ارزیابی عملکرد دسته ای از گروه ها در یک سیستم توزیع شده ناهمگن

مهین وظیفه دان
گروه کامپیوتر،دانشگاه آزاد اسلامی واحد مشهد،ایران
mahinvazifehdan@yahoo.com

۲۳ اسفند ۱۳۹۲

چکیده

سیستم های توزیع شده یک راه حل مقرون به صرفه و مقیاس پذیر به منظور افزایش برنامه های فشرده با استفاده از چندین منبع به اشتراگ گذاشته شده را ارائه میدهد.زمان بندی گروهی یک الگوریتم زمان بندی موثر برای سیستم های توزیع شده و موازی میباشد.در این مقاله ما عملکرد سیاست های زمان بندی دسته ای از گروه های مستقل در یک سیستم توزیع شده ناهمگن را مورد بررسی قرار میدهیم.یک مدل شبیه سازی به منظور ارزیابی عملکرد زمان بندی دسته ای از گروه ها در حضور کارهای با اولویت بالا همراه با پیاده سازی مهاجرت در نظر گرفته شده است.نتایج شبیه سازی نقش قابل توجه طرح مهاجرت پیاده سازی شده را به عنوان فاکتور تعادل بار در یک سیستم توزیع شده ناهمگن نشان میدهد.یکی دیگر از جنبه های قابل توجه پیاده سازی مهاجرت کاهش تکه تکه شدن ناشی از زمان بندی گروه ها و همچنین جلوگیری از اثر ورود کارهای با اولویت بالا در محیط ناهمگن میباشد.

کلمات کلیدی: زمان بندی گروهی،استراتژی های زمان بندی،سیستم های توزیع شده،کارایی،شبیه سازی

۱ مقدمه

سیستم های توزیع شده به دلیل ویژگی های قابل توجه شان مانند مقرون به صرفه بودن،قابلیت مقیاس پذیری،عملکرد و قابلیت اطمینان پذیریشان بسیار محبوب شده اند.یک سیستم توزیع شده از تعدادی منابع به اشتراک گذاری شده تشکیل شده است که به وسیله یک شبکه ارتباطی به یک دیگر متصل شده اند.منابع محلی میتوانند در قالب یک کلاستر گروه بندی شوند و بخشی از یک شبکه گسترده تر(گرید) را ایجاد کنند.به دلیل وجود کلاسترها در اندازه و عملکرد های متفاوت،ما شاهد سیستم های ناهمگن متفاوتی هستیم.

الگویتم زمان بندی برای عملکرد سیستم

های توزیع شده و موازی بسیار قابل اهمیت هستند.هدف اصلی یک الگوریتم زمان بندی این است که وظایف موجود در سیستم را به درستی درمیان پردازنده های در دسترس طوری توزیع نماید که عملکرد سیستم به حداکثر برسد و بالاترین سطح کیفیت سرویس نشان داده شود.در این مقاله کارهای با اولویت بالایی در زمان بندی وجود دارند که به محض ورود به سیستم نیاز شدید به سرویس دهی سریع به منظور برآورده شدن انتظاراتشان دارند.

زمان بندی گروهی یک الگوریتم زمان بندی اشتراک گذاری کارامد برای اجرای موازی وظایف در یک سیستم توزیع شده ناهمگن میباشد.در حقیقت مفهوم اصلی زمان بندی گروهی این است که تمام وظایف مربوط به یک کار را گروه بندی کند تا گروه حاصل شود و سپس بعد از آن شروع به اجرای همزمان وظایف بر روی پردازنده های مختلف کنیم.در این روش،ما خطر انتظار یک وظیفه که هنوز در حال اجرا بر روی هیچ پردازنده ای نیست را از بین میبریم.نکته قابل توجه این است که نباید تعداد وظایف یک گروه از تعداد پردازنده های در دسترس بیشتر باشند،به عبارتی دیگر باید نگاشتی یک به یک بین وظایف یک گروه و پردازنده های در دسترس وجود داشته باشد.در این مقاله،ما کارهایی خواهیم داشت که دسته ای از گروه های مستقل هستند نه گروه های تنها.یک کار یا دسته ای از گروه ها زمانی اجرایش به پایان رسیده است که تمام گروه های متعلق به آن دسته اجرایشان به اتمام رسیده باشد.گروه های متعلق به یک دسته میتوانند بر روی هر پردازنده ای با هر ترتیبی اجرا شوند.در زمان بندی گروهی بیشتر اوقات ما شاهد بیکاری اکثر پردازنده های سیستم هستم به همین دلیل چالشی به نام تکه تکه شدن در اجرای وظایف به وجود میاید که باعث افت شدید کارایی میشود،به بیان ساده تر زمان هایی وجود دارد که اکثر پردانده ها در حال اجرای وظایف هستند ولی برعکس،در واحدی از زمان ما شاهد بیکاری اغلب پردازنده ها هستیم به دلیل آنکه وظیفه مناسبی در اختیار پردازنده قرار نگرفته است.راه حل استفاده شده در این مقاله،مهاجرت پویای وظایفی است که قابلیت اجرا بر روی پردازنده های خود را ندارند،این کار باعث ایجاد سیستمی با انعطاف پذیری بالا و کارایی مناسب خواهد شد.در نتیجه میتوان گفت سیستم های چند کلاستری برای رسیدن به تعادل بار نیاز به عمل مهاجرت دارند.

