مطالعات علوم کاربردی در مهندسی

دوره ۸، شماره ۱، بهار ۱۴۰۱

صفحات ۲۲۰-۲۰۳

Online ISSN: 2476-4507

Print ISSN: 2538-2055

www.irijournals.com

یک روش جدید برای بهبود انتخاب داده و قابلیت اطمینان در گرید با استفاده از ترکیب الگوریتم خفاش با اتاماتای یادگیر

مهدی صادق زاده[۱]، سارا ایوبی[۲]

[۱] هیات علمی گروه مهندسی کامپیوتر، واحد ماهشهر، دانشگاه آزاد اسلامی، ماهشهر، ایران

[۲] دانشجوی ارشد گروه مهندسی کامپیوتر، واحد بوشهر، دانشگاه آزاد اسلامی، بوشهر، ایران

چکیده

در گرید مدیریت حجمی گسترده‌ای از داده‌ها در یک محل متمرکز، زمان دستیابی به داده‌ها را افزایش می‌دهد بنابراین زمان بیشتری برای انجام کار لازم است پس برای کاهش زمان دستیابی داده‌ها از تکثیر برداری استفاده می‌شود. تکثیر برداری تکنیکی است که در سیستم‌های گرید به‌منظور بهبود زمان پاسخ برنامه‌های کاربردی و کاهش مصرف پهنای باند مورد استفاده قرار می‌گیرد. در عمل تکثیر برداری تعداد زیادی از یک فایل تولید می‌شود و در کل سیستم پخش می‌شوند و با هدایت کاربران به در دسترس ترین، نزدیک‌ترین و سریع‌ترین کپی از اطلاعات مورد نیاز آن‌ها هزینه کاهش می‌یابد. نکته قابل‌توجه این است که نسخه‌های هر فایل در کدام سایت و در کجای سیستم توزیع شده قرار گیرند. جایگذاری مناسب نسخه‌ها در مکان مناسب موجب کاهش زمان اجرای کارها می‌شود؛ زیرا فرایند جایگیری بهینه نسخه‌ها به کارها این امکان را می‌دهد که از نسخه‌ای از فایل استفاده کنند که کارایی و زمان دسترسی بهتری داشته باشند. در این پژوهش سه روش جدید جهت جایگذاری بهینه نسخه‌های ایجادشده در گرید داده با استفاده از الگوریتم خفاش و ماشین یادگیرنده ارائه شده است. که الگوریتم‌های پیشنهادی بر پایه الگوریتم خفاش است و ماشین یادگیرنده وظیفه تنظیم سرعت خفاش‌ها را بر عهده دارد. نتایج شبیه‌سازی در محیط MATLAB بهبود کارایی این روش‌ها نسبت به روش‌های قبلی را نشان می‌دهد.

واژه‌های کلیدی: الگوریتم خفاش، تکثیر داده، گرید داده، ماشین یادگیرنده

مقدمه

در جهان امروز علاقه استفاده از اینترنت و همچنین دستیابی آسان به رایانه‌های قدرتمند در کنار شبکه‌های سرعت بالا با پایین ترین قیمت، روش استفاده از رایانه و انجام محاسبات را تغییر داده است. با توجه به حل مسائل بسیار سنگین و با استفاده از ترکیب منابع محاسباتی توزیع شده از لحاظ جغرافیایی علم جدیدی به نام گرید تولید شده است. در اواخر دهه ۱۹۹۰ شبکه‌های محاسباتی به عنوان جایگزینی برای سوپر کامپیوتر های متداول ارائه شد تا به حل مسائلی که نیاز به محاسبات عددی فراوان و همچنین دسترسی به حجم زیادی از داد ه های توزیع یافته داشتند، بپردازد.

محاسبات گریدی در اوایل دهه ۹۰ در انجمن ابر محاسبات مطرح شدو هدف آن به کارگرفتن منابع محاسباتی موجود به شکلی آسان برای محاسبات پیچیده توسط سایت هایی بود که به صورت جغرافیایی توزیع شده می باشند. ایده ی اصلی گرید این است که منابع ماشین های شرکت کننده، از طریق یک لایه نرم‌افزاری به صورت شفاف و مطمئن مورد استفاده قرار بگیرد. این لایه نرم‌افزاری، وظایفی همچون مجازی سازی منابع، یافتن و جستجوی منایع، و مدیریت برنامه‌های در حال اجرا را بر عهده دارد. محاسبات گریدی، در ساده ترین بیان، محاسبات توزیع شده ای که به سطح تکامل بالاتری رسیده است. هدف این است که تصویری از یک کامپیوتر ها بوجود بیاید. این مجموعه از یک سری سیستم‌های نا همگون متصل به هم ترکیبات مختلفی از منابع را به اشتراک گذاشته اند، تشکیل شده است.

گرید محاسباتی به عنوان یک رویکرد جدید برای حل مسایل در مقیاس بزرگ در زمینه های علمی، مهندسی و تجاری بوجود آمده اند آنها این قابلیت را بوجود می آورند که بتوان بنگاه های مجازی را به‌منظور اشتراک ویکپارچه سازی میلیونها منبع که از نظر جغرافیایی در سطح سازمآن‌ها و حوزه های مدیریتی پراکنده اند، ایجاد کرد آنها از یک سری منابع ناهمگون (کامپیوترهای شخصی، ایستگاه های کاری، کلاسترها وابر کامپیوترها) سیستم‌های مدیریت زیر بنایی (سیستم عامل واحد، سیستم‌های صف و غیره) سیاست ها و برنامه‌های کاربردی (علمی، مهندسی و تجاری) با نیازمندیهای مختلف (پردازنده، ورودی و خروجی، حافظه و شبکه) تشکیل شده اند. منابع، متعلق به سازمآن‌های مختلف می باشد که دارای سیاست های مدیریتی و مدل های استفاده و هزینه ای مخصوص به خود برای کاربران مختلف در زمآن‌های گوناگون می باشند. همچنین فراهم بودن منابع و میزان بار بر روی آنها در طول زمان تغییر می کند.

یک الگوریتم انتخاب داده با هدف بهینه سازی هزینه برای گریدهای داده، بایستی با توجه به قیمت و توانمندی منابع گرید، عمل تخصیص آنها به کارهای ناهمگون کاربر را طوری انجام دهد که اجرای کارها با حداقل هزینه و قبل از مهلت تعیین شده پایان یابد، ضمن اینکه اطمینان آن هم منظورشده باشد. در گرید داده، تکرار داده در گره های مختلف باعث افزایش دسترس پذیری و کاهش زمان پاسخ می‌شود. محیط گرید به علت وجود هزاران کاربر متفاوت، بسیار پویا می باشد. از طرفی در گرید با حجم عظیم داده‌ها مواجه هستیم که برای افزایش دسترس پذیری این حجم عظیم داده در سیستم‌های گرید نیازمند سیاست های جدیدی می باشیم. در نهایت گرید، با افزایش کارایی و کاهش هزینه ها، اشتراک گذاری و مدیریت بهتر فایل های بسیار بزرگ را در سرتاسر دنیا فراهم می کند وما سعی داریم که در این کار تحقیقاتی به نحوی بااستفاده ازالگوریتم بهینه سازی واکنش های شیمیایی قابلیت اطمینان و انتخاب داده در گرید را بهبود بخشیم.

