



توازن بار در گریدهای رایانشی با استفاده از کلونی مورچه‌ها

سعید قنبری محمدرضا میدی

آزمایشگاه سیستم‌های نرم افزاری

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

تهران-ایران

Saeed_ghanbari@yahoo.com

چکیده

گریدهای رایانشی مدل جدیدی برای توسعه رایانشهای موازی سنگین می‌باشد که هدف آن ایجاد یک ابررایانه انتزاعی با استفاده از منابع آزاد در شبکه‌های گسترده برای بسترهای نامطمئن (چون اینترنت) می‌باشد. از آنجاییکه در گریدهای رایانشی بستر ارتباطی و موجودیت منابع بسیار پویا است لذا مدیریت منابع از جمله توازن بار از مسائل مهم می‌باشد. در این مقاله یک الگوریتم توازن بار مبتنی بر کلونی مورچه‌ها بنام امتیازدهی پیشنهاد شده‌است. در الگوریتم پیشنهادی، زمان جستجوی مورچه‌ها برای یافتن منابع مناسب کاهش یافته و در نتیجه توانایی سیستم گرید برای توازن بار افزایش می‌یابد. با انجام چند شبیه‌سازی، کارایی الگوریتم پیشنهادی نسبت به دو الگوریتم دیگر گزارش شده که هر دو مبتنی بر کلونی مورچه‌ها می‌باشند، نشان داده شده است.

کلمات کلیدی: کلونی مورچه‌ها، توازن بار، گرید رایانشی، نشانه‌گذاری، امتیازدهی

۱-مقدمه

و بطور خاص توازن بار در گریدهای رایانشی ارائه می‌شوند، باید

مقیاس‌پذیری^۲ و سازگاریپذیری^۳ باشند [۹]. از طرف دیگر مدیریت منابع بایستی بصورت غیر متمرکز صورت بگیرد زیرا که مدیریت متمرکز امکان توسعه‌پذیری گرید را کاهش می‌دهد.

در تعدادی از راه‌حلهای ارائه شده برای توازن بار در گریدهای رایانشی، از عاملها استفاده شده است. عاملها موجودیتهای مستقل و هدفمندی هستند که برای محیطهای توزیع شده و ناهمگون مناسب هستند. کلونی مورچه‌ها یک سیستم متشکل از تعداد زیادی عامل سبک وزن است که از آنها برای مسیر یابی در شبکه‌های ارتباطی و همچنین توازن بار در گریدهای رایانشی استفاده شده است [۱۰].

گریدهای رایانشی^۱ مجموعه‌ای از رایانه‌ها که به صورت جغرافیایی موزون شده اند هستند که با یکدیگر برنامه‌ها، داده‌ها و منابع رایانشی گرید را به اشتراک می‌گذارند. گریدهای رایانشی بعنوان یک مدل جدید رایانشی برای حل مسائل بزرگ و پیچیده در حوزه‌های مختلف مطرح شده اند.

وظیفه مدیریت منابع در گرید، توزیع کارها بصورت متوازن روی بستر یا حجم انبوه منابع ناهمگون و پویا می‌باشد. زمانی یک منبع را پویا می‌نامیم که آن منبع از حیث در دسترس بودن، کارایی، نوع و مکان در حال تغییر باشد. بنابراین راه‌حلهایی که برای مدیریت منابع

^۲ Scalability

^۳ Adaptability

^۱ Computing Grids

در این مقاله با استفاده از ایده‌های مطرح شده در کارهای [5]، [3]، [1] و [10]، الگوریتمی با نام امتیازدهی برای توازن بار در گریدهای رایانشی پیشنهاد می‌گردد و کارایی آن با الگوریتم نشانه گذاری و الگوریتم تمام تصادفی [2]، [1] مقایسه می‌گردد. مزیت الگوریتم پیشنهادی بر الگوریتم نشانه گذاری و تمام تصادفی آنستکه پس از به تعادل رسیدن سیستم، عاملها با جستجوی کمتری به گره‌های مادون بار^۴ شده هدایت می‌شوند و لذا توانایی توازن بار بیشتر می‌گردد.