ساختار مقاله بدین صورت میباشد که در بخش ۲ کارهای مربوطه،در بخش ۳ توصیفی از سیستمی که قرار است ارزیابی بر روی آن صورت گیرد را خواهیم داشت در بخش ۴ استراتژی های زمان بندی،در بخش ۵ نتایج شبیه سازی و ارزیابی و در بخش ۶ جمع بندی و کارهای آینده را توضیح خواهیم داد.

۲ کارهای مربوطه

در [۱]ناهمگنی در سیستم های مختلف از دیدگاه های مختلفی نشان داده شده است،دو نوع ناهمگنی تعریف شده است، ناهمگنی زمانی که به تنوع در محاسبه قدرت یا پهنای باند ارتباطات در دسترس برای یک وظیفه در طول بُعد زمان اشاره دارد و ناهمگنی فاصله ای که به قابلیت تغییر پذیری قدرت محاسبات میان کامپیوترها توجه میکند.در حقیقت اثر ناهمگنی زمانی و فضایی بر روی یک وظیفه مورد نظر(وظیفه هدف) مورد تحلیل قرار گرفت.بر اساس نتایج تجزیه و تحلیل یک رویکرد برای به تعادل بار برا ی به حداقل رساندن متوسط زمان اجرا به طور موازی از یک وظیفه مورد نظر توصیف شده است. در

[۲] زمان بندی کارهای موازی در یک محیط ناهمگن مورد مطالعه قرار گرفته است که در هر موقعیت،مجموعه ای از پردازنده های یکسان در سیستم وجود دارند.پردازنده ها بسته به اینکه در چه شرایطی قرار دارند سرعت های مختلفی را از خود نشان میدهند.

یکی از مهم ترین کارهای مربوطه که در زمینه زمان بندی گروهی ارائه شده است،() میباشد که به نظر میرسد مفهوم زمان بندی گروهی برای اولین بار در این مقاله مطرح شده است.بنا بر گفته های در زمان بندی گروهی کارها شامل تعدادی از وظایف هستند که قرار است بر روی پردازنده های مجرا به طور همزمان اجرا شوند.از دو نوع سیاست زمان بندی گروهی برای اجرای وظایف استفاده شده است و نتایج شبیه سازی حاکی از آن است که انتخاب سیاست بزرگ ترین گروه ها بهتر از انتخاب سیاست اجرای به ترتیب گروه ها میباشد و منجر به عملکرد بهتر سیستم میشود.

عملکرد زمان بندی گروهی بر روی یک سیستم دو کلاستری با وجود عمل مهاجرت در(مرجع) مورد بررسی قرار گرفته است.کار دیگری که مورد مطالعه قرار گرفته شده است اثر شدید مهاجرت بر روی کارایی سیستم با حضور کارهای با اولویت بالا میباشد.در ما عملکرد اثر قابلیت تنوع پذیری را در درجه موازی بودن گروه ها مورد مطالعه قرار داده ایم.لازم به ذکر است بسیاری از مطالب گفته شده در کارهای قبلی،در این مقاله بیان میشوند.اما در کارهای قبلی اگر چه که از سیستم کلاستری برای ارزیابی هایمان استفاده کرده ایم اما گروه های تنها مورد پردازش قرار میگرفته اند نه دسته ای گروه ها،به علاوه ما از سیاست های زمان بندی ای استفاده میکنیم که در موارد قبلی بررسی نشده اند.سهم اصلی این مقاله،مطالعه بر روی عملکرد زمان بندی گروهی دسته ای از گروه ها به حساب میایند و بر روی سیستمی با دو کلاستر که از نظر قدرت محاسباتی متفاوت هستند مورد پردازش قرار می گیرند.ما در این مطالعه به بررسی ویژگی هایی میپردازیم که در مطالعات قبلی مورد بررسی قرار نگرفته اند.

با توجه به عدم شباهت کلاسترها از نظر توان محاسباتی بهتر است در ارزیابی های خود از سیستمی استفاده کنیم که کلاسترهای مشابهی نداشته باشند.ما در تحقیقات خود،سیستمی را به کار برده ایم که از این قاعده تبعیت میکند.سیستمی که تعداد پردازنده ها در کلاسترها متفاوت میباشند هرچند پردازنده های یکسانی به کار برده شده است.در حقیقت این رویکرد میتواند نماینده سیستم های مدرن به حساب آید به دلیل اینکه پردازنده ها در تعدا هستشان متفاوت هستند نه در سرعت محاسباتیشان.

