بسم الله الرحمن الرحيم

تعهدنامه

اینجانب سعید ابریشمی دانشجوی دوره دکتری رشته مهندسی کامپیوتر دانشکده مهندسی دانشگاه فردوسی مشهد، نویسنده پایان‌نامه زمانبندی مبتنی بر کیفیت خدمت جریان های کاری در سیستم های مشبک عمومی و ابرهای محاسباتی تحت راهنمایی جناب آقای دکتر نقیب زاده متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود و یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه فردوسی مشهد می باشد و مقالات مستخرج با نام "دانشگاه فردوسی مشهد" و یا "Ferdowsi University of Mashhad" به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند در مقالات مستخرج از رساله رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است، اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه فردوسی مشهد می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.

استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

دانشکده مهندسی

گروه کامپیوتر

رساله دکترا

زمانبندی مبتنی بر کیفیت خدمت جریان های کاری در سیستم های مشبک عمومی و ابرهای محاسباتی

سعید ابریشمی

استاد راهنما

دکتر محمود نقیب زاده

استاد مشاور

دکتر حسین دلداری

آذر ۱۳۹۰

تصویب نامه هیئت داوران

این رساله با عنوان **زمانبندی مبتنی بر کیفیت خدمت جریان های کاری در سیستم های مشبک عمومی و ابرهای محاسباتی** توسط **سعید ابریشمی** در تاریخ ۱۳۹۰/۹/۲۷ با نمره ۱۹/۷۱ و درجه ارزشیابی **عالی** در حضور هیات داوران با موفقیت دفاع شد.

هیات داوران

ردیف	نام و نام خانوادگی	مرتبه	سمت	دانشگاه	امضا
۱	دکتر محمود نقیب زاده	استاد	استاد راهنما	فردوسی مشهد	
۲	دکتر حسین دلداری	دانشیار	استاد مشاور	فردوسی مشهد	
۳	دکتر کامران زمانی فر	دانشیار	استاد مدعو خارجی	اصفهان	
۴	دکتر علی وحیدیان کامیاد	استاد	استاد مدعو داخلی	فردوسی مشهد	
۵	دکتر محمدرضا اکبرزاده	استاد	استاد مدعو داخلی	فردوسی مشهد	
۶	دکتر محمد حسین یغمایی	دانشیار	استاد مدعو داخلی	فردوسی مشهد	

قدردانی

در ابتدا لازم می دانم از کلیه اساتید عزیزی که در طی سالیان تحصیل در دانشگاه فردوسی مشهد از محضر آنها استفاده کرده ام، تشکر نمایم. از استاد راهنمای گرانقدرم، جناب آقای دکتر نقیب زاده به خاطر راهنمایی های ارزنده ای که در انتخاب موضوع، تدوین ونگارش مقالات و رساله حاضر داشتند، تشکر می نمایم. بی شک جناب آقای دکتر نقیب زاده جایگاهی بسیار بالاتر از یک استاد راهنما برای من داشته اند و سالها است که از راهنمایی های ایشان در مراحل مختلف زندگی برخوردار بوده ام. از جناب آقای دکتر دلداری نیز به خاطر زحماتی که در طی دوره دکتری و پیش از آن برای من کشیده اند، تشکر می نمایم. همچنین از جناب آقای دکتر کاهانی نیز به خاطر کمک هایی که در طی این دوره به من نمودند، سپاسگزارم.

در دوره فرصت تحقیقاتی در دانشگاه دلفت هلند نیز از کمک و محبت دوستان بسیاری برخوردار بوده ام. پیش از همه، باید از آقای دکتر Dick H.J. Epema به خاطر امکاناتی که در اختیارم گذاشتند، و کمک های ارزنده ای که در تدوین مقالات داشتند، تشکر نمایم. همچنین از سایر همکارانم در گروه سیستم های موازی و توزیع شده، آقایان Ozan Sonmez، Nezih Yigitbasi و Alexandru Iosup به خاطر کمک هایشان سپاسگزارم. از دوستانم در دانشکده انفورماتیک دانشگاه دلفت، آقایان علیرضا اسدی و محمود احمدی نیز به خاطر حمایت هایشان تشکر می کنم. بیش از همه، این دوره را مدیون دوست قدیمی و بسیار عزیزم، آرش استادزاده هستم که از کمک های بی دریغش در طی این دوره و سالیان دراز دوستیمان سپاسگزاری می نمایم.

همه ما همواره مدیون پدران و مادرانمان هستیم. اما در اینجا دوست دارم از پدر ومادرم، به خصوص به خاطر امکاناتی که بیدریغ در اختیارم گذاشتند تا بتوانم رشته مورد علاقه خود را دنبال کنم، و اعتمادی که همیشه به من داشتند تشکر کنم. از همسر عزیزم نیز به خاطر صبر و تحملش در دوران تحصیل من، و مهربانی و لطف همیشگیش سپاسگزارم.

لیست مقالات

[1] S. Abrishami, M. Naghibzadeh, and D.H.J. Epema, “Cost-driven Scheduling of Grid Workflows Using Partial Critical Paths,” accepted in IEEE Trans. on Parallel and Distributed Systems.

[2] S. Abrishami, M. Naghibzadeh, “Deadline-constrained Workflow Scheduling in Software as a Service Clouds,” accepted in Scientia Iranica.

[3] S. Abrishami, M. Naghibzadeh, and D.H.J. Epema, “Deadline-constrained Workflow Scheduling Algorithms for Infrastructure as a Service,” submitted to Future Generation Computer Systems.