ادامه مقاله بدین صورت سازماندهی شده است: بخش ۲ اختصاص به بررسی اجمالی روشهای توازن بار با استفاده از عاملها دارد. بخش ۳ به معرفی الگوریتمهای توازن بار گزارش شده می‌پردازد. در بخش ۴ الگوریتم امتیازدهی همراه با مدل انتزاعی سیستم گرید شرح داده خواهد شد. در بخش ۵ نتایج شبیه سازیها ارائه می‌گردد. بخش پایانی مقاله نتیجه گیری می‌باشد.

۲- توازن بار با استفاده از عاملها

اخیرا تلاشهای گسترده‌ای برای حل مسئله توازن بار در گریدهای رایانشی با استفاده از توانمندیهای عاملهای مستقل صورت گرفته است. ARMS^۵ یک سیستم سلسله مراتبی از عاملهای ناهمگون است [۸]، [۹]. در این سیستم هر عامل به کمک PACE toolkit [۷] یک انتزاع^۶ سازگار و توسعه پذیر از منبع رایانشی را ارائه میدهد. عاملها برای برقراری توازن بار، متکی بر اطلاعات درباره منابع در سایر عاملها می‌باشند که این اطلاعات از طریق تبادل پیام بین عاملها حاصل می‌شود. بنابراین در مقیاس بالا، برای مدیریت پویایی منابع، سربار زیادی به سیستم تحمیل می‌شود و سیستم را در برابر خرابی خطوط ارتباطی بسیار آسیب پذیر می‌سازد.

در [۶]، عاملها در نقش تقاضاها عمل می‌کنند. بدین صورت که برای هر تقاضا یک عامل متحرک^۷ ایجاد و شروع به پرسه زدن در شبکه می‌کند تا اینکه به منبع مورد نظر خود برسد. حیطه جستجو و هماهنگی عاملها، توسط دلالتها^۸ تعیین و صورت می‌گیرد. در این سیستم وابستگی به دلالتها می‌تواند باعث آسیب پذیری سیستم گردد. گونه دیگر سیستمهای چند عاملی برای توازن بار در گرید، سیستمهای سازگار پیچیده^۹ هستند. اینگونه سیستمها معمولا برای

¹⁰ Emergent

¹¹ Resnick

¹² Placement

¹³ Partial Objective Function

⁴ Underload

⁵ Agent-based Resource Management System

⁶ Abstraction

⁷ Mobile agents

⁸ Broker

⁹ Complex Adaptive System (CAS)

NodeID	Pheromone score	# ants
Apache-02	0.572	15
Apache-01	0.356	10

شکل ۴- جدول فرومون

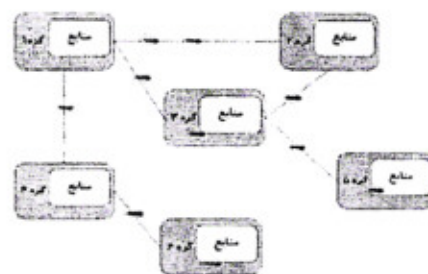
لینکهای ارتباطی^{۲۳}، ارتباط یک گره با سایر گره‌ها را فراهم می‌کند. رد و بدل مورچه‌ها، کارها، گزارش مورچه‌ها و نتایج رایانش کارها از طریق این لینکهای ارتباطی انجام می‌گیرد. وظیفه دیگر لینکهای ارتباطی شناسایی همسایگان و بهنگام کردن جدول همسایگی^{۲۴} (جدول گره‌های همسایه و میزان پهنای باند خط ارتباطی) می‌باشد.

در ادامه به شرح مفصلتری از چگونگی عملکرد الگوریتم امتیاز دهی می‌پردازیم.