۳ سیستم و مدل های حجم کاری

در این مطالعه از یک مدل شبیه سازی برای بررسی عملکرد زمان بندی گروهی دسته ای از گروه ها استفاده شده است.سیستمی با دو کلاستر ناهمگن که کلاستر اولی دارای ۱۶ پردازنده و کلاستر دومی دارای ۳۲ پردازنده میباشد،ساختار سیستم مورد نظر را میتوانید در شکل مشاهده؟؟فرمایید.نتیجه میگیریم که کلاستر دومی از توان محاسباتی بالاتری برخوردار است،هر پردازنده ای مسئولیت سرویس دهی صف خود را برعهده دارد.در این مقاله ما فرض میکنیم که ارتباط بین پردازنده ها بدون مداخله میباشد.زمان سرویس به طور نمایی برابر با $1/\mu$ است.حجم کاری که در سیستم شاهد آن خواهیم بود شامل دو نوع کار است:دسته ای از گروه ها و کارهایی با اولویت بالا.

دسته ای از گروه های مستقل [1] در شکل۳ نشان داده شده است،توجه نمایید که یک گروه قسمتی از یک دسته به حساب میاید.تعداد گروه ها در هر دسته در بازه ۱ـ۴ میباشد.پس در نتیجه میتوان گفت تعداد متوسط گروه ها در هر دسته برابر با ۲.۵=۱+۴ است.همچنین هر گروه دارای وظایفی است که تعداد وظایف در بازه ۸ـ ۱ میباشند که میتوانید ساختار گروه را در شکل۲ مشاهده فرمایید.تعداد متوسط وظایف در هر گروه برابر با ۴.۵=۱+۸ میباشد.

شکل ۱: مدل صف بندی شبکه

شکل ۲: ساختار یک گروه مستقل

نرخ ورود کارهای دسته ای از گروه ها،λ_1 و نرخ ورود کارهای با اولویت بالا [2] برابر با λ_2 است که در این مطالعه توجه ما بیشتر مقدار نرخ ورود دسته ای از گروه ها میباشد.لازم به ذکر است که نرخ ورود دسته ای از گروه ها در سیستم بیشتر از نرخ ورود کارهای با اولویت بالا میباشد.در این مطالعه،ما درنظر گرفته این که هر کار با اولویت بالا تنها دارای یک وظیفه باشد،همان طور که قبلا اشاره شد،کارهای با اولویت بالا به محض ورود به سیستم نیاز دارند که سریع اجرا شوند بنابراین ما مجبور خواهیم شد وظیفه ای که در حال اجرا بر روی یک پردازنده است را به حالت مسدود برده و کار با اولویت بالا را به حالت اجرا هدایت نماییم.وظیفه ای که به حالت مسدود میرود قطعا متعلق به یک گروه خاصی است.مسدود شدن یک وظیفه گروه منجر به مسدود شدن تمام وظایف آن گروه میشود و همچنین باعث از بین رفتن تمام پیشرفت های انجام شده میشود و همه وظایف متعلق به یک گروه باید دوباره اجرا شوند.بدیعی است که هر چه زمان مسدود شدن وظایف بیشتر باشد،زمان پاسخ گروه هم بیشتر و در نهایت زمان پاسخ کار هم بیشتر خواهد شد.متوسط زمان سرویس کارهای با اولویت بالا برابر با $1/\mu_2$ و متوسط زمان سرویس دسته ای از گروه ها برابر با $1/\mu_1$ میباشد.

۴ استراتژی های زمان بندی

۱.۴ مسیریابی کار

یکی از مهم ترین قسمت های یک سیستم کلاستربندی شده،توزیع کننده سراسری [3] است که مسئولیت توزیع کردن مناسب کارها بین کلاسترها را بر عهده دارد،معیار انتخاب مناسب کلاستر برای کاری که به تازگی وارد سیستم شده است تعداد گروه های متعلق به دسته و توان محاسباتی کلاستر برای پردازش میباشد.اگر تعداد گروه های یک دسته مقادیر ۱ یا ۲ باشند،توزیع کننده سراسری ، کار را به کلاستر اولی که دارای ۱۶ پردازنده است ارسال خواهد کرد و اگر تعداد مقادیر ۳

[1] Bag of Gangs [2] High priority jobs

[3] Grid dispatcher

شکل ۳: ساختار دسته ای از گروه ها

یا ۴ باشند،کار به کلاستر دومی ارسال خواهد شد.احتمال حضور یک کار با اولویت بالا در هر کدام از کلاستر ها برابرا است. توزیع کننده محلی [۴] که در هر کدام از کلاستر ها وجود دارند،مسئولیت توزیع کردن وظایف برای صف های در دسترس را بر عهده دارند.باید توجه کرد وظایفی که متعلق به یک گروه هستند(وظایف هم نژاد) نمیتوانند در اختیار یک پردازنده قرار گیرند به دلیل اینکه یک گروه زمانی میتواند اجرا شود که تمام وظیفه هایش به صورت همزمان اجرا شوند.قاعدتا یک پردازنده نمیتواند یه صورت همزمان دو وظیفه هم نژاد را اجرا کند.

