



طراحی و تنظیم کنترل کننده بهینه و هوشمند توازن بار مبتنی بر منطق فازی

درسیستمهای توزیع شده

سید رسول موسوی نژاد^۱، سیدسعیداله مرتضوی^۲، بیژن وثوقی وحدت^۳

^۱ دانشجوی کارشناسی ارشد معماری سیستم های کامپیوتری، دانشگاه آزاد اسلامی، واحد اراک، s.r.mosavi@gmail.com

^۲ عضو هیئت علمی، گروه مهندسی برق، دانشگاه شهید چمران، اهواز، mortazavi_s@scu.ac.ir

^۳ عضو هیئت علمی، گروه مهندسی پزشکی، دانشگاه صنعتی شریف، تهران، vahdat@sharif.edu

چکیده

با پیشرفت تکنولوژی، سیستمهای توزیع شده در زندگی امروزی بسیار گسترش یافته اند. معمولاً الگوی بار سیستم های توزیع شده بسیار سریع و دینامیک در حال تغییر است. موازنه بار یک اصل اساسی در افزایش کارایی سیستم های توزیع شده می باشد. عدم وجود حافظه مشترک بین سیستم های مستقل و تاخیر در کانالهای ارتباطی، موجب ابهام در اطلاع از وضعیت کل سیستم می شود. این ابهام باعث عدم اطمینان در تصمیم گیری موازنه بار می گردد. برای حل این مشکل، در این مقاله یک الگوریتم هوشمند مبتنی بر منطق فازی در سیستم توزیع شده متمرکز پیشنهاد شده است. طول بسته های ارسالی و نرخ سرویس در هر گره متغیر است. ورودی های فازی بار جاری و زمان انتظار در نظر گرفته شده است. توان عملیاتی، میانگین زمان پاسخ و نرخ حذف بسته این الگوریتم با الگوریتم های پویا و ایستا مقایسه شده است و نتایج می دهد که این الگوریتم بهینه تر و کارایی بهتری دارد.

کلمات کلیدی

سیستم های توزیع شده، منطق فازی، تعیین وزن، توازن بار

۱- مقدمه

موازنه بار ایده آل الگوریتمی است که در نهایت باعث شود تمام گره ها همزمان کارهای خود را تمام کنند [۴]. روشهایی که برای توزیع بار ارائه شده است به دو دسته بسیار کلی تقسیم می شوند: الف) الگوریتم هایی که هدفشان یکسان سازی بار پردازشی میان تمام گره ها است ب) الگوریتمهایی که هدف آنها ممانعت از بیکار ماندن بعضی از گره ها در مناطقی است که در سایر گره ها تقاضاها صف کشیده اند. بعبارت دیگر، در این الگوریتم ها به دنبال یکسان سازی کامل بار گره ها نیستند [۵].

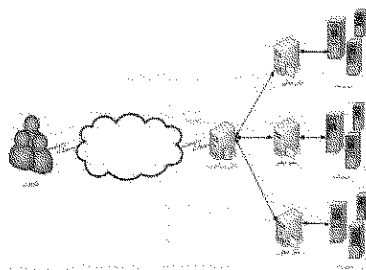
روشهای موازنه بار به دسته بندیهای دیگری مثل مرکزی و غیر مرکزی، ایستا و پویا، دوره ای و غیر دوره ای و نیز دارای حد آستانه و فاقد حد آستانه تقسیم بندی می شود [۶، ۷، ۸، ۹]. در الگوریتم های مرکزی یک گره مرکزی وجود دارد که اطلاعات بار تمام گره ها را جمع آوری کرده و تصمیم مناسب را بر اساس این اطلاعات اتخاذ می نماید. این الگوریتم ها به علت وابستگی به یک گره مرکزی اطمینان نیستند و از کار افتادگی گره مرکزی معادل با از کار افتادن روال موازنه بار است. با این حال از لحاظ پیاده سازی بسیار ساده اند و سر بار ارتباطی کمتری نسبت به روش غیر مرکزی دارند. در مدل های غیر مرکزی (توزیع شده) معمولاً یک یا چند گره خاص به نام سرور یا جمع کننده وجود ندارد، بلکه تمام گره ها دارای اطلاعات تمام یا بخشی از گره های دیگر هستند. از این رو هر گره به تنهایی می تواند در مورد