گرید، یک محیط توزیع شده ناهمگن است که هریک از منابع موجود در آن می تواند متعلق به یک سازمان و یا شخص به خصوص باشند و هدف از آن حل مسائلی است که فقط در محیط‌های توزیع شده میسر است. از دیدگاه کاربر, گرید یک محیط همکاری برای حل مسائل است بدون در نظرگرفتن اینکه این منابع در چه نقطه ای واقع گردیده‌اند ویا متعلق به چه کسی

هستند. در عمل مدیریت منابع و حفظ امنیت به عهده محیط گرید می باشد. در این میان یکی از انواع مهم گرید، گریدهای داده هستند. گرید داده یک سیستم توزیع شده است که سرویسهایی را جهت کشف، انتقال و دستکاری حجم عظیم داده‌هایی که در این سیستم توزیع شده، ذخیره شده است، فراهم می‌کند. مسئله اساسیدرگرید داده، کاهش ترافیک شبکه است. در گرید، کارها و کاربران نیازمند دسترسی به حجم عظیمی از داده‌های ذخیره شده در سراسر محیط جغرافیایی می باشند. پس دراین زمینه باید اطمینان وامنیت تا حد قابل قبولی تامین شود. درگرید که هدف خدمت رسانی خوب ومطمئن به سرویس گیرنده های خود می باشد، چالش هایی درزمینه قابلیت اطمینان۱ وجود دارد و هدف از این تحقیق ارائه یک روش، بوسیله الگوریتم بهینه سازی واکنش های شیمیایی، جهت بالابردن قابلیت اطمینان وانتخاب داده در شبکه‌های گرید. است. این مقاله، به بررسی نواقص کارهای انجام شده و روش های گوناگون که هرکدام جهت بالابردن قابلیت اطمینان در گرید می باشند، پرداخته و با هدف شناسایی نمونه های متعدد گرید و بررسی زمینه های کاربردی آن وبالابردن قابلیت اطمینان انجام گردیده است.

مبانی نظری

الگوریتم خفاش (BA)

الگوریتم خفاش بر مبنای ویژگی بر گرفته از رفتار پژواک یا انعکاس صدای خفاش ها است. قابلیت اعکاس صدا در خفاش های کوچک بسیار جالب است، چرا که این خفاش ها می توانند از این طریق طعمه خود را یافته و حتی در تاریکی کامل نوع حشره را تشخیص داده و ان را شکار نماید.

بیشتر خفاش ها از توانایی انعکاس صدا برخوردارند و هر کدام بسته به شرایط از درجه خاصی برای انعکاس صدا استفاده می کنند. در میان تمام گونه ها، خفاش های کوچک نمونه بارزی از آنها هستند که از ویژگی انعکاس صدا در محیط، طوری استفاده می کنند که اندازه آن با مگابایت هم محاسبـه نمی‌شود بیشتر خفاش های کوچک حشره خوار هستند. این نوع خفاش ها از یک نوع انعکاس صدا به نام سونار برای شناسایی طعمه، اجتناب (فرار) از خطر، و کمین کردن یا خواب در تاریکی استفاده می کنند. این خفاش ها صداهای بلند را تولید می کنند و به انعکاس آن در اطراف گوش می دهند. پالس صدای آنها بسته به شرایط متفاوت است و می تواند با نوع گونه، و استراتژی شکار نیز در ارتباط باشد. بسیاری از خفاش ها از سیگنال های متناسب و اندازه با آنچه که از جای دیگر دریافت می کنند، استفاده می کنند. در حالیکه بیشتر اوقات از سیگنال ها با فرکانس ثابت برای انعکاس صدا در محیط استفاده می کنند. پهنای باند سیگنال های آنها متفاوت است و به گونه آنها نیز بستگی دارد و اغلب در صورت استفاده از هارمونی های بیشتر افزایش میابد.

اگرچه هر پالس چند هزارم ثانیه به طول میی انجامد با این حال دارای فرکانس ثابت در حدود ۰,۲ تا ۰,۱ کیلوهرتز می باشد. محدوده استاندارد فرکانس ها در گونه های مختلف خفاش ها در حدود ۲۵ الی ۱۰۰ کیلو هرتز می باشد، هر چند برخی از این گونه ها می توانند فرکانس های بالا تری را نیز منتشر کنند مثلا در حدود ۱۵۰ کیلوهرتز.

هر انفجار مافوق صوت ممکن است ۵ تا ۲۰ هزارم ثانیه طول بکشد و خفاش های کوچک به طور معمول این انفجار صدا را ۱۰ تا ۲۰ بار در هر هزارم ثانیه تکرار می کنند. مثلا در هنگام شکار طعمه میزان پالس آزاد شده می تواند با سرعت حدود ۲۰۰ پالس در ثانیه تا لحظه نزدیکی به شکار طعمه خود این صدا را تولید کنند. بعضی از پیام های همزمان صوتی کوتاه به توانایی

[1] Reliability

دوره ۸، شماره ۱، بهار ۱۴۰۱، صفحات ۲۲۰-۲۰۳

فوق العاده ای در قردت پردازش سیگنال خفاش ها اشاره دارند در حقیقت مطالعات زمان اتحاد گوش خفاش را نشان میدهد که به طور معمول حدود ۳۰۰ الی ۴۰۰ میکرو ثانیه می باشد

با توجه به اینکه سرعت صوت در هوا به طور معمول $v = ۳۴۰\ m/s$ و طول موج انفجار صدای ما فوق صوت با فرکانس ثابت f داده شده است:

که در محدوده ۲ الی ۱۴ میلی متر برای محدوده فرکانس معمولی ۲۵ کیلو هرتز تا ۱۵۰ کیلو هرتزاست.

معمولا این طول موج با اندازه طعمه در ارتباط است.

شگفت آور اینست که پالس ساطع شده می تواند صدای بلندی در حدود ۱۱۰ دسی بل باشد و خوشبختانه در محدوده ما فوق صوت به حساب می آیند. هنگامیکه خفاش‌ها به دنبال شکار خود هستند کاملا بی صدا هستند و هیچ صدایی تولید نمی کنند محدوده مسافت (عبور و مرور) این پالس ها ی کوتاه عموما چند متر می باشد که البته به فرکانس های واقعی بستگی دارد. خفاش های کوچک می توانند برای اجتناب از موانع به کوچکی ریزی یک تار موی انسان کار خود را انجام دهند.

مطالعات نشان می‌دهد که خفاش های خیلی کوچک از تاخیر زمانی در پخش و انتشار صدا استفاده میکنند و از تفاوت زمانی بین گوش های انها در شنیدن صدا و تغییرات بلندی انعکاس صدا برای ساختن سناریو سه بعدی احاطه شده استفاده میکنند. آنها می توانند از راه دور هدف را شناسایی و به سوی ان جهت گیری نمایند و حتی نوع طعمه در حال حرکت را نیز شناسایی کرده و حتی سرعت ان را تخمین می زنند.

در واقع، مطالعات نشان می دهند که به نظر می رسد خفاش ها می توانند با استفاده از اثر داپلر که از بال بال زدن حشرات القا می‌شود، می توانند هدف (شکار) خود را شناسایی کنند.

بدیهی است که برخی از خفاش‌ها بینایی خوبی دارند و بسیاری از آنها نیز دارای حس بویایی فوق العاده ای هستند. در واقع انها تمامی حواس خود را با هم ترکیب می کنند تا در شناسایی شکار خود کارآمد باشند. با این حال، اکنون، ما بیشتر علاقه مندیم تا درباره ویژگی انعکاس صدا (مکان یابی) و رفتار محیطی آنها بدانیم.

این انعکاس صدا ی خفاشهای کوچک، می تواند به گونه ای ساختار بندی شود که با تابع اصلی در بهینه سازی، مرتبط باشد و این امکان را می‌دهد تا یک الگوریتم بهینه سازی جدید را ساختار بندی کنیم. در ادامه این مقاله، ما برای اولین بارررئوس کلی مطالب را درفرمول اساسی الگوریتم خفاش ها (BA) طرح ریزی نموده ایم و سپس به بحث درباره اجرا و پیاده سازی آن و مقایسه جزییات خواهیم پرداخت.

