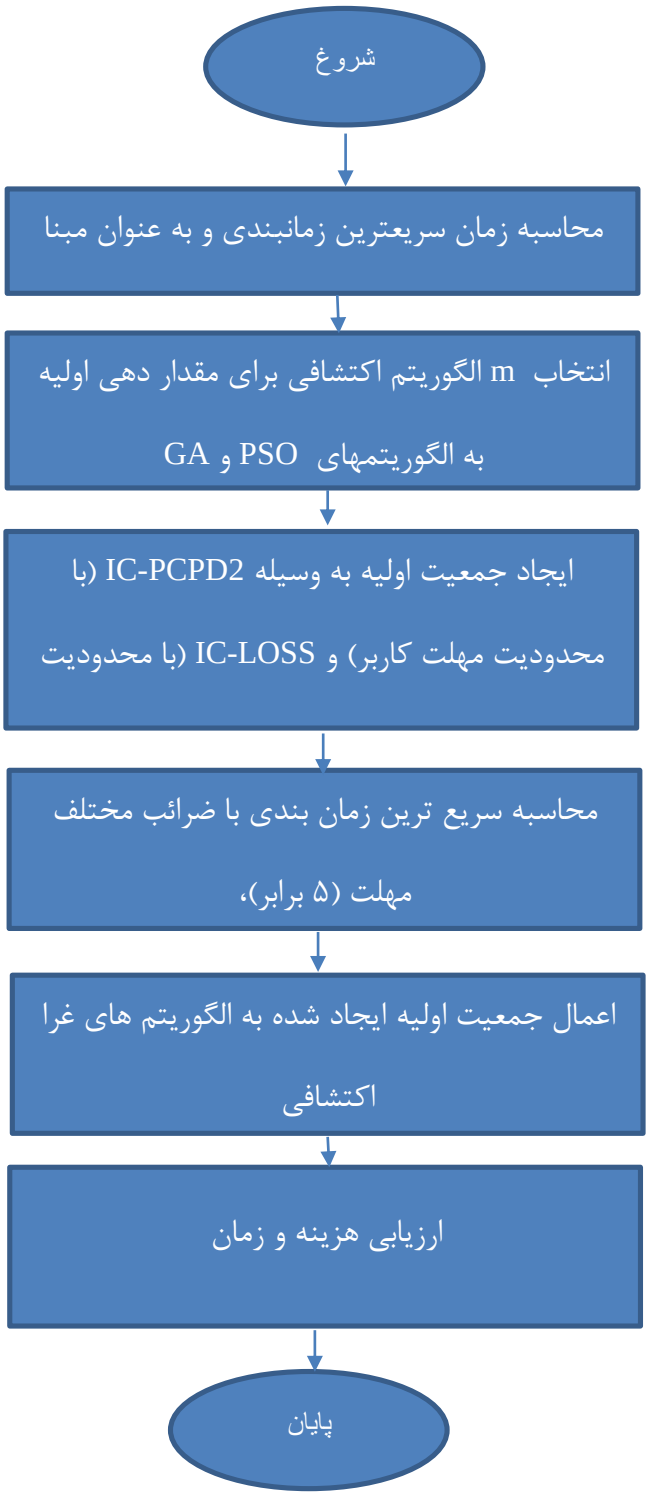
- انتخاب یک معیار به عنوان هدف اصلی و تبدیل سایر معیارها به محدودیت های زمانبندی، مثلا حداقل کردن هزینه در مهلت زمانی مشخص

- تعیین وزن برای هر معیار و ایجاد یک تابع

در این پایان نامه یک الگوریتم ترکیبی برای زمانبندی جریان کاری علمی در محاسبات ابری با استفاده از تقویت جمعیت اولیه الگوریتم های فرا اکتشافی به کمک الگوریتم های اکتشافی ارائه شده است. محدودیتی که در این پژوهش داریم realtime بودن هست. به عبارتی اگر بودجه ۴ دلار مشخص شده باشد کاربر نمی تواند بیشتر از ۴ دلار هزینه کند و یا برای زمان به عنوان مثال مهلت زمانی ۱۰ ثانیه باشد کاربر نمی تواند بیشتر از این زمان تعیین شده استفاده کند. در این روش از دو الگوریتم اکتشافی برای تولید جواب های خوب استفاده شده است و این جوابها بجای جوابهای تصادفی در جمعیت اولیه قرار داده می شوند ما قصد داریم با تولید جمعیت اولیه بهتر، هزینه اجرا و زمان اجرا را حداقل کنیم. به عبارتی دیگر این پایان نامه یک روش جدید با استفاده همزمان از شش الگوریتم ارائه می دهد (چهار فرااکتشافی :CTDHH[4]، HCHS [40]، PSO و GA و دو اکتشافی :IC-PCPD2 و IC-Loss).

۲-۴- الگوریتم پیشنهادی

ایده مطرح شده در [۴، ۴۰] این است که برخی الگوریتم های فرا اکتشافی را استفاده کند و هر یک از آنها را برای اجرا در هر مرحله به کار گیرد. یکی از مسائل موجود در [۴] مرتبه زمانی بالای آن به علت عدم توجه به جمعیت اولیه در الگوریتم ها است. برای حل این مسئله، الگوریتم پیشنهادی از الگوریتم های اکتشافی برای بهبود جمعیت اولیه در این الگوریتم ها استفاده می کند تا به نتایج بهتر و در یک مرتبه زمانی کوتاه‌تر دست پیدا کند. بدین منظور الگوریتم‌های IC-PCPD2 و IC-Loss برای ایجاد جمعیت اولیه مورد استفاده قرار گرفته‌اند. فلوچارت روش پیشنهادی در شکل ۴-۱ نشان داده شده است.



شکل ۴-۱- فلوچارت روش پیشنهادی

الگوریتم عمومی در روش پیشنهادی بدین صورت است:

الگوریتم پیشنهادی برای کارکرد صحیح نیازمند دو ورودی اصلی است. ورودی ها عبارتند از

(i) W=(T,E) ، که در آن(راس ها)Tمجموعه‌ای از وظایف است و(یال ها) E مجموعه یال های جهت دار بین وظایف است

(ii) H، مجموعه ای از الگوریتم های فرا اکتشافی (بهینه سازی ازدحام ذرات (PSO) و الگوریتم ژنتیک(GA)) است.