با رشد روزافزون تکنولوژی محاسبات، کارایی کامپیوترها افزایش و حجم آنها کاهش یافته است. با توسعه کامپیوترهای قدرتمند کوچک، پردازش توزیع شده بعنوان راه حلی بهینه در حصول به قدرتهای پردازش وسیع با استفاده از ظرفیتهای موجود معرفی شده است. در این روش، هر کار به یک سری کارهای کوچک تقسیم شده و به پردازنده ها ارسال می گردند. برنامه های کاربردی ذاتاً توزیع شده، اشتراک اطلاعات میان کاربران توزیع شده، اشتراک منابع، نسبت کارایی به هزینه بهتر، زمان پاسخ کمتر، توان عملیاتی بالاتر، قابلیت اعتماد بیشتر، توسعه پذیری و رشد مضاعف، انعطاف پذیری بهتر در مشاهده و نیاز کاربران باعث شده که در حال حاضر محیط های محاسباتی از حالت مرکزی به حالت توزیع شده تغییر پیدا کنند. لذا سیستم های توزیع شده نقش مهمی در زندگی امروزه دارند. امروزه تکنولوژی های مختلفی برای توسعه و گسترش محاسبات توزیع شده مانند ^۱DCOM، CORBA، ^۲RMI، java وجود دارد [۱، ۲]. موازنه بار یک اصل مهم در سیستم های توزیع شده برای حصول بهره وری بیشتر، دسترسی و پایداری است. بطور کلی، هدف از الگوریتم های موازنه بار توزیع یکسان بار روی منابع و حداکثر کردن کارایی آنها ضمن کم کردن زمان اجرای کلی کارهاست [۳]. در تعریف دیگری، الگوریتم

تنظیم اجزای کنترل کننده فازی از جمله توابع عضویت کار ساده و سر راستی نیست و غالباً کنترل کننده های ارائه شده بهینه و دارای سازگاری کامل درونی نیستند. آقای لی در مقاله خود، یک متد مبتنی بر الگوریتم ژنتیک برای توازن بار نخست-فرستاده در سیستم های توزیع شده ارائه و یک تابع شایستگی مناسب نیز طراحی کرده است. در این طرح، پراسسورهایی که درخواست ها باید به آنها فرستاده شود بوسیله الگوریتم ژنتیک تعیین می شوند. در نتیجه درخواست های غیر ضروری کاهش می یابد. [۱۷]. عیب مهم این روش جستجو زمان-بری آن است که آن را برای کاربرد همزمان ایده آل نمی سازد. در [۱۸] یک الگوریتم فازی با ورودی های زمان cpu و مقدار فضای حافظه استفاده شده ارائه شده و نشان داده شده که کارایی و زمان پاسخ آن در مقایسه با الگوریتم های کلاسیک بهتر است. در این مقاله یک الگوریتم تعادل بار هوشمند مبتنی بر منطق فازی ارائه می شود. کارایی الگوریتم پیشنهادی نسبت به الگوریتم های پویا و ایستا مقایسه می-گردد.

2- مدل سیستم

شکل 1 ساختار یک سیستم توزیع شده ای را نشان می دهد که در این مقاله بعنوان بستر تست استفاده می گردد. ساختار سیستم به صورت سلسله مراتبی است و از روش مرکزی استفاده شده می شود. این مسئله سربار ارتباطی ناشی از جمع آوری اطلاعات را به میزان قابل توجهی کاهش می دهد.



شکل ۱: ساختار استفاده شده

در ساختار طراحی شده کار توازن بار در دو سطح انجام می شود: در سطح گروه توسط مدیر سراسری و در سطح گره توسط مدیر محلی. هر چند گره یک گروه را تشکیل می دهد. در هر گروه یک گره به عنوان تخصیص دهنده بار وجود دارد که کار توازن بار محلی در گروه خود و ارتباط با مدیر سراسری را انجام می دهد. جهت بالا بردن تحمل پذیری خطا در این مقاله گره دیگری در نظر گرفته شده است تا با از کار افتادن گره مدیر جایگزین آن گردد. گره های هر گروه فقط با مدیر گروه در ارتباطند. کار تخصیص داده شده به هر گروه و گره، در همان گروه و گره پردازش شده و جابجا نمی شود. گروه ها با یکدیگر ارتباط ندارند. مدیر سراسری با مدیر هر گروه در ارتباط است.