در مجموع تعداد پردازنده های سیستم برابر با ۴۸=۱۶+۳۲ میباشد،نرخ ورود کارهای با اولویت بالا به نسبت تعداد پردازنده ها بسیار کم و برابر با ۱.۱۰،۱.۵=λ_2 است.در حقیقت میتوان گفت در هر واحد زمانی سیستم پاسخگوی ۴۸ کار با اولویت بالا است ولی از آنجایی که نرخ ورود این نوع کار بسیار کم است ما هیچ زمانی در سیستم نخواهیم داشت که تمام پردازنده ها توسط کارهای با اولویت بالا اشغال شوند.با این وجود حتی یک نرخ ورود کم λ_2 اثر قابل توجهی را بر روی عملکرد سیستم میگذارد زیرا زمانی که یک کار با اولویت بالا وارد سیستم میشود پردازنده ای را از یک وظیفه میگیرد و مسدود شدن وظیفه منجر به مسدود شدن تمام وظایف هم نژادش میشود و وظایف مجبورند تا دوباره زمان بندی شوند و همچنین پیشرفت کارشان را از دست میدهند.

۲.۴ مهاجرت

همان طور که قبلا اشاره شد،یک گروه زمانی میتواند شروع به اجرا کند که تمام وظایفش پردازنده ی در دسترسی را در اختیار داشته باشند،پس دلیل اینکه یک گروه نمیتواند اجرا شود این است که حداقل یک یا دو وظیفه اش،پردازنده در دسترسی را در اختیار ندارند و در حال انتظار میباشند.مهاجرت [۵] یک روشی است که از تکه تکه شدن اجتناب میکند.

مهاجرت یک وظیفه را به جلوی صف دیگری انتقال میدهد.با وجود اینکه مهاجرت یک راه حل ایده آل برای تکه تکه شدن است اما بایستی که با احتیاط با آن رفتار کرد،چرا که باعث ایجاد سرآیند میشود.اگر تعداد مهاجرت ها زیاد شود از کارایی سیستم کاسته خواهد شد.دو نوع مهاجرت وجود دارد،مهاجرت محلی که به انتقال یک وظیفه از یک صف به صفی دیگر در همان کلاستر اشاره میکند و مهاجرت سراسری که به انتقال وظیفه از یک صف به صفی دیگری در حالی که متعلق به کلاستر دیگری است ، اشاره دارد.از بین گروه هایی که نیاز به مهاجرت دارند آن یکی

[۴] Local dispatcher

[۵] Migration

را انتخاب میکنیم که به کمترین مهاجرت برای شروع پردازشش احتیاج دارد.هر گروهی که تعداد وظایفش از تعداد پردازنده های در دسترس بیشتر باشد از این روش محروم میشود.در طول مهاجرت ، صف مورد نظر(گره هدف) تا زمانی که وظیفه در حال انتقال است به حالت رزرو میرود تا توسط وظیفه ی دیگری اشغال نشود.اگر باز هم پردازنده در دسترس در اختیار داشته باشیم همین فرآیند تکرار میشود و ما به دنبال گروه هایی خواهیم بود که در حالت انتظار قرار دارند و به کم ترین تعداد مهاجرت برای شروع پردازششان نیاز دارند.بعد از پایان مهاجرت ما مطمئن خواهیم بود که گروه میتواند سریعا پردازش خود را آغاز کند،اما تنها دلیلی که مانع از اجرای گروه میشود ورود یک کار با اولویت بالا است.

لازم به ذکر است،سرآیند مهاجرت بین گره هایی که متعلق به یک کلاستر هستند ۰.۰۵ واحد زمانی میباشد و سرآیند مهاجرت ببین گره هایی که متعلق به کلاستر یکسانی نیستند ۰.۱ واحد زمانی طول میباشد.

۳.۴ زمان بندی گروهی[۶]

در مدل شبیه سازیمان از استراتژی های زمان بندی زیر برای بررسی عملکرد دسته ای گروه ها استفاده کرده ایم.لازم به ذکر است که زمان بند از زمان اجرای وظایف اطلاعی ندارد بلکه تنها از تعداد وظایف یک گروه مطلع است.

۱.۳.۴ اولین ورود تطبیق داده شده،اولین سرویس داده شده[۷]

این زمان بندی در هر کدام از کلاستر ها اجرا میشود.بر اساس این روش یک کار زمان بندی میشود در حالی که تمام وظایف پردازنده های در دسترس را در اختیار دارند.اگر به اندازه کافی پردازنده ی در دسترس برای یک کار موجود نباشد و تعدادی از وظایف یک کار در جلو صف در حال انتظار باشند،پردازنده از وظایف گرفته میشود و در اختیار وظایف پشت سرشان قرار میگیرد.این نوع از زمان بندی تمایل دارد تا کارهایی را در زمان بندی کند که نیاز به پردازنده کمتری دارند و کوچک تر هستند.اما این روش باعث طولانی شدن زمان پاسخ کارهای بزرگ میشود.