همانگونه که در الگوریتم ۱ (شکل ۴-۲) نشان داده شده است، الگوریتم پیشنهادی از m الگوریتم فرا اکتشافی استفاده کرده است. در ابتدا زمان سریعترین زمانبندی ممکن محاسبه و به عنوان مبنا در نظر گرفته می شود (خط ۱) سپس m الگوریتم اکتشافی برای مقدار دهی اولیه به الگوریتمهای PSO و GA انتخاب می‌شوند.به عبارت دیگر جمعیت اولیه با استفاده از دو الگوریتم که عبارتند از IC-PCPD2 (با محدودیت مهلت کاربر) و IC-LOSS (با محدودیت بودجه کاربر)، ایجاد می شود. با ایجاد جمعیت اولیه به وسیله این الگوریتم ها، زمان اجرا در الگوریتم پیشنهادی کاهش یافته و سپس برای ضرائب مختلف مهلت، سریع ترین زمان بندی محاسبه می شود(ضریب ۵ برابرسریعترین زمان بندی است). زمانبندی برای هر m الگوریتم انتخاب شده محاسبه می شود (خطهای ۲ تا ۱۰). در نهایت الگوریتم‌های فرا اکتشافی که PSO و GA درنظر گرفته شده‌اند، روی جمعیت اولیه که در مرحله قبل ایجاد شده اجرا می شود و در آخر الگوریتم DHHA در مقاله پایه اجرا می شود.

الگوریتم ۱: الگوریتم زماننبدی ترکیبی M - بعدی

Input: W=(T,E), T=$\cup_{i=1}^{n} T_i$, E={(T_i ,T_j , $Data_{ij}$)|(T_i,T_j)∈ $T \times T$}, E_{ij}=(T_i,T_j,$data_{ij}$), H(the set of the Meta-Heuristic algorithms: GA, PSO, CTDHH , HCHS), d is the user deadline and c is user budget for w.
Output: The most optimal solution for cost optimization of w

1: Create initial population // initial population make with Heuristic algorithms (IC- PCPD and IC-Loss)
2: α ← TimeOfFS (w) // α is the time of Fastest Schedule of w
3: for each a_n n ∈ {1,…,m} do // select m Meta-Heuristic algorithm from initializing
4: a_n ← Ø;
5: end for
6: for each a_n n ∈ {1,…,m} do // compute results for each Meta-Heuristic algorithm
7: for i←1 to 5 do // compute the algorithms for different deadline factors
8: insert a_n(w,c,d) into a_n
9: α← α*i
10: end for
11: end for
12: for 1 to 5 do
13: Run h with initial population a_n n ∈ {1,…,m} ∀h ∈ H
14: end for
15: compute DHHA (w)

شکل ۴-۲- الگوریتم پیشنهادی

۱-۲-۴- جمعیت اولیه با استفاده از الگوریتم IC-PCPD

IC-PCPD2 [۴۱]، یک الگوریتم دو مرحله‌ای است که ساختاری مشابه PCP، با دو تفاوت اساسی دارد. اول آنکه سه مسیر سیاست های تخصیص، با یک سیاست واحد جایگزین شده تا با مدل قیمت گذاری جدید سازگار شود. دوم مرحله برنامه ریزی به گونه‌ای اصلاح می شود که یک وظیفه را زمان بندی کند بدین ترتیب که ابتدا سعی می‌کند زمانی باقی مانده از نمونه‌های موجود سرویس‌های محاسباتی را، مورد استفاده قرار دهد و در صورت شکست، یک نمونه جدید را برای اجرای وظیفه، قبل از زیر مهلت مربوط به آن، راه اندازی می کند. از طرف دیگر ICPCP، یک الگوریتم یک مرحله‌ای است که

یک استراتژی مشابه مرحله توزیع مهلت در PCP را به کار می برد اما به جای تخصیص زیر مهلت به وظایف موجود در یک مسیر بحرانی نسبی، تلاش می کند تا به وسیله یافتن یک نمونه سرویس محاسباتی (موجود یا جدید) که می تواند کل مسیر را قبل از آخرین زمان خاتمه آن اجرا کند، آنها را به صورت واقعی زمانبندی کند.

۲-۲-۴- یک جمعیت اولیه با استفاده از الگوریتم IC-LOSS

IC-LOSS، یک نسخه اصلاح شده از الگوریتم Loss [۴۲] برای محیط ابری است. الگوریتم Loss، برای زمانبندی جریان کار در محیط گرید ارائه شده و در مرحله زمان بندی مجدد، وظایف مبادله شده بین منابع را با کمترین اتلاف وزن ، حفظ میکند. از آنجا که در ابرهای IaaS، زمان بندی مجدد وظایف واحد روی یک ماشین ارزان تر ممکن است هزینه اجرای کلی را افزایش دهد، لذا مرحله زمان بندی مجدد در الگوریتم Loss ، بایستی سازگار شود. IC-LOSS، تلاش می کند تا همه وظایف یک نمونه را روی یک نمونه جدید یا موجود ارزانتر، زمان بندی مجدد کند تا هزینه اجرای کلی را کاهش دهد. با توجه به اینکه الگوریتم IC-LOSS الگوریتمی است که با توجه به بودجه داده شده توسط کاربر، زمان بندی بهینه را فراهم می‌کند از این الگوریتم برای ایجاد جمعیت اولیه استفاده شده است.