[4] Abrishami, S., Naghibzadeh, M., Epema, D.H.J. , "Cost-driven scheduling of grid workflows using Partial Critical Paths," 11th IEEE/ACM International Conference on Grid Computing (GRID ’10), Oct. 2010, pp. 81-88.

[5] S. Abrishami, M. Naghibzadeh, “Budget Constrained Scheduling of Grid Workflows Using Partial Critical Paths,” International Conference on Grid Computing and Applications (GCA ’11), pp. 108 –113, July 2011.

چکیده

اخیرا، سیستم های مشبک عمومی و ابرهای محاسباتی خود را به عنوان یک مدل جدید برای ارائه خدمات در سیستم های توزیع شده مطرح کرده اند. در این سیستم ها، منابع و خدمات توزیع شده مختلف (همچون خدمات محاسباتی و ذخیره سازی اطلاعات)، با کارآیی و کیفیت خدمت متفاوت، و با قیمت های مختلف به فروش می رسند. مسئله زمانبندی جریان های کاری، یکی از پیچیده ترین مسائل موجود در سیستم های مشبک عمومی و ابرهای محاسباتی است. در این سیستم ها، زمانبندی باید به گونه ای انجام شود که حداقل کیفیت خدمت موردنظر کاربر برآورده شود. برخلاف سیستم های توزیع شده سنتی که در آنها هدف از زمانبندی تنها بهینه سازی یک معیار می باشد (معمولا زمان اجرا)، در این سیستم ها باید چندین معیار به طور همزمان درنظر گرفته شوند (مانند زمان و هزینه اجرا). به همین دلیل به این نوع زمانبندی، زمانبندی چند معیاری یا زمانبندی مبتنی بر کیفیت خدمت گفته می شود. در این رساله، چندین الگوریتم زمانبندی مبتنی بر کیفیت خدمت برای برنامه های جریان کاری، در محیط سیستم های مشبک عمومی و ابرهای محاسباتی ارائه شده است.

در ابتدا، الگوریتمی به نام Partial Critical Paths (PCP) برای زمانبندی جریان های کاری در سیستم های مشبک عمومی ارائه گردیده است که سعی می کند جریان کاری را با حداقل هزینه، تا پیش از مهلت تعیین شده توسط کاربر به پایان برساند. الگوریتم PCP دارای دو مرحله اصلی است: توزیع مهلت و برنامه ریزی. در مرحله اول، مهلت کلی جریان کار بر روی وظایف آن توزیع می گردد، و در مرحله دوم، هر وظیفه با توجه به زیرمهلت تخصیص یافته به آن، زمانبندی می گردد. سپس عملکرد این الگوریتم بر روی پنج جریان کاری واقعی بررسی شده و نتایج خوبی به دست آمده است. در ادامه، الگوریتم زمانبندی Budget-PCP معرفی شده است که هدف آن حداقل کردن زمان اجرای جریان کاری ورودی است، به شرطی که هزینه اجرا از بودجه تعیین شده توسط کاربر بیشتر نشود. این الگوریتم نیز از دو مرحله اصلی تشکیل می گردد: توزیع بودجه، و برنامه ریزی. نحوه کار هر مرحله تقریبا مشابه الگوریتم PCP است.

در گام بعدی، با در نظر گرفتن تفاوت های موجود بین سیستم های مشبک عمومی و ابرهای نرم افزار به عنوان خدمت، الگوریتم SaaS Cloud-PCP (SC-PCP) را پیشنهاد نموده ایم که عمل زمانبندی را تنها در یک مرحله انجام می دهد. آزمایش های انجام شده نشان می دهد که این الگوریتم از کارآیی بسیار خوبی برخوردار است. در آخرین گام، الگوریتم PCP برای محیط ابرهای زیرساخت به عنوان خدمت تطبیق یافته است. ویژگی مهم این دسته از ابرها در سیاست قیمت گذاری خدمات است که بر اساس بازه های زمانی ثابت (معمولا یک ساعت) می باشد. با ایجاد تغییراتی در الگوریتم اولیه، دو الگوریتم جدید برای این نوع از ابرها پیشنهاد شده اند: الگوریتم یک مرحله ای IaaS Cloud-PCP (IC-PCP)، و الگوریتم دو مرحله ای IaaS Cloud-PCP with Deadline Distribution (IC-PCPD2). در پایان، کارآیی هر دو الگوریتم مورد بررسی و مقایسه قرار گرفته است.

واژه های کلیدی: سیستم های مشبک عمومی، محاسبات مشبک، محاسبات ابری، زمانبندی مبتنی بر کیفیت خدمت، زمانبندی جریان های کاری