۲.۳.۴ بزرگ ترین گروه،اولین سرویس شده[۸]

وظایف متعلق به کارهای بزرگ جلوتر از وظایف متعلق به کارهای کوچک قرار میگیرند،این روش مخالف روش قبلی عمل میکند و بیشتر تمایل دارد تا کارهای بزرگ را زودتر اجرا کند،میتونا گفت این تبعیض قابل قبول است اما پسندیده نیست به دلیل آنکه باعث ایجاد سرآیند میشود.زمانی که یک کار جدید وارد صف پردازنده شود دوباره باید از بزرگ به کوچک مرتب شود.

۳.۳.۴ اولین ورود تطبیق داده شده،اولین سرویس داده شده با مهاجرت[۹]

این روش،همانند زمان بندی "اولین ورود تطبیق داده شده،اولین سرویس داده شده" میباشد با این تفاوت که عمل مهاجرت محلی و سراسری هم پیاده سازی میشود

---

[۶] Gang scheduling

[۷] AFCFS [۸] LGFS [۹] AFCFSwM

۴.۳.۴ بزرگ ترین گروه،اولین سرویس شده با مهاجرت[۱۰]

در این زمان بندی الگوریتم "بزرگ ترین گروه،اولین سرویس شده" به همراه عمل مهاجرت محلی و سراسری هم اجرا میشود.در مورد مهاجرت محلی و سراسری در بخش ۲.۴ توضیحات کامل داده شده است.

۵ ارزیابی و شبیه سازی

۱.۵ معیارهای عملکرد

ما در ارزیابی هایمان از دو معیار برای بررسی عملکرد استفاده کرده ایم.زمان پاسخ r_j یک دسته ای از گروه ها،فاصله زمانی ورود یک دسته ای گروه ها در سیستم تا سرویس کامل همه گروه های متعلق به دسته میباش.اگر m را تعداد کل فرایندهای پردازش شده در سیستم باشد،متوسط زمان پاسخ (؟؟) به صورت زیر تعریف میشود:

$$RT = 1/m \times \sum_{j=1}^{m} r_j \qquad (۱)$$

ما به منظور تخمین تاثیر تعداد وظایف هر دسته بر روی زمان پاسخ کار،معیار جدیدی به نام زمان پاسخ وزنی را تعریف کرده ایم.زمان پاسخ هر دسته هم وزن است با تعداد کل وظایفی که متعلق به آن دسته هستند.$n(x)$ تعداد کل وظایف یک دسته x است.متوسط زمان پاسخ وزنی (۲) از رابطه زیر به دست میاید:

$$WRT = \frac{\sum_{j=1}^{m} n(x_j) \times r_j}{\sum_{j=1}^{m} n(x_j)} \qquad (۲)$$

[۱۰] LGFSwM

جدول ۱: جدول نمادها

P_i	تعداد پردازنده در کلاستر i،$i = 1, 2$
μ_1	متوسط نرخ سرویس وظایف گروه
$1/\mu_1$	متوسط نیاز سرویس وظایف گروه
μ_2	متوسط نرخ سرویس کارهای با اولویت بالا
$1/\mu_2$	متوسط نیاز سرویس کارهای با اولویت بالا
λ_1	متوسط نرخ ورود گروه ها
λ_2	متوسط نرخ ورود کارهای با اولویت بالا
U	متوسط بهره وری پردازنده
RT	متوسطزمان پاسخ گروه ها
D_{RT}	کاهش نسبی RTزمانی که روش Y به جای روش X به کار میرود
WRT	متوسط زمان پاسخ وزنی
D_{WRT}	کاهش نسبی WRTزمانی که روش Y به جای روش X به کار میرود
K	حداکثر تعداد دفعاتی که یک وظیفه در صف پردازنده میتواند توسط وظایفی که به صف منتقل شده اند،کنارزده شود

تمام معیارهای ارزیابی را میتوانید در جدول شماره ۱ مشاهده فرمایید.

۲.۵ پارامترهای ورودی

نتایج شبیه سازی هایمان به طور متوسط از مجموع ۱۰ آزمایش شبیه سازی حاصل شده است.نتایج نشان داده اند که شبیه سازی بیشتر،اثری بر روی نتایج نمیگذارد و همان نتیجه ای را میدهد که شبیه سازی های کمتر میدهد.هر شبیه سازی با موفقیت بر روی ۴۸۰۰۰ دسته ای از گروه ها اجرای خود را به پایان رسانیده است.باید به این نکته توجه کنیم که هر دسته ای

از گروه ها به طور متوسط از تعداد ۴+۱=۲.۵ گروه و هر گروه به طور متوسط از ۸+۱=۴.۵ وظیفه تشکیل شده است.بنابراین میتوان نتیجه گرفت که هر دسته ای گروه ها به طور متوسط شامل ۱۱.۲۵ وظیفه میباشد.بنابراین با داشتن ۴۸۰۰۰ دسته ای از گروه ها به طور متوسط ۵۴۰،۰۰۰ = ۱۱.۲۵ × ۴۸،۰۰۰ وظیفه در حالت موازی توسط سیستم سرویس داده میشود.