۳-۲-۴- رویکرد فرا اکتشاف مشتق شده از زمان تکمیل CTDHH

CTDHH [۴]، برای چالش بهینه سازی هزینه در SWFS روی محیط ابری ارائه شده است. این الگوریتم به عنوان تکنیک پیشرفته ای در نظر گرفته می شود که قادر به تسریع زمان اجرای الگوریتم فرا اکتشاف است. CTDHH، استراتژی اکتشافی سطح بالا (HLH[1]) را با کمک چهار الگوریتم فرا اکتشافی مبتنی

[1] High Level Heuristic

بر جمعیت معروف،که به عنوان اکتشاف سطح پایین (LLH[1]) عمل می کنند، به کار می‌برد. هدف اصلی استراتژی HLH آن است که به طور هوشمندانه ف فرایند جستجو را بر اساس کارآیی الگوریتم های فرا اکتشاف LLH به کار گرفته شده، هدایت کند. کارایی رویکرد CTDHH، به وسیله مقایسه آن با چهار رویکرد مبتنی بر جمعیت(یعنی : GA، PSO، IWO و HIWO) و یک رویکرد فرا اکتشاف موجود، که الگوریتم زمانبندی فوق اکتشافی نامیده می شود، به طور گسترده مورد ارزیابی قرار گرفته است. بر اساس کمترین زمانی تکمیل بدست آمده، رویکرد پیشنهادی به صورت پویا فرآیند های جستجو را هدایت می کند تا یک راه حل بهینه را به وسیله مرتب سازی پیوسته نمرات زمانی محاسبه شده(یعنی زمان های تکمیل از اجراهای قبلی) مربوط به همه الگوریتم های LLH برای هر سناریوی در نظر گرفته شده و بعد از هر اجرا پیدا کند. در نتیجه مکانیزمی فرا اکتشاف مشتق شده از زمان تکمیل، موثر تر عمل کرده و اجازه استفاده مجدد و به کارگیری حداکثر قدرت الگوریتمهای LLH به کار گرفته شده در جستجو را برای راه حل بهینه،از مسئله بهینه سازی هزینه هدف، می دهد. بر اساس نتایج، هزینه محاسبات کلی در این رویکرد، در مقایسه با رویکردهای پایه به ارزانترین هزینه محاسبه کلی دست یافته است. به هرحال این نتایج بر اساس اندازه و نوع SWFS تحت تاثیر قرار می‌گیرند. عمدتاً به دلیل اندازه بزرگ و پیچیدگی وظایف جریان کار SWFS ارسال، در نهایت رویکردهای SWFS برای اجرای این وظایف زمان طولانی تری نیاز دارند. این به دلیل این واقعیت است که وظایف Montage SWFA محدودیت های اولویت کمتری دارند. بنابراین رویکرد CTDHH برای اغلب دیتاست های SWFA و برای اغلب سناریوهای در نظر گرفته شده در مقایسه با رویکردهای پایه و HHSA به نتایج بهینه تری دست یافته است. روش پیشنهادی ما، این الگوریتم را برای ایجاد جمعیت اولیه مناسب مورد استفاده قرار داده است.

[1] Low Level Heuristic

۴-۲-۴- رویکرد ترکیبی زمانبندی ترکیبی با هزینه موثر HCHS

رویکرد ترکیبی زمانبندی ترکیبی با هزینه موثر HCHS [۴۰]، برای ارائه دهندگان سرویس است تا ماشین های مجازی ارائه دهندگان ابر را برای پردازش بارهای کاری ترکیبی اجاره دهند و در عین حال زمان پاسخ لازم برای سرویس های تعاملی و مهلت زمانی کارهای دسته ای را برآورده سازند. هدف اصلی این مقاله ارائه یک رویکرد جدید است که یک الگوریتم فرا اکتشافی همانند، فرا اکتشاف مشتق شده از زمان تکمیل(CTDHH) را به همراه دو الگوریتم اکتشافی از قبیل IC-PCP و IC-Loss، به کار می‌برد. الگوریتم پیشنهادی به گونه‌ای است که الگوریتم های اکتشافی را برای بهبود جمعیت اولیه این الگوریتم ها برای دستیابی به نتایج بهتر در یک مرتبه زمانی کوتاه تر مورد استفاده قرار دهد. برای این منظور الگوریتمهای ic-loss و ic-pcp برای ایجاد جمعیت اولیه مورد استفاده قرار گرفته‌اند.

۴-۲-۵- رویکردی الگوریتم ژنتیک(GA)

رویکرد الگوریتم ژنتیک[۴۳]،یکی از مهم ترین دسته ها در زمینه محاسبات تکمیلی است. این الگوریتم از ژنتیک جمعیتی تکامل یافته تا مسائل بهینه سازی پیچیده را حل کند و اصل ساختار ژنتیک مندل را برای سازماندهی کردن کروموزوم ها، ژن ها و آلل ها مورد استفاده قرار می‌دهد. این الگوریتم شامل دو عملیات اصلی به نام تقاطع و جهش است که برای یافتن راه حل های بهبود یافته استفاده می شوند. جمعیت دو فرد اصلی دارد که ژنوتیپ و فونوتیپ هستند. نسل های بعدی به وسیله فرآیند پیوند توسط ژنوم های مفرد انتخاب شده، تولید می شوند. عملگرهای تقاطع و جهش برای به دست آوردن راه حل های بهتر روی ژنوم های انتخاب شده اعمال می شوند که به عنوان siblings یا off springs برچسب زده می شوند. نتیجه اجرای این عملیات به صورت تکراری،یک راه حل انطباق پذیر تولید می کند. هدف از الگوریتم ژنتیک حداکثر نمودن بازده راه حل های کاندید جمعیت، در مقابل یک تابع هزینه از دامنه

مسئله است. استراتژی الگوریتم ژنتیک، به کار گیری مکرر جایگزین ها برای مکانیزم های ژنتیکی باز ترکیب و جهش، روی جمعیت راه حل های کاندید است. که در آن تابع هزینه(تحت عناوین تابع هدف یا تناسب نیز شناخته می‌شود) به یک نمایش رمزگشایی شده از یک کاندید اعمال می شود که مشارکت های احتمالی را کنترل می کند و می تواند نسل های بع از راه حل های کاندید را ایجاد کند.

۶-۲-۴- رویکرد بهینه سازی مبتنی بر ازدحام ذرات (PSO)

رویکردی بهینه سازی ازدحام ذرات(PSO) [۲۳]، از مهم ترین دسته تکنیک ها در زمینه هوش ازدحامی است. این الگوریتم مبتنی بر رفتار تغذیه اجتماعی پرندگان است. به عنوان مثال حرکت گروهی پرندگان و گروه ماهی ها. برای یافتن یک راه حل بهتر دو استراتژی اصلی ارائه شده اند: (i) موقعیت محلی و(ii) موقعیت سراسری. هر ازدحام موقعیت مکانی و سرعت خود را به طور پیوسته به دنبال عضو ممتاز، به روز رسانی می کند. هر ازدحام تمام تاریخچه مکانی را برای ارجاع آینده نگهداری میکند. هدف PSO، یافتن یک راه حل بهینه بر اساس تمام اطلاعات مکانی ازدحام در یک حجم بزرگ چند بعدی است. راه حل بهینه می‌تواند بر اساس تخصیص تصادفی موقعیت ها و سرعت ها به همه فضای مسئله برای ازدحام به دست آید. رویکرد PSO بر اساس افزایش موقعیت هر ازدحام بر اساس سرعتش به دست می آید. ایده بنیادی در رویکرد PSO، جستجوی فضای مسئله برای مشخص کردن بهترین موقعیت سراسری است،که برای ذره بعد از نمونه سازی هر به روز رسانی جدید مرتبط با موقعیت شناسایی می شود. در نتیجه ذرات با یکدیگر گروه شده یا به وسیله اکتشاف و به کاربردن موقعیت های برتر در فضای جستجو به سمت یک راه‌حل پیشرفته همگرا می‌شوند.