قوانین الگوریتم خفاش

با توجه به ویژگی پژواک یابی خفاش ها Yang Xin-she در سال ۲۰۱۰ الگوریتم خفاش را بر مبنای سه قانون اساسی زیر بنا نهاد:

۱-تمامی خفاش‌ها با بکارگیری پژواک یابی می توانند مسافت را تخمین بزنند و تفاوت بین طعمه و موانع ثابت را تشخیص دهند

۲- خفاش ها به صورت تصادفی با سرعت V_i در جهت X_i با فرکانس F_{min} به طول و بلندی A_0 در جست و جوی شکار هستند همچنین آنها می توانند طول موج پالس های منتشر شده را به طور اتوماتیک تنظیم کردهو نرخ انتشار پالس خود را طبق نزدیکی طعمه ی خود مطابقت دهند

۳-بلندی صدا می تواند به طرق مختلفی تغییر کند ولی فرض می کنیم که بلندی صدا از A_0 تا A_{min} تغییر می کند.

حرکت خفاش‌ها

هر خفاش به سرعت V_i^t و مکان X_i^t با تکرار t در یک جست و جوی d بعدی و یا فضای راه حل وابسته است. در میان تمامی خفاش ها تنها یک راه حل بهینه ی x^* وجود دارد

بنابراین سه قانون ذکر شده در قسمت قبل را می توان از طریق فرمول های زیر محاسبه کرد:

$$Fi=F_{min}+ (F_{max} - F_{min}) \beta \quad (۱)$$

$$V_i^t =V_i^{t-1} + (X_i^{t-1} + X^*) \quad (۲)$$

$$X_i^t =X_i^{t-1} + V_i^t \quad (۳)$$

بطوریکه $\beta[0,1]$ یک بردار تصادفی از یک توزیع یکنواخت بدست آمده است.

همانطور که ذکر شد می توان از طول موج یا فرکانس نیز برای پیاده سازی استفاده کرد. در اینجا $F_{min}=0$و $F_{max}=100$ در نظر گرفته شده است که بسته به اندازه ی مساله علاقه مندی دارد.

در ابتدا به هر خفاش فرکانسی تصادفی که به صورت یکنواخت از $[F_{min} , F_{max}]$ بدست آمده است نسبت داده می‌شود. به همین دلیل می توان گفت که الگوریتم خفاش یک الگوریتم میزان ساز فرکانسی است وبطوریکه ترکیبی متعادل ترکیبی از اکتشاف و استخراج را فراهم می کند. بلندی و نرخ انتشار پالس یک مکانیسم اتوماتیک برای کنترل و بزرگنمایی خودکار ناحیه را فراهم می کند که منجر به را حل های مقرر می‌شود.

روش تحقیق

در این تحقیق برای آشنایی با موضوع مورد بحث ابتدا به بررسی مقالات علمی و منابع موجود در زمینه ی گرید پرداخته و پس از آشنایی کامل با این موضوع، زمینه ی قابلیت اطمینان در گرید را انتخاب و سپس از طریق کتابها و مقالات معتبر در این زمینه اطلاعات خود را تکمیل کرده ودرنهایت، قابلیت اطمینان با استفاده از الگوریتم خفاش به همراه اتاماتای یادگیر را جهت زمینه کاری انتخاب می نماییم.

چالش های موجود در عملکرد گرید

از دید کاربران در استفاده از گرید، سرعت بالای انجام محاسبه، اطمینان‌پذیری و هزینه هر چه پایین‌تر، دارای اهمیت است و از طرف دیگر از دید صاحبان منابعی که در محیط گرید به اشتراک گذاشته می‌شوند، کارایی منابع، سود بیشتر و انجام کارهای شخصی در زمان به اشتراک‌گذاری منابع دارای اهمیت است. از آنجا که این دو موضوع که دارای مغایرت‌هایی نیز می‌باشند، سبب به وجود آمدن مسائلی از جمله زمان‌بندی کارها[۲]، مدیریت منابع، تکثیر[۳] و تخصیص صحیح منابع گرید، تقسیم صحیح بار محاسباتی[۴]، مدیریت اطلاعات و... است. که این مسئله با توجه به تکثیر فایل ها و نسخه ها به دلیل دردسترس بودن بیشتر، قابل حل می باشد. در ادامه مسئله تکثیر داده در گرید را توضیح خواهیم داد.

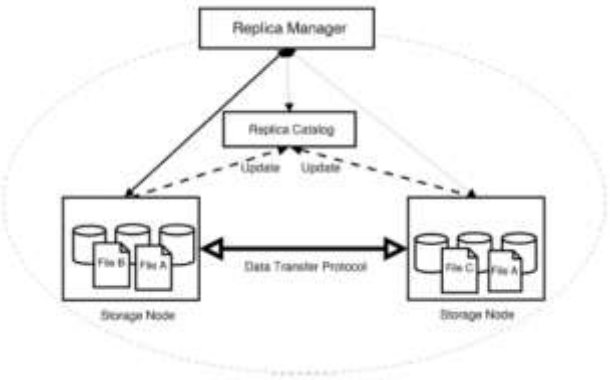
[2] Job scheduling
[3] Replication
[4] Load Balancing

مسله تکثیر در گرید داده

جهت به اشتراک گذاشتن داده‌ها در نقاط مختلف جغرافیایی گرید داده می تواند کمک بسزایی کند. گرید داده سرویس هایی را فراهم می کند جهت به اشتراک گذاشتن و مدیریت فایل های بسیار بزرگ در سراسر دنیا. کاهش ترافیک شبکه اساسی ترین کاری است که گرید انجام می‌دهد و در واقع کاهش ترافیک شبکه تکرار نسخه‌های گوناگون در سایت است که توسط کاربر خونده می‌شود (حری ۱۳۸۸)

جهت دسترسی سریع به فایلها می توان نسخه‌های متعددی در سایت های مختلف پخش شود. در گرید کاربران بر اساس کاتالوگ موجود نیازهای خود را جستجو می کنند و در صورتی که در سایت محلی از نسخه آن فایل وجود نداشته باشد درخواست آن را از یک سایت دیگر می کنند. نقش بسزای تکثیر داده با توجه به افزایش داده‌ها در سازمآن‌های مختلف و درخواست های متعدد اشتراک گذاری داده‌ها در این زمان مشخص می‌شود. (راشد رحمان ۲۰۰۸)

در شکل زیر ارتباطات کاتالوگ داده در گرید را نشان میدهد. در مسئله تکثیر داده عضوی به نام نسخه مدیریت همه نسخه ها را به عهده می‌گیرد. اطلاعات تمامی نسخه‌های تکثیرشده در کاتالوگ وجود دارد. با ایجاد نسخه جدید، کاتالوگ توسط مدیر نسخه‌ها بروز می‌شود.



سرویس های گرید

گروه جهانی Grid با ارائه معماری باز سرویس‌های Grid یا (Open Grid Services Architecture (OGSA و جمعآوری استانداردهای باز، مانند زبان تعریف سرویس‌های وب یا (Web Services Description Language (WSDLتوانسته است استانداردی آسان و در عین حال دقیق برای سیستم‌های Grid تعریف کند. از آن گذشته، OGSA از تجربیات به دست آمده از پروژه‌های بزرگی مانند Globus نیز بهره‌مند است .

استانداردهای باز و پروتکل‌های این معماری راه تولید سرویس‌ها را نشان می‌دهند. این سرویس‌ها قلب Grid هستند و در واقع به استفاده‌کننده اجازه می‌دهند با Grid کار کند. این سرویس‌ها عبارتند از :

-سرویس درخواست‌های پردازنده مرکزی

-سرویس مدیریت کنترل کار سیستم وsession ها

-سرویس جست‌وجوی اطلاعات

-سرویس تعیین پهنای باند شبکه

-سرویس مدیریت اطلاعات

دوره ۸، شماره ۱، بهار ۱۴۰۱، صفحات ۲۲۰-۲۰۳

وقتی متخصصان Grid درباره شروع شدن یک سرویس صحبت می‌کنند، مثلاً شروع شدن سرویس جست‌وجوی اطلاعات، منظور یک نمونه یا Instance سرویس است که می‌تواند تکالیف بلند مدت یا موقتی داشته باشد. این سرویس‌ها می‌توانند به‌صورت فعال یا غیر فعال باشند و زمان فعالیت را می‌توان با زمانبند یا به‌صورت اختیاری تعیین نمود .سرویسی خوب است که بتواند به‌راحتی امکانات خود را در اختیار استفاده کننده قرار دهد. مثلاً وقتی یک وسیله الکترونیکی را به پریز برق متصل می‌کنید، برای شما هیچ اهمیتی ندارد که برق مورد نیازتان از کجا می‌آید؛ فقط می‌خواهید از برق استفاده کنید.