موازنه بار تصمیم گیری نماید. در روش توزیع شده، مشکل قدیمی شدن اطلاعات گره ها (تاخیر) و سربار زیاد ارتباطی وجود دارد. طبقه بندی مهم دیگری که برای الگوریتم های توازن بار وجود دارد، تقسیم آنها به دو گروه ایستا و پویا است. الگوریتم های ایستا روش هایی هستند که تحت تاثیر شرایط سیستم نبوده و رفتار آنها از پیش تعیین شده است. نمونه هایی از این الگوریتم ها، الگوریتم انتخاب گره تصادفی و انتخاب گره به صورت نوبت گردشی است [۹]. اشکال بزرگ روش های ایستا عدم تطبیق پذیری آنها با تغییرات وضعیت کلی سیستم است. الگوریتم های پویا تصمیماتشان را بر اساس شرایط سیستم اتخاذ می کنند. منظور از شرایط سیستم معمولاً میزان بار پردازشی گره ها است که با معیارهای مختلفی سنجیده می شوند. البته کارایی این الگوریتم ها تاحد زیادی وابسته به انتخاب معیار تخمین میزان بار است. در مجموع این روش ها کارایی و قیاس پذیری بهتری نسبت به روش های ایستا دارند [۱۰]. این کارایی بهتر باعث پیچیده شدن ساختار الگوریتم و افزایش سربار جمع آوری اطلاعات از سیستم می شود.

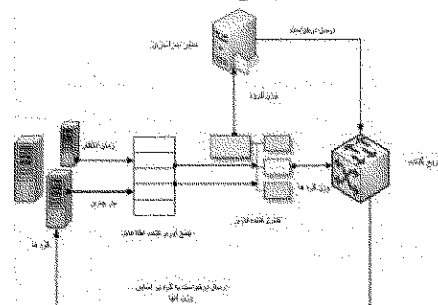
یکی دیگر از روش های توازن بار استفاده از آنالیز عددی و احتمالات جهت پیشگویی وضعیت سیستم و موازنه بار است [۱۱]. اما استفاده از آنالیز عددی دارای محاسبات بسیار پیچیده است. بعلاوه، به دلیل تغییرات زیاد ممکن است موجب ناپایداری در سیستم های توزیع شده شود.

از دیدگاه دیگر، به لحاظ اینکه درخواست موازنه بار از سوی چه گرهی ارسال شود الگوریتم های موازنه بار کلاسیک به سه دسته تقسیم می شوند: 1) راه اندازی توسط فرستنده 2) راه اندازی توسط گیرنده 3) ترکیبی [۱۲]. وجود سربار ارتباطی زیاد و افزایش بار گره های پربار در شرایط پرباری (بدلیل عدم اطلاع و ابهام ذاتی از وضعیت سراسر سیستم) از مشکلات این گونه الگوریتم ها است.

یک الگوریتم توازن بار خوب باید سریع باشد و بار سنگین گره ها اضافه نکند و در ضمن دید درستی از حالت کل سیستم داشته باشد. معمولاً الگوی بار سیستم های توزیع شده بسیار سریع و دینامیک در حال تغییر است. عدم وجود حافظه مشترک بین سیستم های مستقل و تاخیر در کانال های ارتباطی، مستوجب ابهام در اطلاع از وضعیت کل سیستم می شود. این ابهام باعث عدم اطمینان در تصمیم گیری می گردد. برای حل این مشکل، استفاده از هوش مصنوعی و الگوریتم های هوشمند مانند منطق فازی [۱۴] و الگوریتم ژنتیک [۱۵] پیشنهاد می گردد. در [۱۶] یک روش فازی ارائه شده که ابهام از وضعیت سیستم را در نظر گرفته و کارایی بهتری نسبت به الگوریتم های کلاسیک دارد. فاصله گره ها از یکدیگر و میزان بار سیستم، پارامترهای ورودی این الگوریتم هستند. آقای هونگ در مقاله خود، از یک روش مبتنی بر فازی جهت تعادل بار در شبکه های کامپیوتری استفاده کرده است. در این مقاله از دو روش نخست فرستنده و نخست گیرنده استفاده نموده که با تغییر تابع عضویت در حالت فازی الگوریتم جدیدی را ارائه داده است [۱۹]. مشکل این است که تعیین و