فهرست مندرجات

فهرست مندرجات ا

لیست جداول ث

لیست اشکال ث

لیست الگوریتمها ح

۱ مقدمه ۱

۱-۱ نسل جدید سیستم های توزیع شده ۲

۱-۱-۱ سیستم های مشبک ۲

۱-۱-۲ سیستم های مشبک عمومی ۴

۱-۱-۳ ابرهای محاسباتی ۷

۱-۲ برنامه های جریان کاری ۹

۱-۳ انگیزه تحقیق ۱۳

۱-۴ مشارکت علمی ۱۴

۱-۵ ساختار رساله ۱۶

۲ مروری بر پیشینه تحقیق ۱۷

۲-۱ طبقه بندی مسئله زمانبندی جریان های کاری ۱۷

۲-۱-۱ مدل جریان کار ۱۸

۲-۱-۲ معیارهای زمانبندی ۲۰

۲-۱-۳ فرآیند زمانبندی ۲۲

۲-۱-۴ منابع سیستم مشبک ۲۴

۲-۱-۵ وظایف جریان کار ۲۵

۲-۲ مروری بر الگوریتم های زمانبندی حداکثر تلاش ۲۶

۲-۲-۱ زمانبندی پویا ۲۶

۲-۲-۲ زمانبندی ایستا ۲۸

الگوریتم های فهرست ۲۸

الگوریتم های خوشه بندی ۳۰

الگوریتم های مبتنی بر تکثیر وظایف ۳۱
الگوریتم های فوق ابتکاری ۳۲
۲-۳ مروری بر الگوریتم های زمانبندی مبتنی بر کیفیت خدمت ۳۳
۲-۳-۱ الگوریتم های ابتکاری ۳۳
الگوریتم زمانبندی Deadline-MDP ۳۴
الگوریتم Deadline Early Tree (DET) ۳۵
الگوریتم Dynamic Constraint Algorithm (DCA) ۳۵
الگوریتم های Lossو Gain ۳۶
الگوریتم Biobjective Dynamic Level Scheduling (BDLS) ۳۶
۲-۳-۲ الگوریتم های فوق ابتکاری ۳۷
۲-۳-۳ الگوریتم های مبتنی بر مدل های ریاضی ۳۸
۲-۴ مروری بر الگوریتم های زمانبندی در ابرها ۳۹
۲-۵ نگاهی به سیستم های مدیریت جریان کار ۴۱
۲-۶ نتیجه گیری ۴۲

۳ برنامه های جریان کاری علمی ۴۳
۳-۱ ساختارهای پایه ای برنامه های جریان کاری ۴۳
۳-۲ جریان های کاری نمونه ۴۴
۳-۲-۱ جریان کاری Montage ۴۴
۳-۲-۲ جریان کاری CyberShake ۴۵
۳-۲-۳ جریان کاری Epigenomics ۴۷
۳-۲-۴ جریان کاری LIGO ۴۸
۳-۲-۵ جریان کاری SIPHT ۴۹
۳-۳ برنامه های جریان کاری به عنوان محک ۵۱

۴ زمانبندی در سیستم های مشبک عمومی ۵۳
۴-۱ مدل سیستم زمانبندی ۵۳
۴-۲ الگوریتم زمانبندی PCP ۵۵
۴-۲-۱ ایده اصلی الگوریتم ۵۵
۴-۲-۲ تعاریف اولیه ۵۶
۴-۲-۳ الگوریتم زمانبندی جریان کاری ۵۷
۴-۲-۴ الگوریتم تخصیص زیرمهلت به والدین (AssignParents) ۵۹
۴-۲-۵ الگوریتم تخصیص مسیر (AssignPath) ۵۹
سیاست بهینه ۶۰
سیاست کاهش هزینه ۶۱
سیاست منصفانه ۶۳

۴-۲-۶ الگوریتم برنامه ریزی ۶۳
۴-۲-۷ محاسبه مرتبه زمانی ۶۴
۴-۲-۸ تحمل پذیری خطا ۶۶
۴-۲-۹ تشریح الگوریتم در قالب مثال ۶۷
۴-۲-۱۰ ارزیابی کارآیی الگوریتم ۷۰
تنظیمات آزمایش ها ۷۰
نتایج حاصل از آزمایش ها ۷۱
زمان اجرا ۷۵
۴-۳ الگوریتم زمانبندی Budget-PCP ۷۷
۴-۳-۱ ایده اصلی الگوریتم ۷۷
۴-۳-۲ الگوریتم زمانبندی جریان کاری ۷۸
۴-۳-۳ الگوریتم تخصیص زیربودجه به والدین (AssignParents) ۷۹
۴-۳-۴ الگوریتم تنظیم بودجه ۸۰
۴-۳-۵ الگوریتم برنامه ریزی ۸۰
۴-۳-۶ محاسبه مرتبه زمانی ۸۱
۴-۳-۷ ارزیابی کارآیی الگوریتم ۸۱

۵ زمانبندی در ابرهای محاسباتی ۸۴
۵-۱ زمانبندی در ابرهای نرم افزار به عنوان خدمت ۸۴
۵-۲ مدل ابر نرم افزار به عنوان خدمت ۸۵
۵-۳ الگوریتم زمانبندی SC-PCP ۸۶
۵-۳-۱ ایده اصلی الگوریتم ۸۶
۵-۳-۲ تعاریف اولیه ۸۷
۵-۳-۳ الگوریتم زمانبندی پیشنهادی ۸۸
۵-۳-۴ الگوریتم زمانبندی والدین (ScheduleParents) ۸۹
۵-۳-۵ الگوریتم زمانبندی مسیر (SchedulePath) ۹۰
سیاست بهینه ۹۰
سیاست کاهش هزینه ۹۲
سیاست منصفانه ۹۳
۵-۳-۶ محاسبه مرتبه زمانی ۹۴
۵-۳-۷ اثبات درستی الگوریتم ۹۵
۵-۴ ارزیابی کارآیی الگوریتم SC-PCP ۹۷
۵-۴-۱ تنظیمات آزمایش ها ۹۷
۵-۴-۲ نتایج حاصل از آزمایش ها ۹۸
۵-۴-۳ زمان اجرا ۱۰۲
۵-۵ زمانبندی در ابرهای زیرساخت به عنوان خدمت ۱۰۲