سرویس خوب Grid نیز سرویسی است که بتواند سرویس موردنظر استفاده‌کننده را به‌راحتی دراختیار او قرار دهد و استفاده‌کننده بتواند به سادگی از آن استفاده کند. مثلا سرویس بانک‌اطلاعاتی در Grid باید به‌صورتی عمل کند که استفاده کننده فقط یک جست‌وجو وارد کند و جواب جست‌وجوی خود را بگیرد؛ بدون این‌که از جایگاه و عملیات بانک‌اطلاعاتی خبر داشته باشد.

انواع گرید از نظر اندازه و شکل

گرید ها شبیه به هم نیستند و اندازه و شکل های متفاوتی دارند. در زیر ۳ تقسیم بندی یا کلاس Grid های فعلی را که رایانه‌های فردی را تبدیل به یک مزرعه ابر رایانه با هزاران پردازنده را کرده است فهرست شده است:

* خوشه ای: ساده ترین توع این سیستم هستند. از مجموعه ای از رایانه ها که با هم کار می کنند تشکیل شده اند و یک نقطه دسترسی یه سیستم برای کاربران یک پروژه یا بخش سازمانی خاص ایجاد می کند.

*پردیزه: به پروژه ها و بخش های سازمانی متعدد اجازه می‌دهد که منابع رایانه ای خود را با هم به اشتراک گذارند. سازمانها می توانند از این توع سیستم برای مدیریت کارهای زیاد، ازفرآیند های تجاری سازمان تا داده کاوی استفاده کنند.

*گرید جهانی: مجموعه ای از چندینGrid پردیزه می باشند که پا بر فرآی مرزهای سازمانی گذاشته اند ویک سیستم مجازی بسیار بزرگ را ایجاد کرده اند. کاربران می توانند به منابع رانه ای بیش از آنچه که در سازمان دارند دسترسی داشته باشند.

پیشینه تحقیق

نوایی زاده و همکاران (۱۳۹۲) یک روش جدید جهت جایگذاری بهینه نسخه‌های ایجاد شده در گرید داده با استفاده از الگوریتم کرم شب تاب ارائه داده اند. الگوریتم کرم شب‌تاب از رفتارهای اجتماعی کرم های شب‌تاب در طبیعت الهام گرفته شده است. کرم های شب تاب نورهای ریتمیک و کوتاه تولید می‌کنند. کرم‌های شب‌تاب از این نورها برای دومنظور استفاده می کنند۱- پروسه جذب جفت‌ها۲- برای جذب شکار. به علاوه این نورها می‌توانند به عنوان یک مکانیزم محافظتی برای کرمهای شب تاب باشند. نورهای ریتمیک، نرخ تابیدن نور میزان فاصله زمانی که بین سیگنال های نور وجود دارند باعث می‌شوند دو جنس به یکدیگرجذب شوند. همچنین شدت نور در فاصله معین از منبع نور، از قانون مجذور معکوس پیروی می‌کند. گفته می‌شود شدت نور I با افزایش فاصله r کاهش پیدا می‌کند که اصطلاحا آن را با $I \propto {}^{1}/_{r^2}$ نشان می‌دهندبه علاوه جذب نور هوا با افزایش فاصله ضعیف و ضعیف‌تر می‌شود. نور می‌تواند به عنوان تابع هدفی که قرار است بهینه شود فرموله شود (حسن زاده و میبدی، ۱۳۹۰).

یزدی و همکاران (۱۳۹۰)، قابلیت اطمینان در گرید با استفاده از الگوریتم CRO[5] را مورد بررسی قرار دادند .در یک فرایند واکنش شیمیایی، دنباله ای از برخوردها در میان مولکول ها رخ می‌دهد. در این الگوریتم با ایجاد یک جمعیت اولیه شروع به کار می کند که هریک از مولمول ها در آن یک راه حل برای مسئله مکان یابی می باشند انتخاب نوع برخورد با ایجاد یک عدد تصادفی به نام b در محدوده [1,0]بستگی دارد، اگر Molecoll>b باشد برخورد تک مولکولی است، در غیر اینصورت برخورد بین مولکولی رخ خواهد داد. برای برخورد تک مولکولی، یک مولکول از جمعیت اولیه، بصورت تصادفی انتخاب شده و سپس با توجه به معیار تجزیه، تصمیم گرفته می‌شود برخورد بی اثر به دیوار یا تجزیه صورت گیرد. بطور مشابه، برای برخورد بین مولکولی، دو مولکولی از جمعیت انتخاب می‌شوند، و با بررسی کردن معیار ترکیب، مشخص می‌شود که برخورد از نوع برخورد بی اثر بین مولکولی و یا ترکیب است. برای بازیابی کردن شایستگی هر مولکول از رابطه (۳-۸) استفاده می‌شود. هر چه هزینه نهایی مولکول یا TC (X)کمتر باشد، شایستگس مولکول بالاتر خواهد بود.

تو و همکاران[6] (۲۰۰۷) با استفاده از الگوریتم ژنتیک یک روش جدید ایستا در زمینه جایگذاری داده‌ها در سیستم های P2Pدر گرید داده ارائه داده اند .این الگوریتم با هدف افزایش دسترس پذیری داده‌ها در گرید ارائه شده است .

منصوری و همکاران[7] (۲۰۱۴) یک روش تکثیر داده پویا به نام تکثیر سلسله مراتبی پویای اصلاح شده MDHR[8] پیشنهاد دادند. این استراتژی یک نسخه بهبود یافته از DHR[9]می باشد. MDHR نسخه ها را به جای استفاده از روشLRU[10] ، بر اساس آخرین زمانی که نسخه درخواست شده است، تعداد دسترسی ها و سایز نسخه، جایگزین می کند.

صلاح و همکاران[11] (۲۰۱۵)، یک الگوریتم تکثیر پویای چهار فازی به نام 4PDRA[12] پیشنهاد دادند که شامل فازهای زیر است: ۱ - ارزیابی و شناسایی داده محبوب و راه اندازی یک عملیات تکثیر زمانی که محبوبیت داده از یک حد آستانه عبور می کند. ۲-تحلیل و مدلسازی رابطه بین دسترس پذیری سیستم و تعداد نسخه ها و محاسبه یک تعداد مناسب از نسخه‌های جدید ۳- ارزیابی و شناسایی داده محبوب در هر سایت و قرار دادن نسخه ها در بین آنها ۴-حذف فایل ها با حداقل هزینه میانگین زمان دسترسی هنگام مواجهه با فضای ناکافی برای تکثیر.

تغییرات بلندی و نرخ انتشار پالس

به‌منظور فراهم کردن مکانیزمی موثر برای کنترل اکتشاف و استخراج و سوییچ به مرحله ی استخراج در مواقع ضروری مجبور به تغییرات بلندی A_i و نرخ انتشار R_i در حین تکرار هستیم. از انجاییکه با یافتن طعمه توسط خفاش بلندی ونرخ انتشار پالس افزایش می باد می توان به بلندی A_{min} و A_{max} را نسبت داد بطوریکه اگر $A_{min}=0$ باشد به این معنی است که خفاش طعمه خود را یافته و موقتا انتشار هر گونه صدایی را متوقف کرده است.

$$A_i^{t+1}=\alpha A_i^t \; R_i^{t+1}=R_i^0[1-\exp(-t)]$$

بطوریکه α و ¥ مقادیر ثابت هستند. عملکرد α مشابه عملکرد فاکتور کاهش دما در الگوریتم تبرید شبیه سازی شده است. برای

[5] PSO
[6] Tu et al
[7] Mansouri et al
[8] Modified Dynamic Hierarchical Replication
[9] Dynamic Hierarchical Replication
[10] Least recently used
[11] Salah
[12] Four-Phase Data Replication Algorithm

دوره ۸، شماره ۱، بهار ۱۴۰۱، صفحات ۲۲۰-۲۰۳

$$¥ > 0 \quad و \quad 0 < \alpha < 1$$

داریم

$$A_i \; 0 \; R_i^t \; R_i^0 \; as \; t$$

در ساده ترین مورد می توان $¥ = \alpha$در نظر گرفت. شبیه سازی انجام شده در این الگوریتم $a = ¥ = 0.9 \; to \; 0.98$ در نظر گرفته شده است.