3- الگوریتم پیشنهادی

در این الگوریتم از ایده تعیین وزن برای گره ها و گروه استفاده شده است. همانطور که گفته شد تولید وزن در دو سطح مدیرسراسری و محلی صورت می پذیرد براساس وزن تولید شده، درصدی از کارها توسط مدیر سراسری به گروه محلی ارسال شده و در گروه محلی بر اساس وزن گره ها در صدی از کار فرستاده شده به گره ها ارسال می شود. تعیین وزن ها توسط کنترل کننده فازی صورت می گیرد. در این قسمت وزن گره ها و گره ها تعیین شده تصمیمات توزیع بار گرفته می شود. معیار تصمیم گیری دو متغیر بار جاری و زمان انتظار آخرین کار پردازش شده در هر گره در نظر گرفته شده است. شکل 2 ساختار کنترل کننده فازی را نشان می دهد.



شکل 2: اجزای مدل فازی وتبادل داده ها بین آنها
فرایند موازنه بار در 3 مرحله صورت می گیرد:

3-1- مرحله جمع آوری اطلاعات

کنترل کننده فازی برای تولید وزن نیازمند اطلاعات است. هر گره در زمان تعیین شده ای به نام زمان تعیین وضعیت، اطلاعات وضعیتی خود را برای گره مدیرگروهش میفرستد. هر مدیر محلی براساس اطلاعات دریافتی از گره های گروهش کار تولید وزن را انجام می دهد. همچنین بر اساس این اطلاعات، کنترل کننده فازی وزن کل گروه را تعیین کرده و برای مدیر سراسری می فرستد. مدیر سراسری بر اساس اطلاعاتی که از هر گره در اختیار دارد، بار را بین گروه ها توزیع می کند. برای تبادل اطلاعات هر کدام از گره ها وضعیت بار خود را به صورت دوره ای به اطلاع گره مدیر می رساند. یعنی برای تعیین وضعیت جاری گروه ها و گره ها، اطلاعات متوازن کننده در سیستم بدون اینکه درخواستی از طرف آنها، برای اطلاعات وضعیتی ارسال شود، بروز می شود. مدیر هر گروه بر اساس اطلاعات دریافتی از گره ها، وضعیت سراسری گروه خود را برای مدیرسراسری می فرستد. با این روش خیلی از درخواستهای بیهوده و اضافی از سیستم حذف و سربار سیستم کاهش قابل توجهی می یابد. مهمترین مساله در این روش تعیین بازه زمانی برای تبادل اطلاعات است. بازه های بسیار طولانی برای تبادل اطلاعات باعث تصمیم گیری غیر دقیق می شود، در حالیکه بازه های زمانی کوتاه حجم تبادلات را به شدت افزایش می دهد.

3-2- مرحله تعیین وزن

این مرحله در کنترل کننده منطق فازی صورت می گیرد. کنترل کننده منطق فازی مهمترین جزء کنترل است. در این قسمت وزنه های گره ها و گره ها تعیین شده و براساس وزنه ها، تصمیمات توزیع بار گرفته می شود.

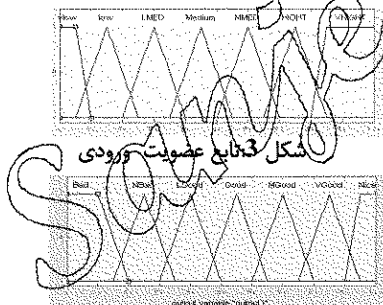
در الگوریتم های مختلف معیارهای مختلفی برای تعیین بار سیستم در نظر گرفته شده است. در واقع یکی از مشکلات الگوریتم های فعلی نداشتن معیاری مناسب برای سنجش میزان بار سیستم است. در الگوریتم های مختلف معیارهایی همچون نرخ ورود درخواست [۲۱]، میانگین زمان تکمیل همه پردازشها و میانگین زمان پاسخ گره ها [۲۰] و یا ترکیبی از آنها [۲۱] به عنوان معیار سنجش در نظر گرفته شده است. در این روش از بار سیستم و زمان انتظار پردازش گره ها در بازه زمانی به عنوان پارامترهای ورودی کنترل کننده فازی جهت تعیین وزن استفاده شده است. زمان انتظار دیگر گره ها نسبت به حداکثر زمان انتظار در یک بازه زمانی سنجیده می شوند.