فصل پنجم

ارزیابی نتایج

۵-۱- محیط شبیه سازی

کلادســـیم نرم افزاری کلی اســـت که می توانیم برای مدل کردن محیط های رایانش ابری و تســـت کارایی سرویس های کاربردی استفاده کنیم. این شبیه ساز مبتنی بر زبان جاوا است و یک شبیه ساز مبتنی بر رویداد است . یعنی موجودیت هایی که در آن تعریف می شود، از طریق فرستادن رویداد با یکدیگر ارتبا ط برقرار می کنند [۴۸،۴۷]. ارائه دهندگان ســرویس ابرر برای این که ســرویســی را در سطح عالی به کاربران ارائه دهند با مسائلی از جمله تضمین کیفیت سرویس که می تواند زمان پاسخ کمتر و یا گذردهی بالاتر باشد، بهره وری کارای منابع، بارهای کاری پویا مانند مدل کردن کاربردهای وبی چون سرویس دهنده-سررویس گیرنده و نقض سطح توافق ارائه سرویس مواجه بوده اند. در این میان ، مشکل تست کردن از جمله مهمترین مسائل است زیرا این امکان وجود ندارد که وجود ندارد که آزمایش های محک را در محیط قابل مقیاس، قابل تکرار و قابل اعتماد با استفاده از یک محیط ابر واقعی انجام داد. پس کلید حل این مشکلات استفاده از ابزار شبیه سازی مانند کلادسیم است [۴۹]. ما هم در این پایان نامه برای پیاده سازی راهکار پیشنهادی از این شبیه ساز استفاده کردیم.

۵-۲- تنظیمات آزمایش

در آزمایش‌های انجام شده هزینه‌ی نرمال به صورت زیر محاسبه شده است:

$$NC = \frac{\text{total schedule cost}}{Cc}$$

که Cc هزینه‌ی اجرای ارزان‌ترین زمانبندی (زمانبندی تمامی گره‌ها بر روی نسخه‌ی مجزایی از ارزان-ترین خدمت موجود) است. زمان اجرای نرمال نیز به صورت زیر محاسبه می‌شود:

$$NM = \frac{schedule\ makespan}{M_H}$$

که M_H زمان اجرای سریع‌ترین زمانبندی (زمانبندی تمامی گره‌ها بر روی نسخه ی مجزایی از سریع-ترین خدمت موجود) است. مهلت پایانی در آزمایشات با ضریب α تغییر می‌کند به این صورت که مهلت پایانی از حاصلضرب α در زمان اجرای سریع‌ترین زمانبندی (زمانبندی تمامی گره‌ها بر روی نسخه‌ی مجزایی از سریع‌ترین خدمت موجود) به‌دست می‌آید.

۳-۵- ارزیابی نتایج

الگوریتم پیشنهادی به منظور مقایسه و ارزیابی الگوریتم های مقاله[۴، ۴۰] را مورد استفاده قرار داده است زیرا این الگوریتم های زمانبندی همان معیارهای الگوریتم پیشنهادی را دارند. برای این مقایسه ها پنج جریان کاری Inspiral، Montage، Epigenimics، Cybershake و SIPHT با کلاس‌های موجود در جدول (۵-۱) آزمایش شده‌اند.

جدول ۵-۱- کلاسهای جریان کاری

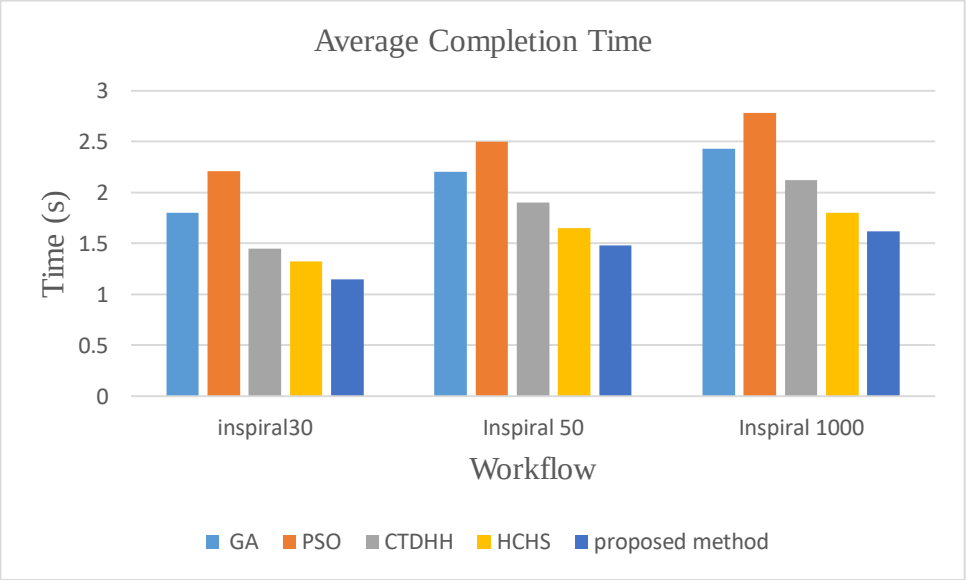
جریان کاری	بزرگ	کوچک
Montage	1000	25-50
Inspiral	1000	30-50
Epigenimics	997	24-46
Cybershake	1000	30-100
SIPHT	1000	30-50

همانگونه که در شکل های ۵-۱ تا ۵-۵ نشان داده شده، برای تمام جریان‌های کاری Inspiral، Montage، Epigenimics، Cybershake و SIPHT، کارایی الگوریتم پیشنهادی با الگوریتم های پایه (CTDHH ، HCHS ، GA وPSO) مقایسه شده است. بر مبنای دو الگوریتمی که ما در نظر گرفته‌ایم، آزمایش‌ها نشان می‌دهند که برای یک جریان کاری اگر چه کارایی دو الگوریتم IC-PCPD2 و IC-LOSS، در تولید جمعیت اولیه بهتر است اما الگوریتم پیشنهادی کارایی بهتری از الگوریتم‌های پایه

داشته است. الگوریتم تکاملی پیشنهادی، برای زمان‌بندی جریان‌های کاری ابری پایدارتر است و به نظر می‌رسد با احتمال بیشتری زمانبندی‌های قابل قبولی را تولید کند.