با توجه به اهمیت و تاثیر مسئله تکرار داده و جایگذاری داده‌ها بر افزایش دسترس پذیری و قابلیت اطمینان در گرید. درنهایت با استفاده ازشبیه سازی نشان می دهیم الگوریتم پیشنهادی در مقایسه با الگوریتم هایز CROوPSO نتایج بهتری را موجب می‌شود.

پارامترهای الگوریتم

فرض کنید محیط گریدی با عناصر و پارامترهای زیر وجود دارد:

M :سایت با عناصر محاسباتی و ذخیره‌سازی مجزا در یک شبکه انتخابی با توپولوژی P2P

S_i: میزان ظرفیت ذخیره‌سازی سایت$i \; (1 <= i <= M)$

$C \; (i, j)$:هزینه ارتباط بین سایت i به سایتj در صورتی که بین این دو سایت لینک مستقیم باشد. در غیر این صورت هزینه ارتباط دو سایت برابر کمینه مجموع هزینه‌های ارتباط غیرمستقیم دو سایت است) در نظر داشته باشیم که $C(i, j) = C \; (j, i).$

N :فایل با نام‌های $O_1, O_2, \ldots, O_N$ و با حجم‌های $O_1, O_2, \ldots, O_K$ و $(1 <= K <= N)$

r_{ik}: تعداد درخواست‌های سایت i برای خواندن فایلk می‌باشند.

W_{ik}: تعداد درخواست‌های سایت i برای نوشتن بر روی فایلk می‌باشند.

P_k : فرض شده از هر فایل O_k در سیستم یک نسخه اصلی و اولیه وجود دارد. مکان نسخه اصلی فایل به هیچ‌وجه قابل‌تغییر و قابل حذف شدن نمی‌باشند. نسخه اصلی فایل O_k با P_k نشان داده می‌شود.

RS_k: در هر سایت شامل نسخه اصلی فایل، به ازای O_kاطلاعات آن فایل نگهداری می‌شود و با RS_k نشان داده می‌شود. (برای مثال لیست سایت‌هایی که فایل، O_k در آن تکرار شده است).

$RS = \{RS_1, RS_2, \ldots, RS_n\}$ شامل تمام اطلاعات تکثیر مربوط به فایل‌ها است.

به‌منظور تسهیل عمل سازگاری بین نسخه‌ها، در صورت تغییر در نسخهO_k، تغییرات به P_k ارسال می‌شود و سپسP_k تغییرات جدید را در مکان‌های مربوطه منتشر می‌کند.

S_{ik} : نزدیک‌ترین سایت (از نظر هزینه) حاوی نسخه O_kبه سایت i

اگر نزدیک‌ترین سایت شامل O_k، P_k باشد آنگاه می‌توان نتیجه گرفت: $P_k \; S_{ik} =$

اگر فایل درخواستی در خود سایتi وجود داشته باشد می‌توان گفت: $i \; S_{ik} =$

شرح روش پیشنهادی

فرض می‌کنیم محیط گرید را به صورت ماتریسی باینری با $M \times N$ بعد نشان می‌دهیم که در آن

$X_{ik} = 1$ اگر نسخه‌ای از فایل O_k در سایت i ذخیره شده باشد.

$X_{ik} = 0$ اگر نسخه‌ای از فایل O_k در سایت i ذخیره نشده باشد.

به‌منظور یافتن روشی مناسب و بهینه از نظر هزینه‌ی ارتباط و همچنین به‌منظور ارزیابی یک روش، محاسبه تابع هزینه از اهمیت زیادی برخوردار است. هدف نهایی ما رسیدن به کمترین هزینه نهایی در این ماتریس است. به این معنا که نسخه‌ها به نحوی در سایت‌ها تکرار شوند که کمترین هزینه برای دستیابی به تمامی آن‌ها را داشته باشیم. این هدف در معادله (۱) به زبان ریاضی بیان شده است.

$$\text{Minimize } Tc(X) \qquad (۱)$$

دو محدودیت در ایجاد ماتریس X وجود دارد:

به ازای هر سایت شامل نسخه اصلی در ماتریس X خانه‌ی مربوطه می‌بایست یک باشد.

$$X_{p_k k} = 1 \text{ for all } 1 \leq K \leq N \qquad (۲)$$

مجموع حجم فایل‌ها و نسخه‌های موجود در سایت i نمی‌بایست از ظرفیت سایت i بیشتر شود.

$$\sum_{k=1}^{N} X_{ik} O_k \leq s_i \text{ for all } 1 \leq i \leq M \qquad (۳)$$

با توجه به موارد ذکرشده هزینه خواندن و نوشتن با توجه به RS هر یک، به صورت زیر است (تو و همکاران، ۲۰۰۷):

$$R(RS) = \sum_{i=1}^{M}\sum_{K=1}^{N} R_{ik} = \sum_{i=1}^{M}\sum_{K=1}^{N} r_{ik} O_K C(i, S_{ik}) \qquad (۴)$$

$$W(RS) = \sum_{i=1}^{M}\sum_{K=1}^{N} W_{ik} = \sum_{i=1}^{M}\sum_{K=1}^{N} w_{ik} O_K \left[C(i, P_k) + \sum_{\substack{\forall j \in RS_k \\ j \neq i}} C(P_k, j) \right] \qquad (۵)$$

معادله (۴) هزینه‌های نهایی خواندن تمامی فایل‌ها را در این گرید فرضی نشان می‌دهد. ذکر این نکته ضروری است که تقاضاهای خواندنی که سایت S_{ik} آن را پاسخ می‌دهد کمترین هزینه را دارد. فرض ما بر این است که به هر تقاضا توسط نزدیک‌ترین نود از نظر هزینه پاسخ داده می‌شود.

هزینه نوشتن شامل دو بخش است که در معادله (۵) نشان داده شده است. قسمت اول، هزینه نوشتن در P_k و دیگری هزینه انتشار از P_k به تمام سایت‌های حاوی نسخه‌های مورد نظر (O_k) است.

هزینه کل شامل تمامی هزینه‌های خواندن و نوشتن می‌شود (معادله ۳-۶)

$$TC(RS) = R(RS) + W(RS) \qquad (۶)$$

با جایگذاری معادلات (۳-۴) و (۳-۵) در معادله (۳-۶) می‌توان گفت:

$$TC(RS)=\sum_{i=1}^{M}\sum_{K=1}^{N} r_{ik}O_k C(i,S_{ik})+\sum_{i=1}^{M}\sum_{K=1}^{N} W_{ik}O_k\left[C(i,P_k)+\sum_{\substack{\forall j\in RS_k\\ j\neq 1}} c(P_k,j)\right] \quad (۷)$$

$$TC(X)=\sum_{i=1}^{M}\sum_{K=1}^{N}\Bigg\{(1-X_{ik})[r_{ik}o_k \min\{C(i,j)|X_{ik}=1\}+w_{ik}o_k C(i,P_k) + x_{ik}\left(\sum_{x=1}^{M} w_{xk}\right)o_k C(i,P_k)\Bigg\} \quad (۸)$$

طبق معادله (۸) هنگامی که فایل درخواستی در خود سایت متقاضی باشد هزینه خواندن صفر است. همچنین هزینه نوشتن برای هر فایل O_k شامل هزینه نوشتن در سایت P_k بعلاوه هزینه انتشار نوشتن در سایت‌های حاوی همان داده است.