۵-۶ مدل ابر زیرساخت به عنوان خدمت ۱۰۴
۵-۷ الگوریتم زمانبندی IC-PCP ۱۰۵
۵-۷-۱ تعاریف اولیه ۱۰۵
۵-۷-۲ الگوریتم زمانبندی پیشنهادی ۱۰۶
۵-۷-۳ الگوریتم تخصیص والدین (AssignParents) ۱۰۶
۵-۷-۴ الگوریتم تخصیص مسیر (AssignPath) ۱۰۷
۵-۷-۵ محاسبه مرتبه زمانی ۱۰۸
۵-۸ الگوریتم زمانبندی IC-PCPD2 . . . . . ۱۰۹
۵-۸-۱ الگوریتم تخصیص والدین (AssignParents) ۱۰۹
۵-۸-۲ الگوریتم تخصیص مسیر (AssignPath) ۱۱۰
۵-۸-۳ الگوریتم برنامه ریزی ۱۱۰
۵-۸-۴ محاسبه مرتبه زمانی ۱۱۱
۵-۹ ارزیابی کارآیی الگوریتم های IC-PCPو IC-PCPD2 . . . . . ۱۱۱
۵-۹-۱ الگوریتم زمانبندی مورد مقایسه ۱۱۱
۵-۹-۲ تنظیمات آزمایش ها ۱۱۲
۵-۹-۳ نتایج حاصل از آزمایش ها ۱۱۴

۶ نتیجه گیری و پیشنهادات ۱۱۷
۶-۱ نتیجه گیری ۱۱۷
۶-۲ پیشنهادات ۱۲۰

واژه‌نامه انگلیسی به فارسی ۱۲۲

مراجع ۱۲۶

لیست جداول

۳-۱ نمونه ای از اجرای یک برنامه Montage [Bha08] ۴۶
۳-۲ نمونه ای از اجرای یک برنامه CyberShake [Bha08] ۴۶
۳-۳ نمونه ای از اجرای یک برنامه Epigenomics [Bha08] ۴۸
۳-۴ نمونه ای از اجرای یک برنامه LIGO [Bha08] ۴۹
۳-۵ نمونه ای از اجرای یک برنامه SIPHT [Bha08] ۵۱

۴-۱ خدمات موجود برای جریان کاری شکل ۴-۱ . . . . . ۶۷
۴-۲ مقادیر پارامترها برای هر گام از اجرای الگوریتم PCPبر روی جریان کاری شکل ۴-۱ . . . . . ۶۸
۴-۳ متوسط درصد تفاوت بین ضریب مهلت و زمان اجرای نرمال جریان های کاری . . . ۷۲
۴-۴ متوسط کاهش هزینه الگوریتم PCPبا سیاست بهینه در مقایسه با الگوریتم Deadline-MDP ۷۴
۴-۵ حداکثر زمان اجرای الگوریتم ها برای جریان های کاری بزرگ (میلی ثانیه) ۷۵

۵-۱ متوسط کاهش هزینه الگوریتم SC-PCPبا سیاست بهینه در مقایسه با الگوریتم Deadline-MDP ۹۸
۵-۲ حداکثر زمان اجرای سیاست های مختلف زمانبندی مسیر برای جریان های کاری بزرگ (برحسب میلی ثانیه) ۱۰۱
۵-۳ انواع خدمات ارائه شده به کاربران توسط Amazon EC2(نوامبر ۲۰۱۱) ۱۰۳

لیست اشکال

۱-۱ ساختار سیستم مدیریت منابع در یک سیستم مشبک [Don06] ۳
۱-۲ ساختار سیستم مدیریت منابع در یک سیستم مشبک عمومی [Buy02b] ۶
۱-۳ نمونه ای از یک برنامه جریان کاری ۱۰
۱-۴ برنامه جریان کاری Epigenomicsبر گرفته از مرجع [Peg] ۱۱
۱-۵ ساختار یک سیستم مدیریت جریان کار [Yu08] ۱۲

۲-۱ طبقه بندی مدل جریان کار [Wie09] ۱۸
۲-۲ طبقه بندی معیارهای زمانبندی جریان کار [Wie09] ۲۰
۲-۳ طبقه بندی فرآیند زمانبندی جریان کار [Wie09] ۲۳
۲-۴ طبقه بندی منابع سیستم مشبک [Wie09] ۲۴
۲-۵ طبقه بندی وظایف جریان کار [Wie09] ۲۵
۲-۶ نحوه بخش بندی جریان های کاری [Yu05b] ۳۴

۳-۱ ساختارهای پایه ای تشکیل دهنده برنامه های جریان کاری [Bha08] ۴۴
۳-۲ برنامه جریان کاری Montage[Bha08] ۴۵
۳-۳ برنامه جریان کاری CyberShake[Bha08] ۴۶
۳-۴ برنامه جریان کاری Epigenomics[Bha08] ۴۷
۳-۵ برنامه جریان کاری LIGO[Bha08] ۴۹
۳-۶ برنامه جریان کاری SIPHT[Bha08] ۵۰

۴-۱ نمونه ای از یک برنامه جریان کاری ۶۷
۴-۲ نتایج حاصل از اجرای برنامه های جریان کاری با سه الگوریتم HEFT، سریع ترین و ارزان ترین ۷۱
۴-۳ هزینه نرمال زمانبندی جریانهای کاری با استفاده از الگوریتمهای PCPو Deadline-MDP ۷۳
۴-۴ زمان اجرای نرمال زمانبندی جریان های کاری با استفاده از الگوریتم های Budget-PCPو Loss ۸۲

۵-۱ زمان اجرای نرمال زمانبندی جریان های کاری توسط الگوریتم های SC-PCPو Deadline-MDP ۹۹