جدول ۵- ۲- نتایج آماری زمان تکمیل برای جریان کاری inspiral

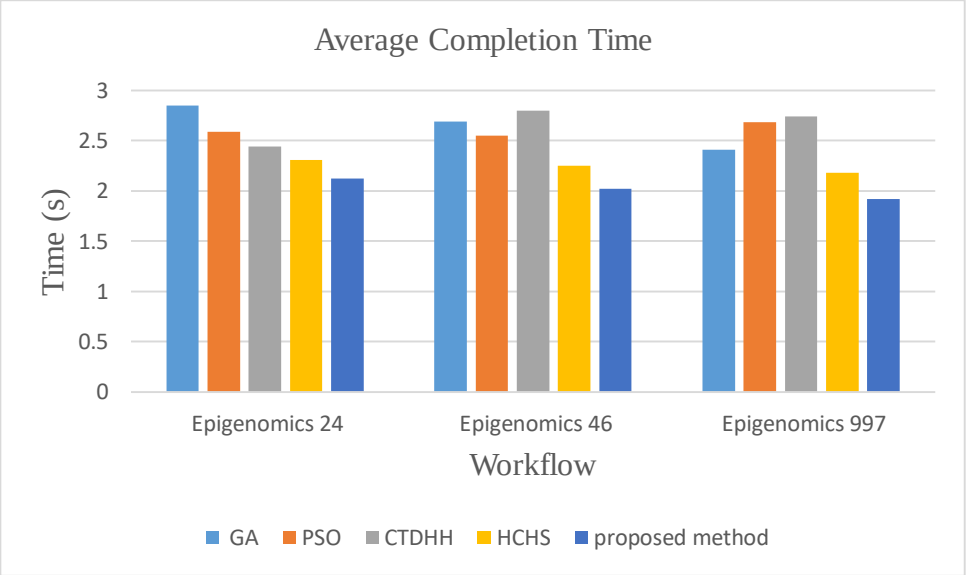
جریان کاری	GA	PSO	CTDHH	HCHS	روش پیشنهادی
inspiral30	1.8	2.21	1.45	1.32	1.15
Inspiral 50	2.2	2.5	1.9	1.65	1.48
Inspiral 1000	2.43	2.78	2.12	1.8	1.62



شکل ۵-۱- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری inspiral)

جدول ۵- ۳- نتایج آماری زمان تکمیل برای جریان کاری Epigenomics

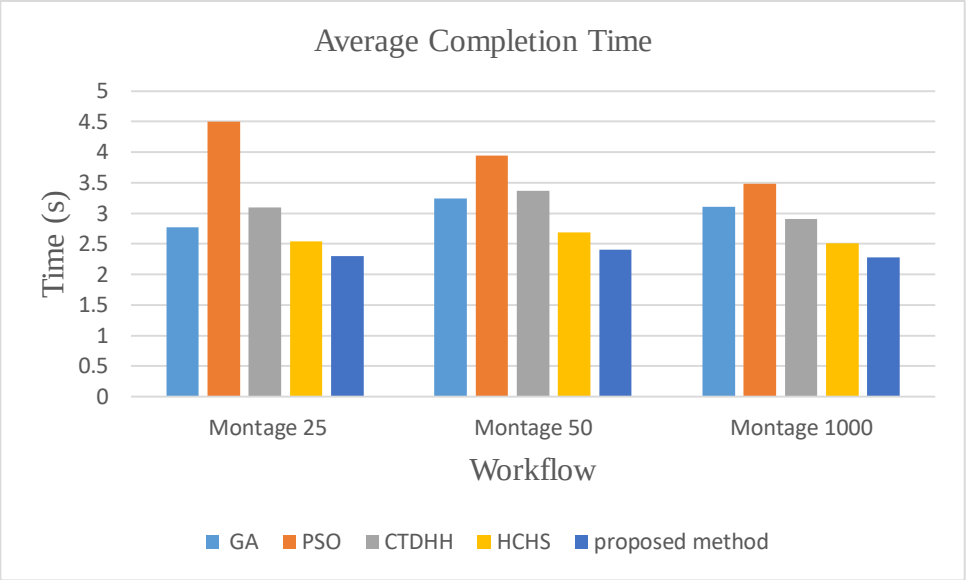
جریان کاری	GA	PSO	CTDHH	HCHS	روش پیشنهادی
Epigenomics 24	2.85	2.59	2.44	2.31	2.12
Epigenomics 46	2.69	2.55	2.8	2.25	2.02
Epigenomics 997	2.41	2.68	2.74	2.18	1.92



شکل ۵- ۲- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری Epigenomics)