در مجموع در سیستم کاری ما شاهد ۴۸ پردازنده هستیم که کلاستر اولی شامل ۱۶ پردازنده (P=۱۶) و کلاستر دومی شامل ۳۲ پردازنده (P=۳۲)میباشد.متوسط نیاز سرویس کارها برابر است با:

$$1/\mu_1 = 1/\mu_2 = 1 \qquad (۳)$$

همان طور که در بالا توضیح داده شد،هر دسته ای از گروه ها به طور متوسط شامل ۱۱.۲۵ میباشد،پس میتوان نتیجه گرفت که در هر واحد زمانی ۴.۱۷۳= ۱۱.۲۵ / ۴۸ وظیفه توسط سیستم میتواند اجرا شود.زمانی که هیچ کار با اولویت بالایی در سیستم وجود ندارد و تمام پردازنده ها در حال پردازش باشند،مقدار برای نرخ ورود کار دسته ای از گروه ها میتواند باشد،ولی با ورود کار با اولویت بالا تعداد پردازنده های در دسترس برای سرویس دهی به دسته ای از گروه ها برابر با میباشد،بنابراین نرخ ورود برای دسته ای از گروه ها مطابق رابطه زیر خواهد شد:

$$\lambda_1 < \frac{48 - (\lambda_2/\mu_2)}{11.5} \qquad (۴)$$

ما در آزمایشات خود از مقادیر زیر برای نرخ ورود دسته ای گروه ها استفاده کرده ایم:

$\lambda_1 =$ ۲.۶ ، ۲.۷ ، ۲.۸ ، ۲.۹

و همچنین ما از دو حالت زیر برای نرخ ورود کارهای با اولویت بالا استفاده کرده ایم:

$\lambda_2 =$ ۱.۵ ، ۱.۱۰

۳.۵ نتایج شبیه سازی

نتایج شبیه سازی ارزیابی عملکرد دسته ای از گروه ها در یک محیط ناهمگن در کلاستره با حضور کارهای با اولویت بالا و عمل مهاجرت را دنبال میکند،متوسط بهره وری پردازنده را در دو روش زمان بندی "اولین ورود تطبیق داده شده،اولین سرویس داده شده " و "بزرگ ترین گروه،اولین سرویس شده " هم با مهاجرت و هم بدون مهاجرت با دو نرخ ورود کارهای با اولویت بالایی که تعریف کرده ایم را میتوانید در جداول ۲و ۳مشاهده نمایید.میتوان نتیجه گرفت زمانی که عمل مهاجرت پیاده سازی میشود ، متوسط بهره وری کلاستر ها هم افزایش می یابد.

همان طور که قبلا اشاره شد،توزیع کننده سراسری بر اساس تعداد گروه ها در هر دسته عمل توزیع کردن را انجام میدهد و انتب میکند که هر دسته وارد کدام کلاستر شود.اگر تعداد گروه ها ۱ یا ۲ باشند دسته وارد کلاستر اولی (P_1=۱۶)و اگر تعداد گروه هایش ۳ یا ۴ باشد وارد کلاستر دومی(P_2=۳۲) میشود.بنابراین میتوان گفت که تعداد وظایف متعلق به یک دسته که برای پردازش وارد کلاستر اولی شده اند به طور متوسط برابر با ۰.۴۲۱ = $\frac{\text{گروه/وظیفه}4.5\times\text{دسته/گروه}1.5}{\text{پروازنده}16}$ و تعداد وظایفی که وارد کلاستر دومی شده اند برابر با ۰.۴۹۲ = $\frac{\text{گروه/وظیفه}4.5\times\text{دسته/گروه}3.5}{\text{پروازنده}32}$ میباشد.از این مقادیر میتوان نتیجه گرفت با پیاده سازی مهاجرت بهره وری در کلاستر دومی بهتر از کلاستر اولی میباشد.