چون از منطق فازی استفاده شده است و این منطق مبتنی بر قانون است برای هر کدام از ورودی ها هفت زیر مجموعه فازی برای تعیین قوانین در نظر گرفته شده است. با استفاده از این ورودی ها ۷۷ قانون بدست آمده است که در زیر به تعدادی از آنها می پردازیم.

۱) If (current_load is Vlow) and waiting_time is Small) then (output^۱ is Nice)

۲) If (current_load is VHigh) and (waiting_time is Med) then (output^۱ is Bad)

به عنوان نمونه در قانون ۱ اگر بار جاری خیلی کم و زمان انتظار کوچک باشد خروجی فازی که نشان دهنده وضعیت گره است بسیار عالی بوده و وزن آن زیاد و تعداد کارهای ارسالی به نسبت زیادتر تصمیم گیری می شود. در قانون ۲ اگر گره خیلی پر بار و زمان انتظار آن متوسط باشد وضعیت گره بد بوده و کاری به آن ارسال نمی گردد. در شکل 3 و 4 توابع عضویت برای دو ورودی و خروجی نشان داده شده است.



شکل 4: تابع عضویت خروجی فازی

3-3- مرحله توزیع

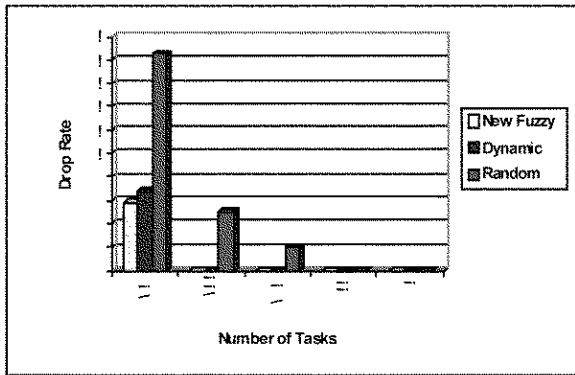
در این مرحله توزیع کننده، بار پردازشی درخواستی را روی گره ها براساس وزن تعیین شده برای آنها توزیع می کند. مدیر سراسری بر

اساس وزن گروه که توسط کنترل کننده فازی تعیین و ارسال شده است، تعدادی از درخواست ها را به توزیع کننده ارسال می کند و توزیع کننده بر اساس وزن هر گروه تعدادی از کارها را بین گروه ها توزیع می کند.

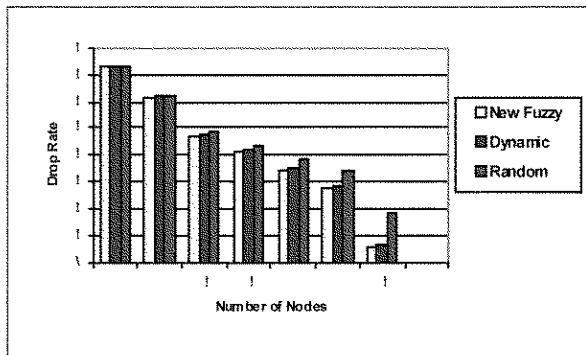
4- شبیه سازی و مقایسه کارایی الگوریتم پیشنهادی