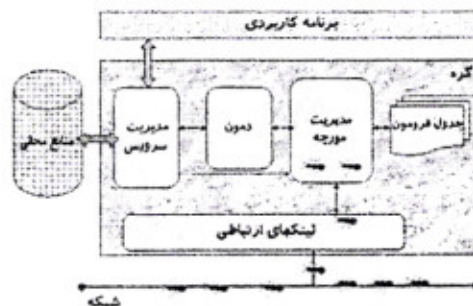
شرح الگوریتم پیشنهادی: هدف اصلی این الگوریتم کاهش

متوسط زمان کاوش مورچه‌ها برای یافتن گره‌های مادون بار شده می‌باشد. در این الگوریتم جدول فرومون هر گره فقط توسط دمون آن گره بهنگام می‌شود و بهنگام سازی بدین ترتیب است که به میزانی که به یک گره کار ارسال می‌شود به آن گره، در گره فرستنده امتیاز داده می‌شود که در اینجا امتیاز همان مقدار فرومون است. بنابراین هر گره (خصوصاً گره‌هایی که بار زیادی تولید می‌کنند) پس از گذشت مدتی، جدول فرومونشان بگونه‌ای پیکر بندی می‌شود که مورچه‌ها با احتمال بالایی به گره‌های مادون بار شده هدایت می‌شوند. در ادامه به شرح قسمتهای الگوریتم می‌پردازیم.

دمون: دمون وظیفه اش نظارت بر عملکرد مدیریت سرویس، ایجاد و ارسال مورچه و در نهایت زمانبندی کارها و ارسال آنها به گره‌های دیگر است. این قسمت دائماً عملکرد مدیریت سرویس را تحت نظر دارد و در صورت مافوق بار^{۲۵} شدن و یا گذشت مدت زمان معین از آخرین تولید و ارسال مورچه، اقدام به ایجاد تعدادی مورچه می‌نماید. یک گره در صورتی دارای وضعیت مافوق بار است که متوسط زمان پاسخگویی آن از میانگین زمان پاسخ سایر گره‌ها در گرید بیشتر باشد. تشخیص وضعیت مافوق بار کار چندان آسانی نیست چراکه اطلاع از وضعیت کلی گرید عملاً غیرممکن است. برای این منظور تخمینی از وضعیت کلی گرید مورد استفاده قرار می‌گیرد. بدین صورت که هر مورچه در طول سفرش میانگین زمان پاسخگویی گره‌هایی که ملاقات می‌کند را محاسبه و به دمون گره تولید کننده گزارش می‌کند. بهنگام سازی تخمین وضعیت کلی گرید بصورت زیر انجام می‌گیرد.



شکل ۲- نمای انتزاعی از یک شبکه گرید



شکل ۳- نمای انتزاعی یک گره

هر گره یک میان افزار^{۱۸} واسطه بین برنامه‌های کاربردی و شبکه می‌باشد و با یک یا چند برنامه کاربردی در تعامل است. هر برنامه کاربردی تقاضاهای خود را به مدیریت سرویس^{۱۹} که لایه ای انتزاعی روی منابع رایانشی محلی می‌باشد، ارائه می‌دهد. سپس مدیریت سرویس کارها را زمانبندی و اجرا می‌کند. دمون^{۲۰} بخشی از گره است که وظیفه اش نظارت بر عملکرد مدیریت سرویس، تولید مورچه و ارسال کارها به گره‌های مادون بار شده برحسب گزارشهای دریافتی از مورچه‌ها می‌باشد. وظیفه مدیریت مورچه^{۲۱} دادن خدمات و پاسخ گویی به مورچه‌های ملاقات کننده گره است. مورچه‌ها پس از تولید، وضعیت گره ایجاد کننده شان را ثبت و به پرسه زدن در شبکه می‌پردازند تا اینکه به یک گره مادون بار شده ای (نسبت به گره مبدأ) بر خورد کنند و یا اینکه عمرشان (TTL) به پایان برسد. مورچه‌ها با ارسال گزارش به دمون گره ایجاد کننده شان به کار خود پایان می‌دهند. جستجوی مورچه‌ها به کمک میزان فرومون مسیرها صورت می‌گیرد. هر گره دارای یک جدول فرومون^{۲۲} می‌باشد که میزان فرومون مسیرهای آن گره به سایر گره‌ها در آن ثبت می‌شود. شکل ۴ نمونه ای از یک جدول فرومون را نشان می‌دهد.