۵-۲ هزینه نرمال زمانبندی جریان های کاری توسط الگوریتم های SC-PCPو -Deadline MDP ۱۰۰
۵-۳ هزینه نرمال زمانبندی جریان های کاری با بازه زمانی برابر یک ساعت ۱۱۴
۵-۴ هزینه نرمال زمانبندی جریان های کاری با بازه زمانی برابر پنج دقیقه ۱۱۵

لیست الگوریتم ها

۱-۴ الگوریتم زمانبندی PCP ۵۸
۲-۴ الگوریتم تخصیص زیرمهلت به والدین یک گره (در الگوریتم PCP) ۵۸
۳-۴ الگوریتم تخصیص مسیر بهینه (در الگوریتم PCP) ۶۰
۴-۴ الگوریتم تخصیص مسیر کاهش هزینه (در الگوریتم PCP) ۶۲
۵-۴ الگوریتم تخصیص مسیر منصفانه (در الگوریتم PCP) ۶۳
۶-۴ الگوریتم برنامه ریزی (در الگوریتم PCP) ۶۴
۷-۴ الگوریتم زمانبندی Budget-PCP ۷۸
۱-۵ الگوریتم زمانبندی SC-PCP ۸۸
۲-۵ الگوریتم زمانبندی والدین (در الگوریتم SC-PCP) ۸۹
۳-۵ الگوریتم زمانبندی مسیر بهینه (در الگوریتم SC-PCP) ۹۲
۴-۵ الگوریتم زمانبندی مسیر کاهش هزینه (در الگوریتم SC-PCP) ۹۳
۵-۵ الگوریتم زمانبندی مسیر منصفانه (در الگوریتم SC-PCP) ۹۳
۶-۵ الگوریتم زمانبندی IC-PCP ۱۰۶
۷-۵ الگوریتم تخصیص والدین (در الگوریتم IC-PCP) ۱۰۷
۸-۵ الگوریتم تخصیص مسیر (در الگوریتم IC-PCP) ۱۰۷
۹-۵ الگوریتم برنامه ریزی (در الگوریتم IC-PCP) ۱۱۰

فصل۱
مقدمه

سیستم های توزیع شده، همانند خوشه ها[۱]، مدت زیادی است که برای منظورهای مختلفی همچون محاسبات با کارآیی بالا[۲]، دسترس پذیری بالا[۳] و برون دهی بالا[۴] استفاده می گردند. اما ظهور اینترنت و شبکه های پر سرعت باعث پیدایش سیستم های توزیع شده جدیدی همانند سیستم های مشبک[۵] و ابرهای محاسباتی[۶] شده است که به کاربران اجازه می دهند به منابعی که از نظر جغرافیایی در نقاط مختلفی قرار گرفته اند، دسترسی پیدا کنند. این سیستم های جدید دانشمندان و محققان را قادر می سازند که برنامه های علمی با پردازش های سنگین[۷]، همچون برنامه های جریان کاری[۸] خود را بر روی مجموعه ای از سیستم های توزیع شده در نقاط جغرافیایی مختلف اجرا نمایند [Hey02]. اما برای استفاده بهینه از توان بالقوه این سیستم های توزیع شده عظیم نیاز به الگوریتم های کارآ و موثر زمانبندی داریم که بتوانند بهترین منابع را برای اجرای برنامه ها انتخاب نمایند. به دلیل تفاوت های ساختاری سیستم های توزیع شده جدید با نمونه های سنتی، همچون خوشه ها و سیستم های چند پردازنده ای متقارن[۹]، الگوریتم های زمانبندی موجود برای سیستم های سنتی کارآیی خوبی در محیطهای جدید ندارند (و یا اصلا قابل استفاده نیستند). علاوه براین با ظهور سیستم های مشبک عمومی[۱۰] و ابرهای محاسباتی عامل دیگری نیز به مسئله زمانبندی اضافه شده است: قیمت منابع. در این رساله قصد داریم الگوریتمی ابتکاری[۱۱] برای زمانبندی برنامه های جریان کاری در محیط سیستم های مشبک عمومی و ابرهای محاسباتی طراحی نماییم که بتواند برنامه را پیش از مهلت تعیین شده توسط کاربر و با حداقل قیمت اجرا نماید. در ادامه این فصل ابتدا به بررسی سیستم های توزیع شده جدید همانند سیستم های مشبک و ابرهای محاسباتی می پردازیم، سپس برنامه های جریان کاری را معرفی کرده و در ادامه به بررسی مسئله زمانبندی در محیط سیستم های توزیع شده می پردازیم. در پایان انگیزه خود از بررسی مسئله زمانبندی مبتنی بر کیفیت خدمت برنامه های جریان کاری را بیان خواهیم کرد.

---

۱ Clusters
۲ High Performance Computing
۳ High Availability
۴ High Throughput
۵ Grids
۶ Clouds
۷ e-Science Applications
۸ Workflow Applications
۹ Symmetric Multiple Processor
۱۰ Utility Grids
۱۱ Heuristic