در این مقاله کارایی سه الگوریتم دینامیک، تصادفی و الگوریتم فازی پیشنهادی، بر اساس سه معیار $response$ ، $throughput$ ، $drop rate$ ، time مقایسه شده اند. برای ارزیابی کارایی الگوریتم، شبیه سازی در چندین شرایط متفاوت انجام شده است. مقایسه ها به دو صورت ابتدا بر اساس تعداد مختلف گره ها با تعداد کار ثابت و سپس برای تعداد مختلف کارها با تعداد گره ثابت انجام شده اند. گره های هر گروه از نظرمیزان حافظه و نرخ سرویس متفاوت تعیین شده اند. مشخصات کلی گروهها مشابه اند. وقتی تعداد کارها متفاوت است، تعداد گره ها در هر گروه ۱۳ است. وقتی تعداد گره ها ثابت است، تعداد کارها ۴۴۰۰۰ بسته است. در سیستم سه تولید کننده کار از طریق لینک ارتباطی با مدیر سراسری ارتباط دارند و درخواستها را به آن تحویل می دهند. در لینک های ارتباطی نیز مقداری منطقی تاخیر در نظر گرفته شده است. طول بسته ها متغیر در نظر گرفته شده است. برای اینکه تفاوت بین الگوریتم طراحی شده بهتر مشخص گردد، تولید کننده های کار به صورتی طراحی شدند که نرخ ورود به سیستم غیر قابل پیش بینی باشد. این وضعیت تفاوت بین الگوریتم های ایستا و پویا را بهتر مشخص می کند. در سیستم های واقعی نیز هیچگاه نرخ ورود به سیستم ثابت نیست و در لحظات مختلف متفاوت است. نمودارهای زیر نتایج آزمایشهای بدست آمده در محیط نرم افزار OPNET را در حالت های مختلف نشان می دهد.

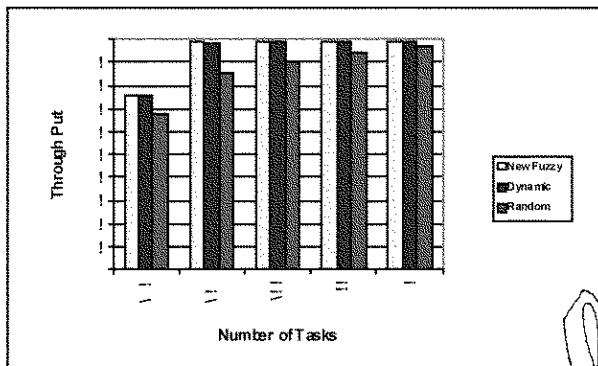
شکل ۵ drop rate را برای تعداد ۱۳ گره ثابت و کارهای مختلف نشان می دهد. همانطور که دیده می شود الگوریتم فازی زمانی که بار زیاد است نسبت به دیگر الگوریتم ها بهتر عمل می کند. شکل ۶ بیانگر حالتی است که تعداد گره ها بین ۱۵ تا ۳۹ گره در حال تغییر و تعداد کارها ۴۴۰۰۰ بوده است. همانطور که در شکل می بینید در حالتی که تعداد گره ها کم است فاصله بین الگوریتم ها نیز کم بوده و با افزایش تعداد گره ها الگوریتم فازی با فاصله ای محسوس drop rate کمتری نسبت به الگوریتم دینامیک و تصادفی دارد. در کل در دو حالت الگوریتم فازی drop rate کمتری نسبت به دو الگوریتم دیگر داشته است. شکل ۷ و شکل ۸ throughput را برترتیب در حالت های گره ثابت و کار متغیر و حالت گره متغیر و کار ثابت نشان می دهد که وضعیت الگوریتم فازی در هر دو شرایط بهتر و بهینه تر است. شکل ۹ میانگین زمان پاسخ را نشان می دهد که الگوریتم فازی زمان کمتری نسبت به دیگر الگوریتم ها دارد. در مجموع نتایج نشان می دهند که الگوریتم فازی کارایی بیشتر و زمان پاسخ کمتری نسبت به دو الگوریتم دینامیک و تصادفی داشته است.