جدول ۵-۴- نتایج آماری زمان تکمیل برای جریان کاری Montage

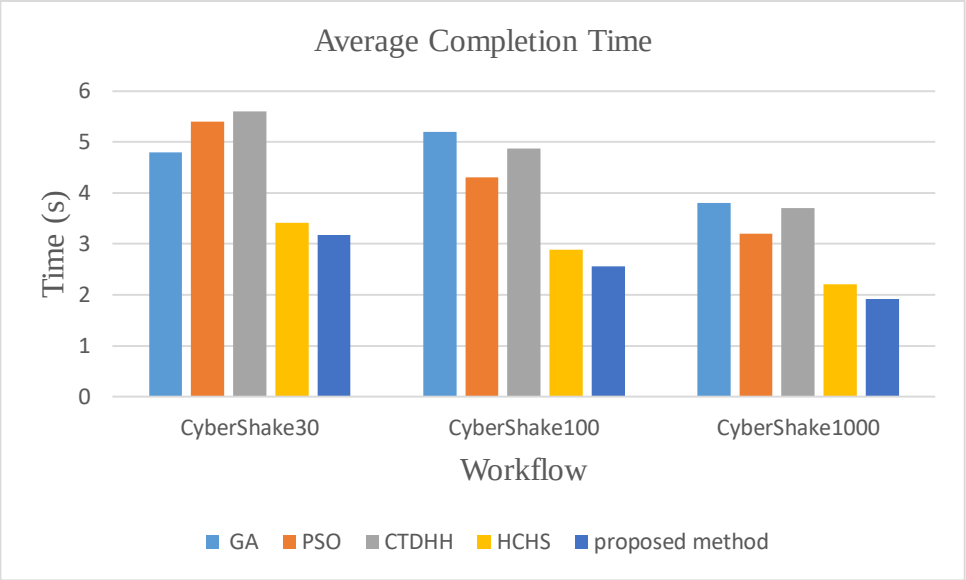
جریان کاری	GA	PSO	CTDHH	HCHS	روش پیشنهادی
Montage 25	2.77	4.5	3.1	2.54	2.3
Montage 50	3.24	3.94	3.37	2.69	2.41
Montage 1000	3.11	3.48	2.91	2.51	2.28



شکل ۵- ۳- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری Montage)

جدول ۵- ۵- نتایج آماری زمان تکمیل برای جریان کاری Cybershake

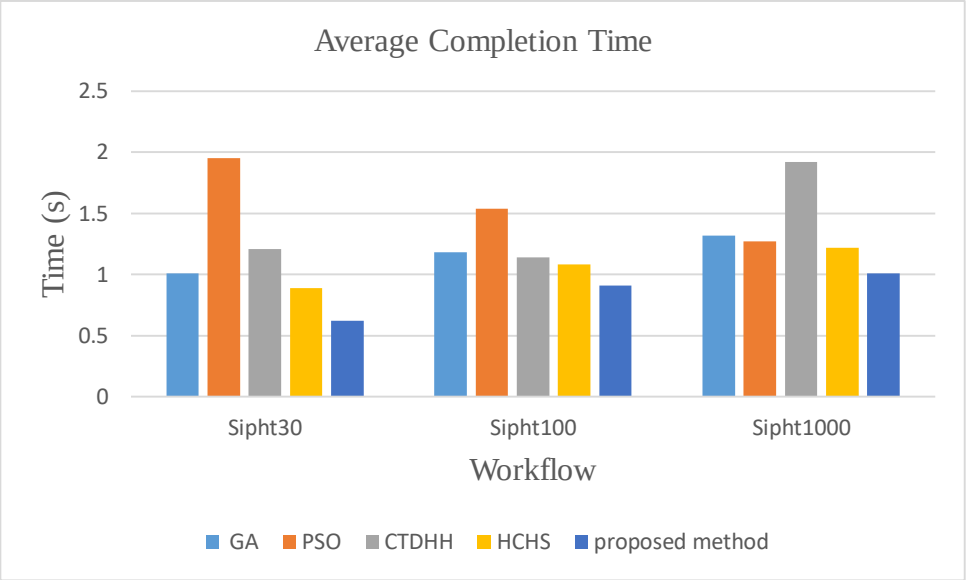
جریان کاری	GA	PSO	CTDHH	HCHS	روش پیشنهادی
CyberShake30	4.8	5.4	5.6	3.41	3.17
CyberShake100	5.2	4.3	4.87	2.89	2.56
CyberShake1000	3.8	3.2	3.7	2.21	1.92



شکل ۵- ۴- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری Cybershake)

جدول ۵- ۶- نتایج آماری زمان تکمیل برای جریان کاری SIPHT

جریان کاری	GA	PSO	CTDHH	HCHS	روش پیشنهادی
Sipht30	1.01	1.95	1.21	0.89	0.62
Sipht100	1.18	1.54	1.14	1.08	0.91
Sipht1000	1.32	1.27	1.92	1.22	1.01



شکل ۵- ۵- مقایسه الگوریتم پیشنهادی با الگوریتم های دیگر (جریان کاری SIPHT)

میانگین ارزش زمان تکمیل در روش پیشنهادی ، نسبت به میانگین زمان تکمیل CTDHH و HCHS، برای همه جریانهای کاری بهتر است. این امر به علت استفاده از الگوریتم های اکتشافی (IC-PCPD2 و IC-LOSS)، برای تولید جمعیت اولیه در الگوریتم های(CTDHH ، HCHS ، GA وPSO) است که برای یافتن بهینه ترین راه حل برای بهینه سازی هزینه جریانهای کاری در نظر گرفته شده‌اند. همانگونه که در جداول ۲-۵ تا ۶-۵ دیده می شود ، نتایج الگوریتم پیشنهادی از الگوریتم های دیگر بهتر است و مرتبه زمانی برای جریان کاری کاهش یافته است.

فصل ششم

نتیجه گیری

۱-۶- نتیجه گیری

جریان کاری مجموعه ای از وظایف به هم مرتبط می باشند که توزیع آنها بین منابع در دسترس گوناگون با هدف زمانبندی بهتر انجام می گیرد. امروزه جریان های کاری اغلب برای مدل کردن برنامه های کاربردی علمی در مقیاس بزرگ به کار می روند. جریان های کاری، توصیف بارهای کاری با وظیفه چندگامی را ساده می کند به طوری که فهم، خطایابی و نگهداری آن ساده باشند. با استفاده از جریان های کاری علمی چندین محقق می توانند روی طراحی یک برنامه کاربردی توزیع شده با یکدیگر همکاری کنند. این امر به این خاطر است که جریان های کاری به شکل گراف های بدون دور جهت دار طراحی می شوند که هر گره به تنهایی یک وظیفه است و یال ها وابستگی های بین وظایف را نشان می دهند. به طور کلی این وابستگی ها ورودی /خروجی هستند که بایستی بین وظایف منتقل شوند. با افزایش محبوبیت جریان های کاری در جامعه علمی، به منظور فراهم کردن محیط های اجرایی اختصاصی سیستم های مدیریتی اجرایی خاص منظوره با یکدیگر ادغام شده اند. سیستمی که می تواند جریان های کاری را روی کامپیوترهای رومیزی، کلاسترها و سیستم های مشبک یا ابرها اجرا کند. چنین اجرایی یک عمل مهم است مخصوصا روی ابرها که تامین منابع و رهاسازی آن ها، محاسبه هزینه و آماده کردن منابع نیز بایستی در نظر گرفته شوند. برخلاف کارهای زیادی که در زمینه زمانبندی جریان کاری انجام گرفته است.