۱-۱ نسل جدید سیستم های توزیع شده

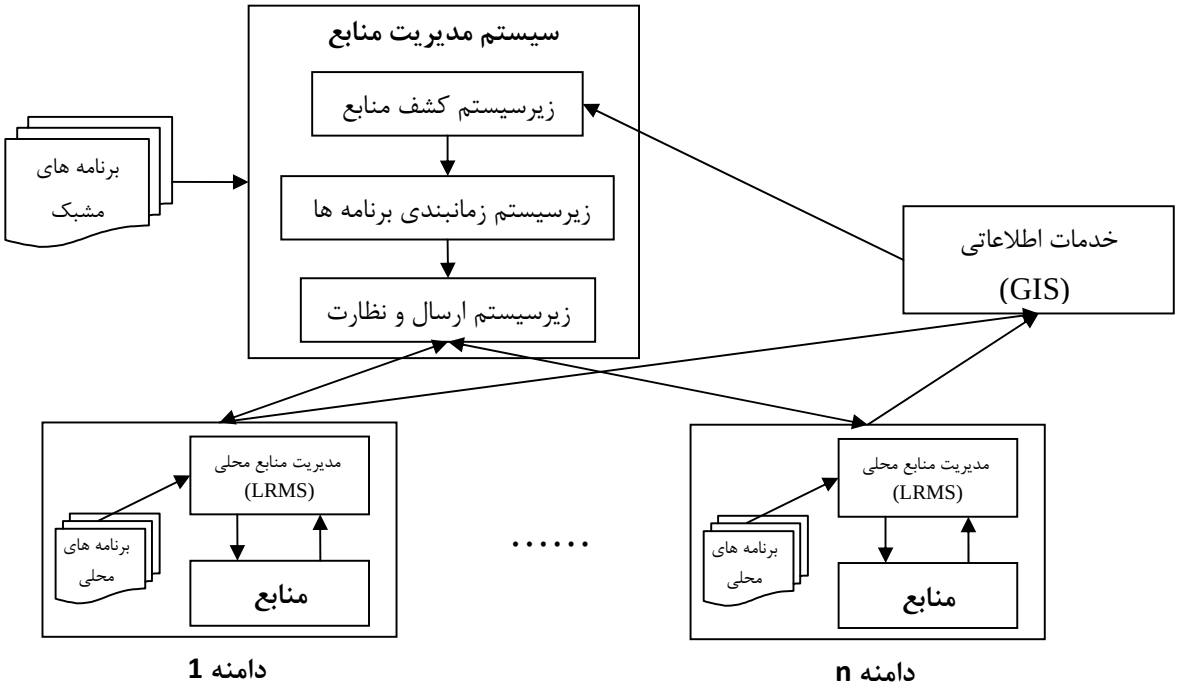
همان گونه که گفته شد، با ظهور شبکه های سرعت بالا در سطح کشورها و خطوط پر سرعت اینترنت، امکان استفاده کاربران از منابعی که از لحاظ جغرافیایی در مکان های دیگری (و حتی در کشور دیگری) قرار گرفته اند فراهم گردیده است. در ادامه به بررسی برخی از این سیستم های توزیع شده نوظهور می پردازیم.

۱-۱-۱ سیستم های مشبک

سیستم های مشبک، توسط Ian Foster که از وی به عنوان پدر این سیستم ها یاد می گردد، این گونه توصیف می گردد: "به اشتراک گذاری منابع و حل مسئله هماهنگ، در سازمان های مجازی[۱]، پویا و چند موسسه ای" [Fos01]. در حقیقت سیستم های مشبک امکان به اشتراک گذاری مجموعه ای از منابع ناهمگون مانند ابررایانه ها، سیستم های ذخیره سازی، منابع داده و دستگاه های خاص را که از نظر جغرافیایی در مکان های مختلفی قرار گرفته و متعلق به موسسات متفاوتی هستند را فراهم می سازد. منظور از یک سازمان مجازی، مجموعه ای از موسسات است که برای رسیدن به یک هدف مشترک منابع خود را به اشتراک گذاشته اند. هدف از ایجاد چنین سیستمی حل مسائل علمی و مهندسی است که نیاز به محاسبات سنگین و یا حجم عظیمی از داده ها (ی توزیع شده) دارند. اصطلاح محاسبات مشبک[۲] به استفاده از تعداد زیادی از منابع ناهمگون و توزیع شده از نظر جغرافیایی، برای حل یک مسئله بسیار سنگین اطلاق می گردد. از دیدگاه کاربر، سیستم مشبک یک سیستم یکپارچه برای حل مسائل سنگین است، بدون آنکه بداند منابع از لحاظ فیزیکی در کجا قرار گرفته و یا متعلق به چه موسسه ای هستند.

یکی از مهمترین اجزای سیستم های مشبک، بخش مدیریت منابع است. مدیریت منابع وظیفه دریافت برنامه کاربر و تعیین منابع مورد نیاز آن، کشف منابع در دسترس و مناسب از طریق خدمات اطلاعاتی سیستم مشبک یا GIS[۳]، تخصیص منابع و زمانبندی برنامه بر روی آنها، ارسال برنامه به منابع و در نهایت نظارت بر اجرای آن بر روی منابع را به عهده دارد. شکل ۱-۱ ساختار سیستم مدیریت منابع در یک سیستم مشبک را نشان می دهد [Don06]. همان طور که در شکل دیده می شود، یک سیستم مشبک از تعدادی عرضه کننده خدمات تشکیل شده است که هریک توسط یک دامنه مدیریتی مستقل نشان داده شده اند. در هر دامنه یک سیستم مدیریت منابع محلی[۴] وجود دارد که وظیفه زمانبندی برنامه هایی که از طرف کاربران داخلی و یا کاربران سیستم مشبک به آن ارسال می گردد را بر روی منابع محلی خود بر عهده دارد. هر دامنه موظف است اطلاعات منابع موجود خود را بر روی سیستم خدمات اطلاعاتی سیستم مشبک ثبت نماید. این اطلاعات بر دو دسته اند: اطلاعات ایستا (همانند نوع و ظرفیت منبع) و اطلاعات پویا (مانند میزان بار فعلی). هنگامی که یک برنامه به سیستم مدیریت منابع سیستم مشبک ارسال می گردد، ابتدا زیرسیستم کشف منابع اطلاعات منابع موجود را از سیستم خدمات اطلاعاتی دریافت و منابع مناسب برای برنامه را انتخاب می کند. سپس زیرسیستم زمانبندی براساس معیارهای موردنظر سیستم مشبک (مانند بالا رفتن میزان بهره وری) و/یا معیارهای موردنظر کاربر (مانند حداقل شدن زمان اجرا) منابع