¹⁸ Middleware¹⁹ Service Manager²⁰ Demon²¹ Ant Manager²² Pheromone Table²³ Communication link²⁴ Neighborhood Table²⁵ Overload

$$\psi^i = \frac{\text{Total Jobs Received From Node}(i)}{\text{Total Jobs Received}} \quad (6)$$

مورچه: مورچه ای که توسط دمون ایجاد شده، شروع به گردش در شبکه گریز می کند. با رسیدن به یک گره (فرضا گره j)، مقدار AW_j که متوسط زمان انتظار یک کار برای اجرا در گره j می باشد را بصورت زیر محاسبه می کند.

$$AW_j = \max \left(AW_j, \frac{1}{CMC_{i,j}} \right) \quad (7)$$

که $CMC_{i,j}$ ظرفیت ارتباطی بین گره مورد ملاقات (j) و گره مبدأ (i) است. مورچه مقدار AW_j را با مقدار آستانه AW_0 که توسط دمون در گره ایجاد کننده اش تعیین شده مقایسه می کند. اگر مقدار AW_j از مقدار AW_0 کمتر باشد، آن گره را انتخاب می کند و سپس با ارسال آدرس گره، مقدار AW^{avg} ، مقدار WJ_R (تعداد کارهای منتظر در صف) و مقدار CC_R (قدرت پردازشی گره مورد ملاقات) به کار خود را خاتمه می دهد. اگر مقدار AW_j از مقدار AW_0 بیشتر باشد، در این صورت مورچه با احتمال

$$\lambda = 1 - e^{-\left(\frac{n_0}{2n}\right)}$$

گره ای که مورچه می تواند ملاقات کند) به جدول فرومون و با

$$\tau = e^{-\left(\frac{n_0}{2n}\right)}$$

احتمال τ به جدول همسایگی مراجعه می کند. اگر به جدول فرومون مراجعه کند، گره ای را با احتمالی متناسب با مقدار فرومونش انتخاب می کند و اگر به جدول همسایگی مراجعه کند، یکی از گره هایی را که ملاقات نکرده به صورت تصادفی (با احتمال برابر) انتخاب می کند و به آن وارد می شود. یک مورچه در طول سفرش بیش از یک بار از یک گره عبور نمی کند و اگر پس از ملاقات n_0 گره، به گره مناسب برخورد نکرده، با دادن گزارش به گره تولید کننده، به کار خود خاتمه می دهد.

در الگوریتم تمام تصادفی، مورچه از جدول فرومون برای انتخاب گره بعدی استفاده نمی کند بلکه فقط به جدول همسایگی مراجعه می نماید.

جدول فرومون: با استفاده از جدول فرومون، مورچه ها

بصورت تصادفی گره های بعدی خود را انتخاب می کنند. در روش تمام تصادفی جدول فرومون مورد استفاده قرار نمی گیرد. نقش جدول فرومون در دو روش امتیاز دهی و نشانه گذاری متفاوت است. در الگوریتم امتیاز دهی مورچه ها در طول سفرشان از جدول های فرومون برای یافتن مکان های مادون بار شده استفاده می کنند اما آنها را بهنگام نمی کنند بلکه بهنگام سازی توسط دمون، پس از دریافت گزارش از مورچه ها و ارسال کار انجام می شود. به هر گره به نسبت مقدار کاری که پذیرا بوده فرومون اضافه می شود. بدین صورت

$$AW_0^{new} = AW_0^{old} + \beta(AW_0^{avg} - AW_0^{old}) \quad 0 < \beta < 1 \quad (1)$$

که AW_0^{avg} میانگین زمان پاسخگویی محاسبه شده برای گره های ملاقات شده توسط مورچه و AW_0^{old} مقدار قبلی برای این تخمین می باشد.