کد الگوریتم جایگذاری:

در یک فرایند الگوریتم خفاش، تعدادی از خفاش‌ها به صورت پراکنده قرار میگیرند. آنها برای رسیدن به هدف دو مورد را در نظر میگیرند. آنها جهت رسیدن به هدف خود پس از پیدا کردن موقعیت بهتر حتما باید پالس بهترین خفاش را دریافت کنند منظور از بهترین خفاش یعنی خفاشی که در نزدیکترین مکان به هدف قرار دارد. در این الگوریتم حرکت به سوی بهنرین خفاش از طریق این دو مرحله انجام می‌شود.

۱-یافتن موقعیت بهتر

۲- دریافت پالس از بهترین خفاش

. در این الگوریتم هر خفاش نشان دهنده یک راه حل می باشد. هرچه هزینه نهایی خفاش‌ها یا $TC(X)$ کمتر باشد شایستگی خفاش بالاتر خواهد بود.

در این مقاله جهت بهینه شدن پاسخ ها الگوریتم خفاش با با اتاماتای یادگیر ترکیب کرده ایم. همان طور که در ابتدای تعریف مسئله در فرمول (۳-۲) و (۳-۳) آورده شد ۲ محدودیت در ماتریس X بایستی رعایت شود. با توجه به اینکه جمعیت اولیه همواره صورت تصادفی ایجاد می‌شود، به‌منظور تضمین اینکه هرمولکول یک راه حل مناسب و قابل‌استفاده است، تابعی به نام **adjustment** در هر مولکول اجرا خواهد شد. در این تابع ابتدا تضمین می‌نماید که فایل k در سایت اصلی خود قرار داشته باشد. سپس برای هر سایت i کنترل می‌کند که آیا در هر سایت مجموع فایل‌های ذخیره‌شده از ظرفیت سایت i بیشتر است یا خیر. در صورت مثبت بودن باید فایل‌هایی را از سایت i حذف کرد. اما سؤال اینجاست که چه فایل‌هایی بایستی حذف شوند؟برای پاسخ دادن به این پرسش باید بررسی کرد که حذف هر فایل چه تأثیری بر ($TC(X)$) خواهد گذاشت. برای این منظور پارامتر جدیدی به نام $\overline{S_{ik}}$ معرفی می‌کنیم که نشان‌دهنده دومین نزدیک‌ترین سایت (از نظر هزینه) حاوی نسخه O_k به سایت i است. برای بررسی تأثیر حذف فایل k از سایت i با توجه به فرمول هزینه (۳-۸) در ۳ قسمت زیر باید بررسی شود.

کاهش هزینه نوشتن از P_k به سایت i به ازای تمام نوشتن ها

دوره ۸، شماره ۱، بهار ۱۴۰۱، صفحات ۲۲۰-۲۰۳

$$T_1 = C\,(P_K, i)\; o_k \sum_{j=1}^{M} W_{jk} \qquad (۹)$$

افزایش هزینه ناشی از خواندن فایل k از سایت i به دومین نزدیک‌ترین سایت شامل فایل k

$$T_2 = r_{ik} o_k C(i, \bar{S}_{ik}) \qquad (۱۰)$$

سایت j که سایت i نزدیک‌ترین سایت حاوی فایل k به آن است $(S_{jk} = i)$ اکنون برای دستیابی به فایل k به بایستی به $\bar{S}_{ik}$ مراجعه کنند. تفاوت این هزینه دستیابی با فرمول (۳-۱۶) محاسبه می‌گردد. مقادیر C_1 در معادله (۳-۱۱) و C_2 در معادله (۳-۱۲) محاسبه می‌گردد لازم به ذکر است که مجموعه U عبارت است از

$$U = \{j | S_{jk} = i\}.$$

$$C_1 = o_k \sum_{u \in U} r_{uk} C\,(u, i) \qquad (۱۱)$$

$$C_2 = o_k \sum_{u \in U} r_{uk} C(u, \bar{S}_{uk}) \qquad (۱۲)$$

$$T3 = C1 - C2 \qquad (۱۳)$$

بنابراین تفاوت هزینه عبارت است از:

$$\Delta C_{ik} = T_2 + T_3 - T_1 \qquad (۱۴)$$

با محاسبه ΔC_{ik} می‌توان متوجه شد که با حذف هر فایل هزینه چه تغییری خواهد کرد پس ابتدا آن فایل‌هایی را حذف می‌کنیم که کمترین افزایش هزینه را دارند.

همان طور که قبلاً ذکر شد ایجاد جمعیتی به صورت تصادفی قادر به پوشش محدودیت‌های مسئله نیست، از این رو تابع Adjustment بر روی هر جمعیت ایجادشده اعمال می‌شود تا هر دو محدودیت‌های مسئله پوشش داده شود. این تابع به این صورت عمل می‌کند:

در ماتریس X به ازای هر فایل، P_K آن بدست می آید و درایه $X\,(P_K, k)$ برابر یک قرار می‌گیرد با این عمل محدودیت اول تضمین می‌شود. به‌منظور اعمال محدودیت دوم به ازای هر سایت i، سطر i ام ماتریس X استخراج شده ودر ok متناظرش ضرب شده و مجموع آن محاسبه می‌شود که بیانگر مجموع حجم فایل موجود بر روی سایت i می باشد.

اگر این حجم از ظرفیت سایت i بیشتر باشد ΔC طبق رابطه (۳-۱۴) محاسبه شده و سپس ابتدا فایل هایی با کمترین افزایش هزینه حذف می‌شوند. در واقع برای اعمال محدودیت دوم هر سطر ماتریس X استخراج می‌شود و پس از اصلاح دوباره به جای خود بازمی گردد و سطر بعدی ماتریس X برداشته می‌شود و به همین ترتیب ادامه می‌یابد تا کل ماتریس محدودیت دوم را تحت پوشش قرار دهد. شکل ۳-۱ و شکل ۳-۲ به ترتیب شبه کد الگوریتم پیشنهادی و شبه کد اعمال محدودیت ها را نشان می‌دهد.

دوره ۸، شماره ۱، بهار ۱۴۰۱، صفحات ۲۲۰-۲۰۳

1-For i=1 to popsize
2-Randomly generate molecule struct X
3- X=Adjustment(X)
4-$PE \leftarrow f(X), KE \leftarrow InitialKE,\ NumHit \leftarrow 0, MinStruct \leftarrow X, MinPE \leftarrow 0, MinHit \leftarrow 0$
5-End //End of initialization
6-while the stopping criteria not met do
7- Generate $b \in [0,1]$
8= if $b > MoleColl$ then
9- Randomly select one molecule with struct X
10- if Decomposition criterion (24-2) met then
11- create two molecule and Obtain their molecule struct X_1 and X_2 from X
12- $X_1 = Adjustment(X_1)\ , X_2 = Adjustment(X_2)$
13- $PE_{X_1} \leftarrow f(X_1), PE_{X_2} \leftarrow f(X_2)$ //calculate fitness with (9-3)
14- Calculate KE,buffer,MinHit,MinStruct,MinPE of X_1 and X_2 with Decomposition
15- Else
16- Create one molecule and Obtain its molecule struct X_1 from X
17 – $X_1 = Adjustment(X_1)$
18- $PE_{X_1} \leftarrow f(X_1)$ //calculate fitness with (9-3)
19- Calculate KE,buffer,MinHit,MinStruct,MinPE of X_1 with OnWallineffective
20- End
21- Else
22- Randomly select two molecules with their struct X_1 and X_2
23- if Synthesis criterion (25-2) met then
24- Obtain one molecule struct X' from X_1 from
25- $X' = Adjustment(X')$
26- $PE_{X'} \leftarrow f(X')$ //calculate fitness with (9-3)
27- Calculate KE,MinHit,MinStruct,MinPE of X' with Synthesis
28- Else
29- Create two molecule and Obtain their molecule structs X_1' and X_2' from X_1 from X_2
30- $X_1' = Adjustment(X_1')$
31- $X_2' = Adjustment(X_2')$
32- $PE_{X_1'} \leftarrow f(X_1')$ //calculate fitness with (9-3)
33- $PE_{X_2'} \leftarrow f(X_2')$ //calculate fitness with (9-3)
34- Calculate KE,MinHit,MinStruct,MinPE of X_1' and X_2' with interMolecular
35- End
36- end
37-check for any new minimum solution
38-end while
39-Output the best solution found and its objective function value