۱ Virtual Organization
۲ Grid Computing
۳ Grid Information Service
۴ Local Resource Management System



شکل ۱-۱ : ساختار سیستم مدیریت منابع در یک سیستم مشبک [Don06]

مناسب را انتخاب کرده و برنامه را بر روی آنها زمانبندی می نماید. در پایان، زیرسیستم ارسال و نظارت با سیستم مدیریت منابع محلی دامنه (دامنه های) موردنظر ارتباط برقرار کرده و برنامه را برای اجرا به آن(ها) ارسال می کند. علاوه براین، این زیرسیستم بر اجرای صحیح برنامه نیز نظارت کرده و درصورت بروز خطا، راهکارهای لازم را اتخاذ می نماید.

در میان سیستم های فوق، زمانبند سیستم مشبک نقش بسیار مهمی در بالا رفتن میزان بهره وری سیستم و همچنین برآورده شدن نیازهای کاربر (همچون اتمام هرچه سریع تر برنامه) دارد. گاهی برای آن که زمانبند سیستم مشبک با زمانبند محلی دامنه اشتباه نشود، آن را فوق زمانبند[۱] یا دلال[۲] می نامند. دلیل انتخاب نام دلال آن است که این زمانبند به صورت مستقیم به منابع اصلی سیستم مشبک دسترسی ندارد و تنها به عنوان واسطه ای بین کاربر و زمانبند محلی عمل می نماید. گرچه سیستم های مشبک نیز جزو سیستم های موازی و توزیع شده محسوب می گردند، اما به دلیل تفاوت های ساختاری که با سیستم های سنتی (همچون خوشه ها و سیستم های چندپردازنده ای متقارن) دارند، الگوریتم های زمانبندی طراحی شده برای آنها در این محیط جدید کارآیی خوبی ندارد. مهم ترین این تفاوت ها عبارتند از:

- **ناهمگونی منابع**[۳]: برخلاف سیستم های توزیع شده سنتی که معمولا از منابع همگون استفاده می کنند، منابع در سیستم های مشبک کاملا ناهمگون هستند؛ از کامپیوترهای شخصی گرفته تا خوشه ها و ابرکامپیوترهای موازی. علاوه براین، به دلیل قرار گرفتن منابع در دامنه های مدیریتی متفاوت در مکان های جغرافیایی مختلف، پهنای باند و سرعت شبکه نیز ناهمگون می باشد.

۱ Meta-scheduler
۲ Broker
۳ Resource Heterogeneity

- **استقلال منابع**[۱]: در سیستم های مشبک، هر دامنه کاملا مستقل بوده و از طریق سیستم مدیریت منابع محلی، منابع خود را کنترل می نماید. بنابراین زمانبند سطح بالای سیستم مشبک هیچ گونه کنترل مستقیمی بر روی منابع و سیاست های دسترسی به آنها (مانند تعیین اولویت کاربران و برنامه ها) ندارد و تنها می تواند از طریق زمانبندهای محلی به منابع دسترسی پیدا کند.
- **پویایی:** سیستم های مشبک شدیدا پویا هستند: وضعیت منابع موجود (مانند میزان بار آنها) مرتبا تغییر می کند، منابع جدیدی به سیستم اضافه می شوند و برخی منابع نیز به دلیل بروز خطا از دسترس خارج می گردند. از آنجا که منابع علاوه بر برنامه های سیستم مشبک، برنامه های محلی مربوط به دامنه خود را نیز اجرا می نمایند، کارآیی آنها وابسته به تعداد کارهای ارجاعی تغییر می نماید. علاوه براین، به دلیل گستردگی این سیستم ها، امکان خرابی منابع بسیار بالا است و سیستم های زمانبندی باید راهکارهای لازم برای جابجایی و یا زمانبندی مجدد برنامه ها را درنظر بگیرند.

وجود این تفاوت ها، مسئله زمانبندی در سیستم های مشبک را تبدیل به مسئله ای بسیار پیچیده و چالش برانگیز کرده است که هنوز فاصله زیادی تا یافتن یک راه حل کامل و همه جانبه دارد. Dong و همکاران [Don06] الگوریتم های متداول در زمانبندی سیستم های مشبک را مورد بررسی قرار داده اند.

۱-۱-۲ سیستم های مشبک عمومی

در سال های اخیر، سیستم های مشبک به سمت ایجاد یک بازار آزاد از خدمات توزیع شده حرکت کرده اند که در آن منابع و خدمات مختلف، با کارآیی و کیفیت خدمت[۲] متفاوت، و با قیمت های مختلف به فروش می رسند [Laf06]. ایده اصلی سیستم های مشبک عمومی، ارائه خدمات فناوری اطلاعات (همچون خدمات محاسباتی و ذخیره سازی اطلاعات) به کاربران نهایی مشابه با سیستم های خدمات عمومی سنتی همچون آب، برق و تلفن است. در این مدل، کاربر بر اساس نیازها و کیفیت خدمت مورد نیاز خود با یک عرضه کننده خدمات[۳] وارد مذاکره شده و به توافق می رسد. سپس کاربر می تواند به خدمات دسترسی پیدا کند بدون آنکه بداند میزبان این خدمات در کجا واقع شده و یا این خدمات چگونه برای وی مهیا می شوند، و در پایان کاربر بر اساس میزان استفاده از خدمات و یا به صورت یک حق اشتراک ثابت دوره ای، بهای آن را می پردازد.