۴.۵ تحلیل عملکرد با توجه به متوسط زمان پاسخ

در شکل ۴ میتوانید زمان پاسخ الگوریتم های زمان بندی "اولین ورود تطبیق داده شده،اولین

جدول ۲: متوسط بهره وری پردازنده،$1/\lambda_2 = 10$

$Cluster2$		$Cluster1$		λ_1
$migrative$	$Nonmigrative$	$Migrative$	$Nonmigrative$	
۰.۶۲	۰.۶۴	۰.۶۰	۰.۵۵	۲.۶
.۶۴	۰.۶۶	۰.۶۲	۰.۵۷	۲.۷
۰.۶۷	۰.۶۹	۰.۶۴	۰.۶۰	۲.۸
۰.۶۹	۰.۷۰	۰.۶۶	۰.۶۲	۲.۹

جدول ۳: متوسط بهره وری پردازنده،$1/\lambda_2 = 5$

Cluster 2		Cluster 1		λ_1
migrative	Non migrative	Migrative	Non migrative	
۰.۶۳	۰.۶۵	۰.۶۰	۰.۵۷	۲.۶
.۶۵	۰.۶۷	۰.۶۳	۰.۵۸	۲.۷
۰.۶۷	۰.۶۹	۰.۶۵	۰.۶۱	۲.۸
۰.۷۰	۰.۷۱	۰.۶۷	۰.۶۳	۲.۹

سرویس داده شده " و "بزرگ ترین گروه،اولین سرویس شده " هم با پیاده سازی رویداد مهاجرت و هم بدون پیاده سازی رویداد مهاجرت مشاهده فرمایید.میتوان نتیجه گرفت که الگوریتم "بزرگ ترین گروه،اولین سرویس شده " نسبت به دو حالت $1/\lambda_2 = 5$ و $1/\lambda_2 = 10$ در الگوریتم "اولین ورود تطبیق داده شده،اولین سرویس داده شده " از زمان پاسخ بهتری برخوردار است.البته این دیدگاد در حالتی است که رویداد مهاجرت در زمان بندی پیاده سازی نشده است.اما در زمانی که مهاجرت در سیستم پیاده سازی میشود متوسط رمان پاسخ کارها به شدت کاهش پیدا میکنند و این بیان میکند که عمل مهاجرت میتواند اثر بسیار قابل توجهی را در عملکرد سیستم از خود نشان دهد.در این حالت الگوریتم "بزرگ ترین گروه،اولین سرویس شده " بهتر از الگوریتم دیگری عمل میکند.

تنها راه برای کنترل تکه تکه شدن اجرای پردازنده ها در سیستم،استفاده از عمل مهاجرت در زمان بندی کارهاست.درست است که مهاجرت

شکل ۴: مقایسه زمان پاسخ

منجر به سرآیند میشود اما به به دلیل اینکه به طور قابل توجهی بر روی عملکرد تاثیر میگذارد در نتیجه میتوان سرآیند را تحمل کرد. زمانی که یک وظیفه در صف پردازنده در حال انتظار میباشد،احتمال اینکه به جلوی صف دیگر برود،بسیار زیاد است.با توجه به نکاتی که قبلا بیان شد،احتمال انتخاب وظایف متعلق به گروهی که تعداد وظایف در حال انتظار کمتری دارند،بیشتر از وظایف دیگر گروه ها است.اثر مهاجرت سراسری بر روی تعادل بار در بین کلاستر ها به مراتب بیشتر از مهاجرت محلی

میباشد.دلایل فوق ما را مجاب میکند تا از الگوریتم های زمان بندی "اولین ورود تطبیق داده شده،اولین سرویس داده شده با مهاجرت" و "بزرگ ترین گروه،اولین سرویس شده با مهاجرت" به جای الگوریتم های "اولین ورود تطبیق داده شده،اولین سرویس داده شده' و "بزرگ ترین گروه،اولین سرویس شده " استفاده کنیم.

شکل ۵: D_{RT} زمانیکه از $AFCFSwM$ به جای $AFCFS$ استفاده میشود

شکل ۶: D_{RT} زمانیکه از $LGFSwM$ به جای $LGFS$ استفاده میشود

در شکلهای ۵ و ۶ کاهش زمان پاسخ در زمانی که عمل مهاجرت بر روی الگوریتم های زمان بندی اعمال شده اند را مشاهده میکنید.به عبارتی دیگر این نمودارها عملکرد بهتر سیستم در هنگام پیاده سازی مهاجرت نشان میدهد.D_{RT} با عملکرد سیستم رابطه مستقیمی دارد،میتوان گفت با افزایش D_{RT}،بهره وروی سیستم افزایش پیدا میکند.

شکل ۷ کاهش زمان پاسخ در حالی که از الگوریتم زمان بندی "بزرگ ترین گروه،اولین سرویس شده $(LGFS)$" به جای "اولین ورود تطبیق داده شده،اولین سرویس داده شده $(AFCFS)$" استفاده میشود،نشان میدهد.با مقایسه این دو الگوریتم میتوان نتیجه گرفت که الگوریتم "بزرگ ترین گروه،اولین سرویس شده "بهتر از 'اولین ورود تطبیق داده شده،اولین سرویس داده شده " عمل میکند. با مقایسه شکل های ۶ و ۷ مشاهده میکنیم که بنابراین حالت دومی () کاهش زمان پاسخ بیشتری نسبت به حالت اولی () دارد،در نتیجه بهتر است. درست است که الگوریتم "بزرگ ترین گروه،اولین

شکل ۷: D_{RT} زمانیکه از $LGFS$ به جای $AFCFS$ استفاده میشود

سرویس شده "بهتر از "اولین ورود تطبیق داده شده،اولین سرویس داده شده " عمل میکند اما تاثیری بر روی مشکل تعادل بار و تکه تکه شدن نمیگذارد.