شکل ۱ الگوریتم پیشنهادی جهت جایگذاری فایل ها در گرید

دوره ۸، شماره ۱، بهار ۱۴۰۱، صفحات ۲۲۰-۲۰۳

```
1-For each file k
2-    i = pk_k
3-    X(i,k) = 1
4-End                                   //first constraint
5-For each site i
6-    a = X(i,~)                        //Obtain Row ith of Matrix X and put it in array a
7-  b = a.* ok
8-  If sum(b) > si(i)
9-     Create an empty array ΔC with Demension of  [1,nk]
10-       For each file k
11-  Calculate ΔC(K)
12-      End
13-  End
14-  Sort (ΔC) in descending
15-  Obtain indexes of Matrix ΔC  and put them in j
16-  For each j
17-    If a(j)=1 && pk(j) ≠ i
18-        a(j)=0
19-       b = a.* ok
20-       if sum(b) <= si(i)
21-break
22-         end
23-    end
24-end
25-X(i,~) = a                          //Return Matrix a to Row ith of Matrix X
26-End                                  //secound constraint
```

شکل ۲ شبه کد اعمال محدودیت ها

الگوریتم های مورد مقایسه

دو روشی که ما کار خود را با آن‌ها مورد مقایسه قرار داده ایم الگوریتم خفاش و PSO می باشند. هر یک از این روش‌ها مزایای و معایب خاص خود را دارند. هدف ما ارائه‌ی راه‌حلی بهینه‌تر و کارآمدتر در مقایسه با دیگر الگوریتم‌های ایستا است. از آنجا که در این تحقیق بر بهبود همزمان دسترسی به داده و قابلیت اطمینان آن تمرکز می‌کنیم و با توجه به اینکه الگوریتم‌های ایستا در جهت افزایش قابلیت اطمینان بسیار کارا می‌باشند، با ارائه الگوریتمی ایستا ولی کارا، سعی در بهبود جایگذاری و انتخاب نسخه در گرید داریم.

شبیه‌سازی و ارزیابی روش پیشنهادی

یکی از روش‌های نشان دادن قابلیت‌های یک روش نسبت به روش‌های دیگر ارزیابی آن الگوریتم با دیگر الگوریتم‌ها است. این کار را با کمک شبیه‌سازی انجام می‌دهیم. به همین منظور الگوریتم خود را با الگوریتم‌های مشابه دیگری که در زمینه تکثیر داده استفاده‌شده‌اند مقایسه کرده‌ایم.

پارامترهای مسئله

هزینه بین نودها را عددی تصادفی بین ۱ تا ۱۵ در نظر گرفته‌ایم. حجم فایل‌ها بین ۱۰ مگابایت تا ۷۰ مگابایت در نظر گرفته‌ایم. همچنین ظرفیت عناصر ذخیره‌سازی بین ۱۵ تا ۸۵ درصد مجموع حجم فایل‌ها در نظر گرفته‌ایم .تعداد درخواست برای خواندن و نوشتن هر فایل توسط هر سایت یا همان نرخ خواندن و نرخ نوشتن را عددی تصادفی بین ۱ تا ۱۵ در نظر گرفتیم.

پارامترهای شبیه سازی

توپولوژی گرید استفاده شده در این شبیه سازی P2P می باشد. تعداد جمعیت برای هر سه الگوریتم ۱۰ و تعداد تکرار نیز ۱۰۰۰ می باشد.

جدول ۱-مقادیر در نظر گرفته شده برای پارامترهای الگوریتم خفاش

الگوریتم خفاش	KElossRate	InitialKE	∝	β	buffer	MoleColl
	۲ .۰	۱۰۰۰	۵۰۰	۱۰	۰	۲ .۰

در اولین بررسی ما تعداد سایت ها را ثابت فرض کرده ایم و با متغیر قرار دادن فایل ها نتیجه ها را ارزیابی کرده ایم.

جدول۲-مقایسه چهار الگوریتم براساس تغییر تعداد فایل ها

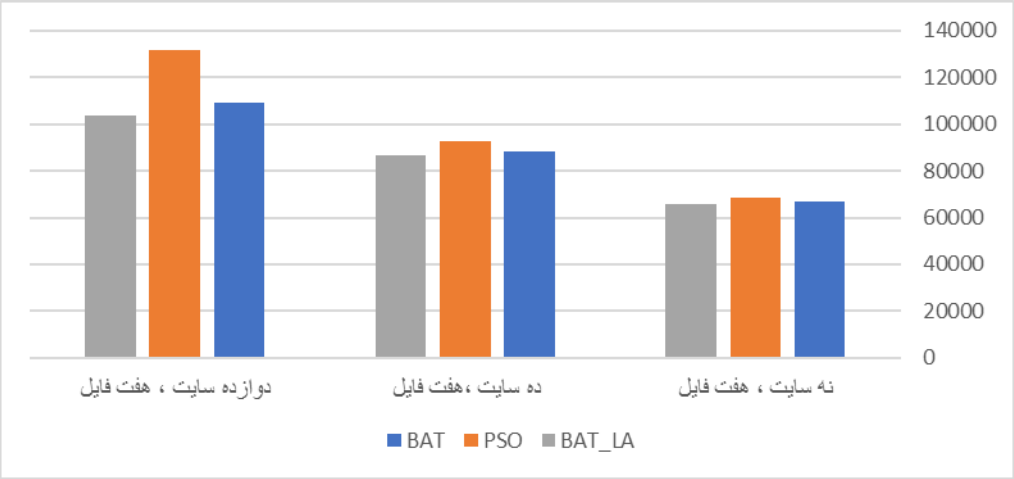
تعداد سایت ها و فایل ها	Bat	PSO	Bat_la
ده سایت و شش فایل	۶۶۰۸۹	۷۵۴۱۸	۶۴۷۴۶
ده سایت و هفت فایل	۸۸۴۹۲	۹۰۵۳۵	۷۹۸۷۵
ده سایت و نه فایل	۱۱۸۸۰۴	۱۲۱۸۶۰	۱۱۵۲۳۸

نمودار۱تاثیر افزایش تعداد فایل ها بر هزینه

در بررسی دوم تعداد فایل ها را ثابت در نظر می گیریم و تاثیر افزایش تعداد سایت ها یا عناصر ذخیره سازی را بر هزینه مورد ارزیابی قرار می دهیم.