یکی از رویکردهای زمان‌بندی، تمرکز بر کیفیت سرویس مورد نظر کاربر، در زمان‌بندی است. الگوریتم‌های زمانبندی جریان کار مبتنی بر کیفیت سرویس، معمولا تعدادی پارامتر کیفیت سرویس، از قبیل مهلت تکمیل وظیفه (و یا در کل، پارامترهایی در سطح توافق سرویس) را برای زمان‌بندی در نظر می‌گیرند. در این پایان نامه، ترکیبی از الگوریتم‌های اکتشافی و فرا اکتشافی برای زمانبندی جریان‌های کاری در محیط ابری ارائه شده است. به این ترتیب الگوریتم‌های اکتشافی (IC-PCPD2 و

IC-LOSS) برای ایجاد جمعیت و ورودی اولیه برای الگوریتم‌های تکاملی PSO، GA، CTDHH و HCHS مورد استفاده قرار گرفته‌اند. بر اساس نتایج به دست آمده، الگوریتم پیشنهادی، در بسیاری از موارد بهتر از الگوریتم های اولیه است.

۲-۶- کار آینده

در آینده ما تصمیم داریم تا الگوریتم پیشنهادی را برای جریان‌های کاری چندگانه و محیطهای ابری چندگانه توسعه دهیم.

مراجع

[1] Gupta, Ashish, and Ritu Garg. "Workflow scheduling in heterogeneous computing systems: A survey." In Computing and Communication Technologies for Smart Nation (IC3TSN), 2017 International Conference on, pp. 319-326. IEEE, 2017.

[2] Vinothina, V. "Scheduling scientific workflow tasks in cloud using swarm intelligence." In Current Trends in Advanced Computing (ICCTAC), 2017 IEEE International Conference on, pp. 1-5. IEEE, 2017.

[3] Hu, Haiyang, Zhongjin Li, Hua Hu, Jie Chen, Jidong Ge, Chuanyi Li, and Victor Chang. "Multi-objective scheduling for scientific workflow in multicloud environment." Journal of Network and Computer Applications 114 (2018): 108-122.

[4] Alkhanak, Ehab Nabiel, and Sai Peck Lee. "A hyper-heuristic cost optimization approach for Scientific Workflow Scheduling in cloud computing." Future Generation Computer Systems (2018).

[5] Soltani, Nasim, Behzad Soleimani, and Behrang Barekatain. "Heuristic algorithms for task scheduling in cloud computing: a survey." International Journal of Computer Network and Information Security 9, no. 8 (2017): 16.

[6] Hajibaba, Majid, and Saeid Gorgin. "A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing." CIT. Journal of Computing and Information Technology 22.2 (2014): 69-84.

[7] Fernando, Niroshinie, Seng W. Loke, and Wenny Rahayu. "Mobile cloud computing: A survey." Future Generation Computer Systems 29.1 (2013): 84-106.

[8] Hashemi, Seyyed Mohsen, and Amid Khatibi Bardsiri. "Cloud computing vs. grid computing." ARPN journal of systems and software 2, no. 5, pp.188-194, 2012.

[9] AlHakami, Hosam, Hamza Aldabbas, and Tariq Alwada'n. "Comparison between cloud and grid computing." International journal on cloud computing: services and architecture (IJCCSA) 2, no. 4, pp.1-21, 2012.

[10] Srinivas, J., K. Venkata Subba Reddy, and A. MOIZ Qyser. "Cloud computing basics." International journal of advanced research in computer and communication engineering 1, no. 5, pp.343-347 2012.

[11] Arya, Lokesh Kumar, and Amandeep Verma. "Workflow scheduling algorithms in cloud environment-A survey." In Engineering and Computational Sciences (RAECS), 2014 Recent Advances in, pp. 1-4, 2014.

[12] Qian, Ling, Zhiguo Luo, Yujian Du, and Leitao Guo. "Cloud computing: An overview." In IEEE International Conference on Cloud Computing, pp. 626-631, 2009.

[13] Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation computer systems 25, no. 6, pp.599-616, 2009.

[14] Singh, Lovejit, and Sarbjeet Singh. "A survey of workflow scheduling algorithms and research issues." International Journal of Computer Applications 74, no. 15, 2013.

[15] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).

[16] Marston, Sean, et al. "Cloud computing—The business perspective." Decision support systems 51.1 (2011): 176-189.
[17] Bhardwaj, Sushil, Leena Jain, and Sandeep Jain. "Cloud computing: A study of infrastructure as a service (IAAS)." International Journal of engineering and information Technology 2.1 (2010): 60-63.
[18] Yang, Jianfeng, and Zhibin Chen. "Cloud computing research and security issues." Computational intelligence and software engineering (CiSE), 2010 international conference on. IEEE, 2010.
[19] Popović, Krešimir, and Željko Hocenski. "Cloud computing security issues and challenges." Proceedings of the 33rd International Convention. IEEExplore, Opatija. 2010.
[20] Sakellariou, Rizos, and Henan Zhao. "A hybrid heuristic for DAG scheduling on heterogeneous systems." In Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International, p. 111, 2004.
[21] Zhao, Henan, and Rizos Sakellariou. "Scheduling multiple DAGs onto heterogeneous systems." In Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, pp. 14-pp, 2006.
[22] Yu, Zhifeng, and Weisong Shi. "A planner-guided scheduling strategy for multiple workflow applications." In Parallel Processing-Workshops, 2008. ICPP-W'08. International Conference on, pp. 1-8, 2008.
[23] Rodriguez, Maria Alejandra, and Rajkumar Buyya. "A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments." Concurrency and Computation: Practice and Experience 29, no. 8, pp. e404, 2017.
[24] Mohanapriya, N., G. Kousalya, and P. Balakrishnan. "Cloud workflow scheduling algorithms: a survey." Int J Adv Eng Technol 7, no. III, pp. 88-195, 2016.
[25] Casavant, Thomas L., and Jon G. Kuhl. "A taxonomy of scheduling in general-purpose distributed computing systems." IEEE Transactions on software engineering 14, no. 2, pp. 141-154, 1980.
[26] Yu, Jia, Rajkumar Buyya, and Kotagiri Ramamohanarao. "Workflow scheduling algorithms for grid computing." In Metaheuristics for scheduling in distributed computing environments, pp. 173-214, 2008.
[27] Arabnejad, Vahid, Kris Bubendorfer, and Bryan Ng. "Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources." Future Generation Computer Systems 75 (2017): 348-364.
[28] Sahni, Jyoti, and Deo Prakash Vidyarthi. "A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment." IEEE Transactions on Cloud Computing 6, no. 1 (2018): 2-18.
[29] Rimal, Bhaskar Prasad, and Martin Maier. "Workflow scheduling in multi-tenant cloud computing environments." IEEE Transactions on Parallel and Distributed Systems 28, no. 1 (2017): 290-304.
[30] Reddy, G. Narendrababu, and S. Phanikumar. "A novel method for scheduling workflows in cloud computing environment." In Science Technology Engineering & Management (ICONSTEM), 2017 Third International Conference on, pp. 12-16. IEEE, 2017.