۵.۵ تحلیل عملکرد با توجه به متوسط زمان پاسخ وزنی

شکل ۸متوسط زمان پاسخ وزنی که به وسیله تعداد وظایف گروه محاسبه میشود،نشان داده شده است.با مقایسه اشکال ۴ و ۸ میتوان نتیجه گرفت که هر چخ تعداد مظایف یک گروه بیشتر

باشد به همان نسبت هم متوسط زمان پاسخ گروه بیشتر میشود.دلیل دیگر بالاتر بودن متوسط زمان پاسخ وزنی از متوسط زمان پاسخ این است که گروه هایی که از تعداد زیاد وظایف تشکیل شده است نیاز به پردازنده های بیشتری برای اجرایشان دارند.بنابراین آنها باید زمان بیشتری منتظر پردازنده بمانند.اشکال ۹ - ۱۱ ،متوسط زمان پاسخ وزنی در همه حالت ها را مورد بررسی قرار داده شده است.نکته قابل توجه این است که بدانید پیاده سازی که افزایش نرخ ورود کارهای با اولویت بالا تاثیری بر روی عمل مهاجرت نگذاشته و عملکرد سیستم به خوبی پیشرفت میکند.

شکل ۸: مقایسه متوسط زمان پاسخ وزنی

۶ جمع بندی و کارهای آینده

ما در این مقاله عملکرد زمان بندی دسته ای از گروه ها در یک سیستم توزیع شده ناهمگن را مورد ارزیابی قرار داده ایم.اثر مهاجرت بر روی عملکرد سیستم مورد بررسی قرار گرفت.نتایج ارزیابی و شبیه سازی حاکی از آن است که مهاجرت علاوه بر کم رنگ کردن مشکل تکه تکه شدن ، باعث ایجاد تعادل بار در سیستم شده است.ما همچنین عملکرد سیستم را در زمانی که کار با اولویت بالا وارد میشود را مورد بررسی قرار دادیم.از دو معیار متوسط زمان پاسخ (RT) و متوسط زمان پاسخ وزنی (WRT) در ارزیابی هایمان استفاده کردیم

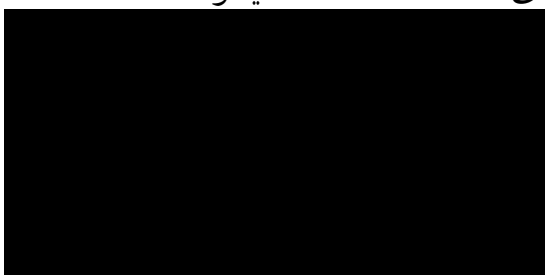
در این مقاله ما زمان بندی گروه ها را با استفاده از الگوریتم های زمان بندی "اولین ورود تطبیق داده شده،اولین سرویس داده شده " و "بزرگ ترین گروه،اولین سرویس شده " هم با مهاجرت و هم بدون مهاجرت مورد بررسی قرار دادیم.در هر حالتی روش زمان بندی "بزرگ ترین گروه،اولین سرویس شده با مهاجرت" بهتر از روش های دیگر رفتار میکند.اما در صورتی که مهاجرت پیاده سازی نشود،روش "بزرگ ترین گروه،اولین سرویس شده"نتیجه بهتری نسبت به "اولین ورود تطبیق داده شده،اولین سرویس داده شده"را از خود نشان میدهد.از کارهایی که میتوان در آینده انجام داد،ارائه راه حلی مناسب برای رفع ناعدالتی در زمان ورود کارهای با اولویت بالا و به کارگیری معیارهای بهتر برای ارزیابی عملکرد زمان بندی دسته ای از گروه های مستقل میباشد.

مراجع

[1] J. Huang and S.-Y. Lee, "A heterogeneity-aware approach to load balancing of computational tasks: a theoretical and simulation study," *Cluster Computing*, vol.11, no.2, pp.133–149, 2008.

[2] G. Sabin, R. Kettimuthu, A. Rajan, and P. Sadayappan, "Scheduling of parallel jobs in a heterogeneous multi-site environment," in *Job Scheduling Strategies for Parallel Processing*, pp.87–104, Springer, 2003.

شکل ۹: D_{WRT} زمانیکه از $AFCFSwM$ به جای $AFCFS$ استفاده میشود



شکل ۱۰: D_{WRT} زمانیکه از $LGFSwM$ به جای $LGFS$ استفاده میشود

شکل ۱۱: D_{WRT} زمانیکه از $LGFS$ به جای $AFCFS$ استفاده میشود