جدول۲مقایسه سه الگوریتم بر اساس متغیر بودن تعداد سایت هاو ثابت بودن فایل ها

تعداد سایت ها و فایل ها	BAT	PSO	BAT_LA
نه سایت و هفت فایل	۶۶۹۹۲	۶۸۶۹۷	۶۵۵۶۱
ده سایت و هفت فایل	۸۸۴۹۲	۹۰۵۳۵	۷۹۸۷۵
دوازده سایت و هفت فایل	۱۰۹۰۱۱	۱۰۳۹۳۷	۱۱۲۰۳۸



نمودار۲-تاثیر افزایش سایت ها بر هزینه

بحث و نتیجه گیری

گرید یک سازوکار برای دسترسی به منابع محاسباتی (پردازنده‌ها، منابع ذخیره‌سازی، داده، برنامه و(...، بدون آگاهی داشتن از مکان این منابع، فناوری استفاده‌شده در این منابع، سخت‌افزار و سیستم‌عامل در سازمان‌های مختلف است. یکی از انواع مهم گرید، گریدهای داده هستند. روش اساسی برای کاهش ترافیک شبکه در گرید داده ایجاد نسخه از فایل‌ها در سایت‌های مختلف است، که به آن عمل تکثیر یاReplication می‌گویند. تکثیر داده در سیستم‌های توزیع‌شده روشی است تضمینی به‌منظور بهبود کارایی در این سیستم‌ها. البته برای این کار نیازمند مشخص کردن تعداد نسخه‌ها و مکان‌آن‌ها هستیم. در این تحقیق هدف ما پاسخ به این سؤال که در هنگام عمل تکثیر بهترین مکان برای کپی کردن نسخه‌های متعدد فایل‌ها کدام سایت‌ها می‌باشند که بهترین زمان پاسخ نهایی و یا به عبارتی سیستم گرید کمترین هزینه نهایی را داشته باشد. این تحقیق را با فرض وجود محدودیت در حافظه ذخیره‌سازی سایت‌ها انجام دادیم .در این تحقیق با استفاده از یک روش مبتنی بر هزینه، الگوریتم جدیدی را برای تکرار داده‌ها در گرید، معرفی کرده ایم. هرروزه تحقیقات زیادی در زمینه‌ی گرید داده و مسئله تکرار داده انجام می‌شود. هدف نهایی اکثر این تحقیقات بهبود زمان پاسخ و افزایش کارایی می‌باشند. در این پژوهش الگوریتمی

خفاش به همراه اتاماتای یادگیر در گرید ارائه شد. مقایسه این روش با روش‌های گذشته بهبود کارایی و افزایش دسترس‌پذیری این روش را نشان می‌دهد.

در اینجا می‌توان پیشنهادهای زیر را ارائه نمود :

- برای بهبود تکثیر داده و کاهش حجم تکثیر می‌توان فایل‌ها را به قسمت‌های کوچک‌تری تقسیم کرد؛ و به جای تکرار کل فایل فقط قسمت مشخصی از آن را تکثیر کرد .
- استفاده از الگوریتم های تکاملی دیگر در حل مکان یابی مناسب داده‌ها در گرید
- اسنفاده از تابع برازندگی جدیدتری در این راستا
- می توان علاوه بر دومین نزدیکترین، مرحله سومی از نزدیکترین‌ها را نیز در نظر گرفت.

منابع

۱. حسن زاده ط، میبدی م، ۱۳۹۰، ارائه یک الگوریتم کرم شب تاب بهبود یافته برای بهینه سازی در محیط های ایستا، پنجمی کنفرانس داده کاوی ایران، دانشگاه صنعتی امیرکبیر، تهران، ایران.

۲. حری، ع. ۱۳۸۸. ارائه روش‌های جدید تکرار داده در گرید داده با رویکرد سازگاری بین داده‌ها. پایان‌نامه کارشناسی ارشد، دانشگاه شیراز .

۳. رحیمی، غ. ۱۳۸۱. مروری بر مفاهیم سیستم‌های توزیع‌شده، گزارش .

۴. شجاعتمند، ا. ۱۳۹۰. ارائه روشی برای بهبود داده و قابلیت اطمینان آن در گرید با استفاده از الگوریتم PSO. پایان‌نامه کارشناسی ارشد، دانشگاه آزاد اسلامی واحد علوم و تحقیقات خوزستان.

۵. شیبانی م، میبدی م، ۱۳۸۵، PSO-LA یک مدل جدید برای بهینه‌سازی، دوازدهمین کنفرانس بین‌المللی انجمن رایانه، تهران، دانشگاه شهید بهشتی، ۳-۱ اسفند. ص ۱۸۶-۱۷۹.

6. Abawajy J. 2004. Placement of file replicas in data grid environments: Lecture Note in Compue Science (LNCS). Springer, Heidelberg. 3038: 66–73.

7. Ardalan A. 2006. The Replica Placement Problem In Multimedia CDNs: Use of Ant Clustering. Special Topics in Databases Course. ECE Dept. University of Tehran.

8. Bell W, Cameron G, Carvajal-Schiaffino R, Millar A. P, Stockinger K, Zini F. 2003. Evaluation of an Economy-Based File Replication Strategy for a Data Grid, Proceedings of Agent Based Cluster and Grid Computing at CCGrid, IEEE-CS Press. Tokyo, Japan, Nov 22–25, pp 126–132.

9. Chervenak A. 2002. Giggle: A framework for constructing scalable replica location services. Proceeding of the ACM/IEEE Super Computing Conference, Los Alamitos, USA, Nov 02-04, pp: 1-17.

10.]Ding Y. 2009. Automatic data placement and replication in grids. Proceeding of International IEEE conference on High Performance Computing (HIPC). Goa, India, Dec 20-22, pp: 30-39.

11. Kim D H, Kang K W. 2006. Design and implementation of integratedinformation system for monitoring resources in grid computing. Proceeding of 10th International Conference on Grid Computing. Nanjing, China, Oct 13-16, pp: 1-6.
12. Drezner Z, Hamacher H W. 2002. Facility Location Applicationand Theory, Springer, 54: 131-135.
13. Ferreira L, Berstis M, Armstrong J, Kendzierski M, Neukoetter A, Takagi M, Richard Bing-Wo, Adeeb A, Ryo M, Olegario H, James M, Norbert B. 2003. Introduction to Grid Computing with Globus. Journal of International Technical Support Organization IBM.
14. [22]Foster I, Kesselman C. 2002. Grid services for distributed system integration. Journal of IEEComputer, 35: 37-46.
15. [23]Foster I, Kesselman C, Kaufman M. 2004. The Grid Blueprint for a new computing Infrastructure. 1st edn, Massachusetts: Morgan Kaufmann Publishers, 667 p.
16. Foster I, Iamnachi A. 2003. On death, taxes, and the convergence of peer to-peer and Grid computing, Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS).
17. Foster I, Zhao Y, Raicu I, Lu S. 2008. Cloud computing and grid computing 360-degree compared, Proceedings of IEEE Grid Computing Environments Workshop. Austin, USA, Nov 12-16, pp 1-10.
18. Khanli M, Isazadeh A, Shishavanc N. 2011. PHFS: A dynamic replication method to decrease access latency in multi tier data grid. Future Genaration Computer System. Elsevier, 27: 233-244
19. Lam, A. Y. , & Li, V. O. (2010). Chemical-reaction-inspired metaheuristic for optimization. *Evolutionary Computation, IEEE Transactions on*, *14* (3) , 381-399.
20. Moallem A. 2009. Using swarm intelligence for distributed job scheduling on the grid. MSc thesis, University of Saskatchewan, Saskatoon, Canada.
21. Perez J, Garcia-Carballeira F, Carretero J, Calderon A, Fernandez J. 2010. Branch replication scheme: A new model for data replication in large scale data grids. Journal of Future Generation Computer Systems. Elsevier. 26: 12-20.
22. Rahman R M, Barker K, Alhajj R. 2005. Replica Placement in Data Grid: A Multi-objective Approach. Lecture Note in Computer Science (LNCS). Springer, Heidelberg. 3795: 645-656.
23. [43]Saleh, A. , Javidan, R. , & FatehiKhajeh, M. T. (2015). A four-phase data replication algorithm for data grid. Journal of Advanced Computer Science & Technology, 4 (1) , 163-174.
24. Sushant Goel. 2002, Replica managment in datagrid. leanne guy,peter kunszy, data replication strategies in wide strategies system.
25. Tang M, Lee B S, Tang X, Yeo C K. 2005. The impact of data replication on job scheduling. performance in the Data Grid School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, Singapore.
26. Yang M, Fei Z. 2003. A model for replica placement in content distribution networks for multimedia applications, Proceedings of 3rd IEEE International Conference on Communications, Lexington, Kentucky, USA, May 11-13, pp: 557-561.