پس از ارسال مورچه ها، دمون پس از سپری شدن یک مدت زمان معین و یا دریافت تمام گزارش ها، اقدام به زمان بندی می نماید. برای زمان بندی ابتدا گره های گزارش شده را بر اساس گزارش مورچه ها بصورت نزولی مرتب می کند. سپس از گره ای که بالاترین امتیاز را دارد شروع کرده و به هر گره X واحد کاری ارسال می کند. مقدار X به صورت زیر محاسبه می شود:

$$\frac{WJ_L - X}{CC_L} > \frac{X}{CMC} + \frac{X + WJ_R}{CC_R} \Rightarrow X = \frac{\frac{WJ_L - WJ_R}{CC_L - CC_R}}{\frac{1}{CC_L} + \frac{1}{CMC} + \frac{1}{CC_R}} \quad (2)$$

که WJ_L کارهای منتظر در گره محلی؛ WJ_R کارهای منتظر در گره دور؛ CC_L ؛ CC_R ؛ ظرفیت رایانشی گره دور (job/unit of time) و CMC ظرفیت ارتباطی بین گره محلی و گره دور (job/unit of time) می باشد.

در الگوریتم امتیاز دهی، فرومون یک گره به میزانی که به آن کار فرستاده می شود، به صورت زیر در جدول فرومون افزایش می یابد:

$$SC' = \phi \cdot X \quad 0 < \phi < 1 \quad (3)$$

$$p^{new} = \gamma \cdot p^{old} + (1 - \gamma) \cdot SC' \quad 0 < \gamma < 1 \quad (4)$$

در الگوریتم نشانه گذاری جدول فرومون بدین صورت بهنگام نمی شود بلکه هر مورچه هنگام ترک کردن یک گره و ورود به گره دیگر ردیف متناظر با گره مقصد را در جدول فرومون به میزان معینی افزایش می دهد. در الگوریتم تمام تصادفی هیچگونه بهنگام سازی ای صورت نمی گیرد.

کاهش امتیاز²⁶: روش نشانه گذاری دارای این مشکل است که

در مواردی که دو گره از قدرت پردازشی و الگوی دریافت بار یکسانی برخوردار باشند، احتمال بروز وضعیت پیچگی پنگی افزایش می یابد. برای کاهش احتمال بروز چنین وضعیتی استفاده از کاهش امتیاز پیشنهاد می شود. بدین صورت که فرومون گره فرستنده (i) در جدول فرومون گره گیرنده به صورت زیر کاهش می یابد:

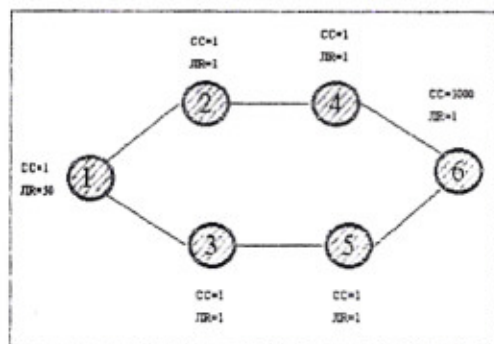
$$p_i^{new} = \psi \cdot p_i^{old} \quad 0 < \psi < 1 \quad (5)$$

که مقدار مناسب برای ψ ، می تواند نسبت کل کارهای دریافت شده از گره فرستنده به مجموع کل کارهای دریافت شده از سایر گره ها باشد:

²⁶ Remote node

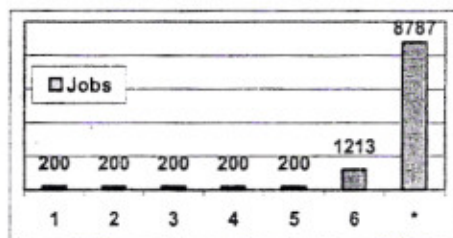
²⁷ Down scoring

3	1	1	4
4	1	1	4
5	1	1	4
6	1000	1	4

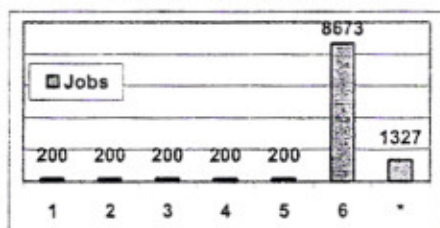


شکل ۵-نمای اتصالات شبکه سازی اول تا سوم

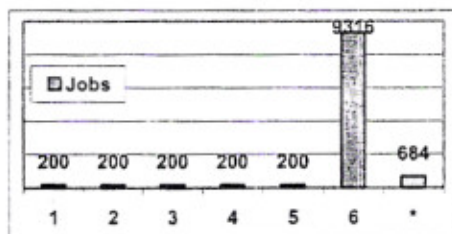
در شبکه سازی اول توانایی سیستم در توازن بار در کوتاه مدت مورد بررسی قرار می گیرد. نتایج شبکه سازی ها برای هر الگوریتم در شکل ۶ (الف-ب-پ) نشان داده شده است. سنون آخر که با ستاره مشخص شده است، نشانگر تعداد کارهای پردازش نشده می باشد.



شکل ۶-الف-تمام تصادفی



شکل ۶-ب-نشانه گذاری



شکل ۶-پ-امتیازدهی

شکل ۶-توزیع بار هر گره در مرحله ۱۲۰۰م

شکل ۷ سه الگوریتم را روی نرخ کارهای پردازش نشده با یک دیگر مقایسه می کند. منظور از نرخ کارهای پردازش نشده، نسبت

جدول فرمون هر گره پس از مدتی بصورتی تعیین می شود که گره-هایی که پذیرای بار بیشتری بوده اند دارای فرمون بیشتری در جدولهای فرمون خواهند بود و بدین ترتیب مورچه های تولید شده توسط یک دمون با احتمال بالایی مستقیماً به گره های مادون بار شده هدایت می شوند و در نتیجه توزیع بار مناسب تری حاصل می شود. در الگوریتم نشانه گذاری، مورچه ها هنگامیکه گره ای را ترک می کنند فرمون متناظر گره مقصد را در گره مبدأ افزایش می دهند. از چنین روشی در [۱] و [۲] استفاده شده است.

برای اینکه تغییرات گذرا^{۲۸} در عملکرد سیستم تاثیر کمتری داشته باشند و سیستم بتواند خود را با تغییرات سازگار کند، مقادیر فرمون در جدول فرمون در هر مرحله زمانی بصورت زیر با ضریبی کاهش می یابند.

$$p_{new} = \delta \cdot p_{old} \quad 0 < \delta < 1 \quad (۸)$$

که δ پارامتر بخار شدن^{۲۹} می باشد.

۵-نتایج شبکه سازی ها

هدف از شبکه سازی های انجام گرفته، نشان دادن برتری الگوریتم امتیازدهی نسبت به دو الگوریتم تمام تصادفی و نشانه گذاری می باشد. پنج شبکه سازی انجام گرفته است. شبکه سازی های اول، دوم و سوم که در یک گرید با تعداد گره های کم و تعداد اتصالات پایین انجام گرفته است، توانایی توازن بار و تطبیق پذیری کدام از سه الگوریتم را در محیط گرید بررسی می کنند. آزمایشهای چهار و پنج سه الگوریتم را در یک گرید با تعداد اتصالات بالا و تعداد گره های بالا مورد بررسی قرار می دهد.