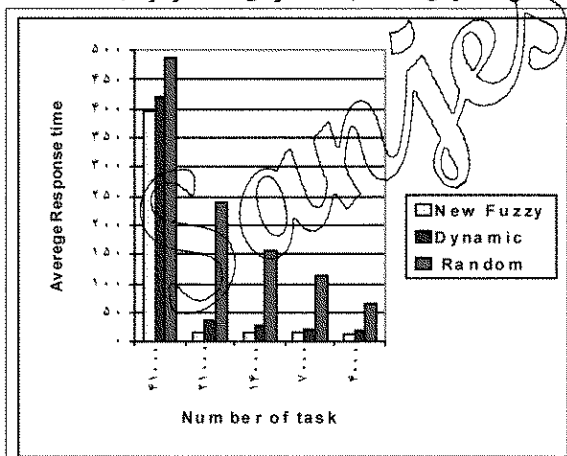
شکل ۵: drop rate با تعداد کارهای مختلف و گره ثابت



شکل ۶: drop rate با تعداد گره های مختلف و کار ثابت

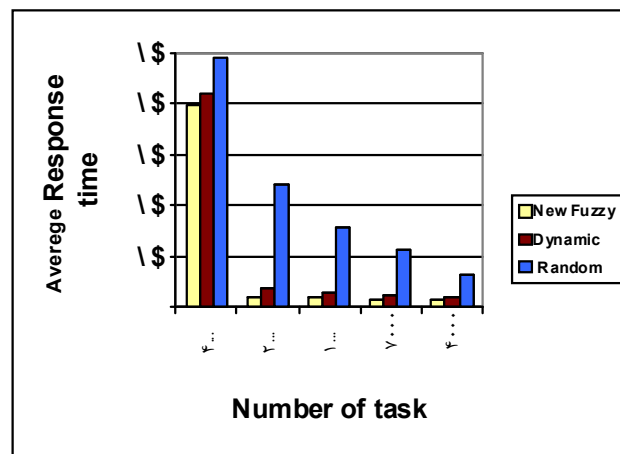


شکل ۷: Throughput با تعداد کارهای مختلف و گره ثابت



شکل ۸: Throughput با تعداد گره های مختلف و کار ثابت

- [۶] F. Bonomi and A. Kumar, *Adaptive Optimal Load-Balancing in a Heterogeneous Multiserver System with a Central Job Scheduler*, IEEE Trans. Computers, vol. ۳۹(۱۰), pp. ۱۲۳۲-۱۲۵۰, ۱۹۹۰.
- [۷] H.C. Lin and C.S. Raghavendra, *A Dynamic Load-Balancing Policy with a Central Job Dispatcher (LBC)*, IEEE Trans. Software Eng., vol. ۱۸, no(۲), pp. ۱۴۸-۱۵۸, ۱۹۹۲.
- [۸] Y. Lan and T. Yu, *A Dynamic Central Scheduler Load-Balancing Mechanism*, Proc. IEEE ۱۴th Ann. Int'l Phoenix Conf. Computers and Comm., pp. ۷۳۴-۷۴۰, ۱۹۹۵.
- [۹] L.SUN CHEUNG, *A fuzzy approach to load balancing in a distributed object computing network*, In Proc. Of ۹th Int IEEE Conf. HPDC, ۲۰۰۰.
- [۱۰] J. L. Deneubourg, S. Goss, N. Franks, *The dynamics of collective sorting robot-like ants and ant-like robots*. In From Animals to Animates, Proc. of the First Int. Conference on Simulation of Adaptive Behavior, pages ۳۵۶-۳۶۳. MIT Press, ۱۹۹۰.
- [۱۱] E. Gelenbe and R. Kushwaba, *Dynamic Load Balancing in Distributed Systems*. IEEE, ۱۹۹۲.
- [۱۲] K.Ming V. Yu, and C. Chou, *A Fuzzy-Based Dynamic Load-Balancing Algorithm*, Proc. Of fuzzy System Application Symp. IEEE Press ۲۰۰۰.
- [۱۳] Sandeep S. W. "Classification of dynamic load balancing strategies in a Network of Workstations". Fifth International Conference on Information Technology: New Generations. IEEE. ۲۰۰۸, pp ۱۰۰۵-۱۰۱۴.
- [۱۴] Kasabov, Nikola K. *Foundations of neural networks, fuzzy systems, and knowledge engineering*. The MIT Press. ۱۹۹۸.
- [۱۵] Albert Y. Zomaya and Yee-Hwei. *Observations on Using Genetic Algorithms for Dynamic Load-Balancing*. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. ۱۲, NO. ۹, ۲۰۰۱.
- [۱۶] Chulhye Park and Jon G. kuhl. *A Fuzzy-Based Distributed Load Balancing Algorithm for Large Distributed Systems*. Second International Symposium on Autonomous Decentralized Systems (ISADS'۹۵) p. ۰۲۶۶, IEEE. ۱۹۹۵.
- [۱۷] Seong-hoon Lee et al. *A Genetic Algorithm Method for Sender-based Dynamic Load Balancing Algorithm in Distributed Systems*. ۱۹۹۷ First International Conference on Knowledge-Based Intelligent Electronic Systems, ۲۱-۲۳ May ۱۹۹۷, Adelaide, Australia.
- [۱۸] Hyo Cheol Ahn, Hee Yong Youn, Kyu Yeong Jeon, and Kyu Seol Lee. *Dynamic Load Balancing for Large-scale Distributed System with Intelligent Fuzzy Controller*. IEEE International Conference on Information Reuse and Integration. Aug ۲۰۰۷. pp. ۵۷۶ – ۵۸۱.
- [۱۹] Ming-Chang Huang. *load balancing and congestion control using fuzzy logic control method in computer networks*. The University of Wisconsin - Milwaukee December ۲۰۰۳.
- [۲۰] B. S. Joshi, Seyed H. Hosseini, *A Methodology for Evaluating Load Balancing Algorithms*, in Proc. Of ۹th Int. Conf. IEEE HPDC, ۱۹۹۳.
- [۲۱] Z. Zeng and B. Veeravalli, *Rate-Based and Queue-Based Dynamic Load Balancing Algorithms in Distributed Systems*, ۱۰th Int. Conference on Parallel and Distributed Systems, IEEE ۲۰۰۰.