[31] Arabnejad, Hamid, Jorge G. Barbosa, and Radu Prodan. "Low-time complexity budget–deadline constrained workflow scheduling on heterogeneous resources." Future Generation Computer Systems 55 (2016): 29-40.
[32] Casas, Israel, Javid Taheri, Rajiv Ranjan, Lizhe Wang, and Albert Y. Zomaya. "GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments." Journal of computational science 26 (2018): 318-331.
[33] Guo, Pengze, and Zhi Xue. "An adaptive PSO-based real-time workflow scheduling algorithm in cloud systems." In Communication Technology (ICCT), 2017 IEEE 17th International Conference on, pp. 1932-1936. IEEE, 2017.
[34] [34] Kumar, Bhopender, Mala Kalra, and Poonam Singh. "Discrete binary cat swarm optimization for scheduling workflow applications in cloud systems." In Computational Intelligence & Communication Technology (CICT), 2017 3rd International Conference on, pp. 1-6. IEEE, 2017.
[35] Xiao, Qin-zhe, Jinghui Zhong, Wen-Neng Chen, Zhi-Hui Zhan, and Jun Zhang. "Indicator-based multi-objective genetic programming for workflow scheduling problem." In Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 217-218. ACM, 2017.
[36] Rohit Nagar, Deepak K. Gupta and Raj M. Singh, " Time Effective Workflow Scheduling using Genetic Algorithm in Cloud Computing" Information Technology and Computer Science, 1, pp. 68-75, 2018.
[37] Singh, Ajit. "A multi-objective workflow scheduling algorithm for cloud environment." In 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), pp. 1-6. IEEE, 2018.
[38] Verma, Amandeep, and Sakshi Kaushal. "A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling." Parallel Computing 62 (2017): 1-19.
[39] Kaur, Parmeet, and Shikha Mehta. "Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm." Journal of Parallel and Distributed Computing 101 (2017): 41-50.
[40] Alhwayzee, A., "A Cost-effective Hybrid-Scheduling Algorithm for Scientific Workflow in cloud computing." Sci.Int. (Lahore), 30(4), 575-580, 2018.
[41] Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema. "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds." Future Generation Computer Systems 29, no. 1 (2013): 158-169.
[42] Sakellariou, Rizos, Henan Zhao, Eleni Tsiakkouri, and Marios D. Dikaiakos. "Scheduling workflows with budget constraints." In Integrated research in GRID computing, pp. 189-202. Springer, Boston, MA, 2007.
[43] Zhu, Zhaomeng, Gongxuan Zhang, Miqing Li, and Xiaohui Liu. "Evolutionary multi-objective workflow scheduling in cloud." IEEE Transactions on parallel and distributed Systems 27, no. 5 (2016): 1344-1357.
[44] Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and experience 41, no. 1 (2011): 23-50.

[45] Krishnadhan Das,” Extension of cloudsim: cloud computing simulator”, A Thesis Submitted in partial fulfillment of the requirements for the degree of Master of Technology under the guidance of Prof. Purushottam Kulkarni and Prof. Umesh Bellur.

[46] Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities." In High Performance Computing & Simulation, 2009. HPCS'09. International Conference on, pp. 1-11. IEEE, 2009.

[47] Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and experience 41, no. 1 (2011): 23-50.

[48] Krishnadhan Das,” Extension of cloudsim: cloud computing simulator”, A Thesis Submitted in partial fulfillment of the requirements for the degree of Master of Technology under the guidance of Prof. Purushottam Kulkarni and Prof. Umesh Bellur.

[49] Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities." In High Performance Computing & Simulation, 2009. HPCS'09. International Conference on, pp. 1-11. IEEE, 2009.

Abstract

Cloud computing is emerging with growing popularity in workflow scheduling, especially for scientific workflow. With the emergence cloud computing, can benefit from virtually unlimited resources with minimal hardware investment. Scheduling the submitted Scientific Workflow Application (SWFA) tasks to the available computational resources while optimizing the cost of executing the SWFA is one of the most challenging processes of Workflow Management System (WfMS) in a cloud computing environment. Several cost optimization approaches have been proposed to improve the economic aspect of SWFS in cloud computing. The main goal of the thesis is to present a new M-dimension hybrid algorithm, which uses a meta-heuristic algorithm such as Completion Time Driven Hyper-Heuristic (CTDHH), Hybrid Cost-effective Hybrid-Scheduling (HCHS), particle swarm optimization (PSO) and genetic algorithm (GA) and with using heuristic algorithms such as the IC-PCPD2 and IC-Loss algorithms. At the end, our proposed algorithm is compared with the basic algorithm in terms of runtime and cost, and the results show improved results. The proposed algorithm is more stable for scheduling cloud workflows and seems more likely to generate acceptable scheduling.

Keywords: Workflow, Cloud computing, Scheduling, Cost optimization, Meta-Heuristic and Heuristic.

Emam Reza University of Mashhad

Faculty of Engineering

Copmputer Engineering Department

MSc Dissertation

M-dimension hybrid algorithm for scientific workflow scheduling in cloud computing

By:

Zahra Al hamdavi

Supervisor:

Dr. Seyyed Majid Mazinani

January 2020