۵-۱-شبکه سازی اول

شبکه سازی در یک گرید به صورتی که در جدول ۱ و شکل ۵ نشان داده شده انجام گرفته است. JIR نرخ تولید کار در گره می باشد. Ants تعداد مورچه های که دمون در هر بار تولید می کند را نشان می دهد که در این شبکه سازی این مقدار را ۲/۳ تعداد گره ها در گرید در نظر گرفته شده است. در ابتدا مقدار فرمون همه گره ها در همه جدولهای فرمون مساوی و برابر $1/N$ (تعداد گره ها) مقداردهی شده است.

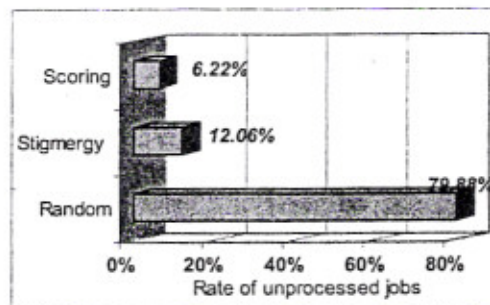
جدول شماره ۱-مشخصات گره ها

NodeID	CC	JIR	Ants
1	1	50	4
2	1	1	4

²⁸ Transient changes

²⁹ Evaporation parameter

کل کارهای تولید شده به مقدار کارهای پردازش شده در گریه است. همانطور که مشاهده می شود برای الگوریتم امتیازدهی این نرخ نسبت به دو الگوریتم دیگر پایین تر است که این خود نشانگر کارایی بالاتر الگوریتم امتیازدهی می باشد.



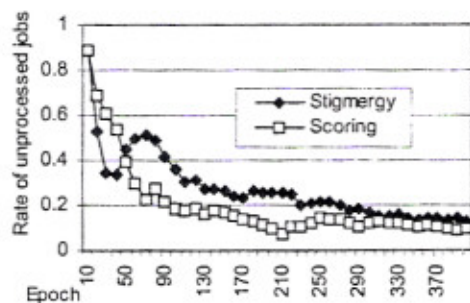
شکل ۲-مقایسه نرخ کارهای پردازش نشده در مرحله ۱۲۰۰م

۲-۵- شبیه سازی دوم

سیستم تا مرحله ۱۲۰۰م بگونه ای خود را پیکر بندی می کند که بارها را از گره شمار ۱ به گره شمار ۶ هدایت کند. در مرحله ۱۲۰۰م با افزایش قدرت پردازشی گره شمار ۵، به یک باره تمام بار آن گره پردازش شده (که در نمودار بصورت یک کاهش در نرخ کارهای پردازش نشده مشاهده می شود). پس از آن سرعت کاهش نرخ پردازش نشده ها برای مدت کوتاهی ثابت می ماند تا اینکه سیستم خود را با شرایط جدید وفق دهد. در واقع تغییرات اعمال شده که در پاراگراف فوق به آن اشاره شد خود را در مراحل ۲۱۰ تا ۲۴۰ نشان می دهد. روند کاهش نرخ پردازش نشده ها ادامه می یابد تا اینکه سیستم به تعادل برسد.

۴-۵- شبیه سازی چهارم

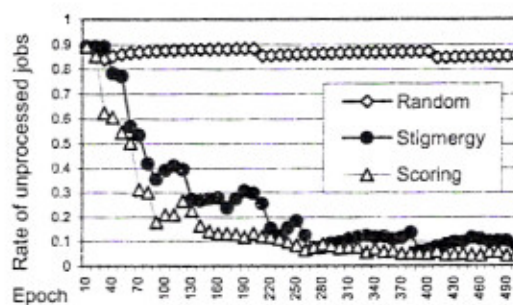
در شبیه سازی چهارم همگرایی برای یک شبکه با اتصالات کامل با ۱۰ گره در نظر گرفته شده است. هر خط ارتباطی دارای ظرفیت ۲۰۰ می باشد. قدرت پردازشی همه گره ها بجز گره های شمار ۱ و ۶ که دارای قدرت پردازشی ۵۰۰ می باشند، برابر ۱ است و نرخ تولید کار در همه گره ها برابر ۱۰ می باشد. تعداد مورچه تولید شده در هر مرحله برای هر گره برابر ۷ در نظر گرفته شده است.