شکل 9- میانگین زمان پاسخ

5- نتیجه گیری

در این مقاله یک روش هوشمند مبتنی بر منطق فازی در سیستم‌های توزیع شده متمرکز ارایه شده است. کارایی این الگوریتم در یک مجموعه شبیه‌سازی در فضای OPNET بررسی شده است. طول بسته-های ارسالی و نرخ سرویس در هر گره، متغیر در نظر گرفته شده است. بار جاری سیستم و زمان انتظار آخرین کار پردازش شده در هر گره به عنوان ورودی‌های کنترل کننده فازی و وزن هر گره به عنوان خروجی آن در نظر گرفته شده است. بر اساس وزن تولید شده برای هر گره درصدی از کارهای موجود به گره‌ها اختصاص داده می‌شود. نتایج آزمایش‌ها در حالت‌های گره‌های ثابت و متغیر، و کارهای ثابت و متغیر نشان می‌دهد که این الگوریتم نسبت به الگوریتم‌های ایستاد پویا از لحاظ Response Time و Drop Rate و Throughput بهینه‌تر عمل می‌کند.

مراجع

- [۱] Thomas Schnekenburger, "Load Balancing in CORBA: A Survey of Concepts, Patterns, and Techniques", The Journal of Supercomputing, ۲۰۰۰, pp. ۱۴۱-۱۶۱.
- [۲] L.-S. Cheung, Y.-K. Kwok, "On Load Balancing Approaches for Distributed Object Computing Systems", The Journal of Supercomputing, ۲۰۰۴, pp. ۱۴۹-۱۷۵.
- [۳] A. Y. Zomaya and Y. Teh, *Observations on using genetic algorithms for dynamic loadbalancing*, IEEE Trans. on Parallel and Distributed Systems ۹ (September ۲۰۰۱), pp. ۸۹۹-۹۱۱.
- [۴] B. Veeravalli and W. Han Min, *Scheduling Divisible Loads on Heterogeneous Linear Daisy Chain Networks with Arbitrary Processor Release Times*, IEEE TRANS. ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. ۱۵(۳), MARCH ۲۰۰۴.
- [۵] O. Krem and J. Kramer, *Methodical Analysis of Adaptive Load Sharing Algorithms*, IEEE Trans. On High Performance Computing Vol ۴(۳), ۱۹۹۲.

۱ common object broker architecture
 ۲ remote method invocation
 ۳ Sender- Initiated
 ۴ Receiver-Initiated