تفاوت اصلی بین سیستم های مشبک گروهی[۴] سنتی و سیستم های مشبک عمومی در مفهوم کیفیت خدمت است: درحالی که سیستم های مشبک گروهی خدمات را براساس حداکثر تلاش[۵] خود ارائه می دهند و هیچ تضمینی بر روی کیفیت خدمات ندارند، سیستم های مشبک عمومی کیفیت خدمات را براساس توافقنامه سطح خدمت یا SLA[۶] تضمین می نمایند [Yu06a]. توافقنامه سطح خدمت، یک قرارداد بین عرضه کننده و مصرف کننده خدمات است که کیفیت خدمت و تعهدات دو طرف را مشخص می کند.

---

۱ Resource Autonomy
۲ Quality of Service (QoS)
۳ Service Provider
۴ Community Grids
۵ Best Effort
۶ Service Level Agreement

مصرف کنندگان می توانند از طریق مذاکره با عرضه کنندگان بر سر کیفیت خدمت مورد نیاز خود و قیمت، به یک توافقنامه سطح خدمت دست یابند. قیمت نقش کلیدی در این قرارداد دارد: عرضه کنندگان را تشویق می کند تا خدماتشان را به بازار ارائه نمایند، و از طرف دیگر کاربران را تشویق می کند که کیفیت خدمات مورد نیاز خود را به صورت واقعی (و نه بیشتر از میزان مورد نیاز) تعریف نمایند. پارامترهای مختلفی برای کیفیت خدمت مورد نیاز کاربر وجود دارد که مهمترین آنها عبارتند از: زمان اجرای کار، دسترس پذیری[۱]، قابلیت اطمینان[۲]، امنیت[۳] و خوشنامی[۴] عرضه کننده خدمات.

مسلما سیستم های مدیریت منابع سنتی برای مدیریت سیستم های مشبک عمومی مناسب نیستند. پیش از هرچیز، سیستم مدیریت منابع نیاز به چارچوبی برای مذاکره و عقد قرارداد بین مصرف کنندگان و عرضه کنندگان خدمات دارد. علاوه براین نیاز به مدل های اقتصادی داریم که براساس آنها عرضه کنندگان خدمات قیمت های منابع خود را تعیین کرده و مصرف کنندگان نیز راهبردهای استفاده و خرید خود را تعیین نمایند. شکل ۱-۲ ساختار یک سیستم مدیریت منابع برای سیستم های مشبک عمومی را نشان می دهد که با مقداری تغییر از مرجع [Buy02b] گرفته شده است. همان طور که در شکل دیده می شود، علاوه بر زیرسیستم خدمات اطلاعاتی، زیرسیستم دیگری به نام راهنمای بازار سیستم مشبک یا GMD[۵] وجود دارد که کلیه عرضه کنندگان می توانند خدمات خود را به همراه نحوه قیمت گذاری آنها در آن منتشر نمایند. علاوه براین، سیستم دلال منابع که از طرف کاربر وظیفه زمانبندی و اجرای برنامه را به عهده دارد، مجهز به یک زیرسیستم مدیر تجاری شده است که قادر است با زیرسیستم کارگزار تجاری یک عرضه کننده خدمات وارد مذاکره بر روی کیفیت خدمت و قیمت شده و به یک توافقنامه سطح خدمت دست یابد. البته برای مذاکره نیاز به پروتکل ها و استانداردهایی است که بستگی به مدل اقتصادی مورد استفاده عرضه کننده خدمات دارد. زیرسیستم کارگزار تجاری مربوط به ارائه کنندگان خدمات خود دارای دو زیرسیستم دیگر است: الگوریتم های قیمت گذاری که با توجه به مدل اقتصادی مورد استفاده، میزان تقاضای بازار و منابع موجود، قیمت منابع را تعیین می نمایند؛ و زیرسیستم حسابرسی که میزان استفاده هر کاربر از منابع را ثبت کرده و درپایان صورتحساب وی را صادر می نماید. علاوه براین، رزرو پیشاپیش[۶] منابع نیز یکی از روش های متداول به منظور تضمین سطح خدمت موردنیاز کاربر است، به همین منظور زیرسیستم رزرو منابع به سیستم مدیریت منابع ارائه دهندگان خدمات اضافه شده است.

مدل های اقتصادی مختلفی توسط Buyya و همکاران [Buy02a] برای سیستم های مشبک عمومی پیشنهاد شده است که عبارتند از:

- مدل بازار کالا[۷]: در این مدل عرضه کنندگان خدمات، قیمت منابع خود را مشخص کرده و برمبنای آن صورتحساب کاربران را صادر می نمایند. قیمت ها ممکن است ثابت، و یا متغیر برحسب میزان عرضه و تقاضا باشد. علاوه براین ممکن است صورتحساب کاربران براساس میزان استفاده آنها از منابع صادر شود، و یا اینکه هر کاربر یک حق عضویت ثابت را برای یک بازه مشخص (مثلا ماهانه) پرداخت نماید (ترکیبی از هردو نیز ممکن است).

---

۱ Availability
۲ Reliability
۳ Security
۴ Reputation
۵ Grid Market Directory
۶ Advance Reservation
۷ Commodity Market Model