شکل ۹-تطبیق پذیری در مرحله ۱۲۰۰م

نتایج شبیه سازی در شکل ۱۰ نشان داده است. همانطور که دیده می شود الگوریتم امتیازدهی نسبت به الگوریتم نشانه گذاری از

اصطلاحاً می گوئیم سیستم همگرا شده و یا به تعادل رسیده اگر نرخ کارهای پردازش نشده در سیستم بدون تغییر باقی بماند. در شبیه سازی دوم همگرایی الگوریتم های فوق الذکر، مورد بررسی قرار می گیرند. نتایج شبیه سازی نشان می دهد که نرخ کارهای پردازش نشده برای الگوریتم تمام تصادفی در بالاترین حد نسبت به دو الگوریتم دیگر قرار دارد و از بین دو الگوریتم امتیازدهی و نشانه گذاری، الگوریتم امتیازدهی دارای نرخ پایین تری است. نتایج شبیه سازی همچنین نشان می دهد که در مرحله ۱۵۰۰م نرخ کارهای پردازش نشده برای الگوریتم امتیاز دهی حدوداً ۴٪، برای الگوریتم نشانه گذاری حدوداً ۸٪ و برای الگوریتم تمام تصادفی حدوداً ۸۵٪ می باشد (شکل ۸).



شکل ۸-مقایسه همگرایی

۳-۵- شبیه سازی سوم

در این شبیه سازی توان سیستم در ارتباط با تطبیق پذیری مورد آزمایش قرار می گیرد. در مرحله ۱۲۰۰م نرخ تولید کار در گره شمار ۱، از ۵۰ به ۱ و گره شمار ۲، از ۱ به ۵۰ تغییر می کند. همچنین

عملکرد بهتری برخوردار است. در انتهای مرحله ۵۰۰ام نرخ کارهای

[1] A. Andrzejak, S. Graupner, V. Kotov and H. Trinks, "Algorithms for self-organization and adaptive service placement in dynamic distributed systems", Technical Report, HP Laboratories Palo Alto (www.hpl.hp.com/techreports/2002/HPL-2002-259.pdf), September 2002.

[2] A. Montresor, H. Meling, and O. Babaoglu, "Messor: load-balancing through a swarm of autonomous agents," *Lecture Notes in Computer Science*, Volume 2530/2003, 2003, pp. 125-137.

[3] A. Montresor, H. Meling, and O. Babaoglu, "Anthill: a framework for the development of agent-based peer-to-peer systems," *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, Vienna, Austria, July 2002, pp. 15.

[4] Y. Wang and J. Liu, "Dynamics of agent-based load balancing on grids," *Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems*, July 2003, pp. 804-811.

[5] R. Schoondewoerd, O. Holland et al., "Ant-like agents for load balancing in telecommunications networks," *First ACM International Conference on Autonomous Agents (Agents'97)*, February 1997, pp. 209-216.

[6] C. Z. Xu and B. Wims, "A mobile agent based push methodology for global parallel computing," *Concurrency: Practice and Experience Volume 12*, July 2000, pp. 705-726.

[7] J. Cao, D. J. Kerbyson et al., "Performance Modeling of parallel and distributed computing using PACE," *proceedings of 19th IEEE International Performance, Computing and Communications Conference (IPCCC 2000)*, Phoenix, USA, February 2000, pp. 485-492.

[8] J. Cao, S. A. Jarvis et al., "ARMS: An agent-based management system for grid computing," *Scientific Programming 10(2002)*, ISSN 1058-9244, IOS Press, 2002, pp. 135-148.

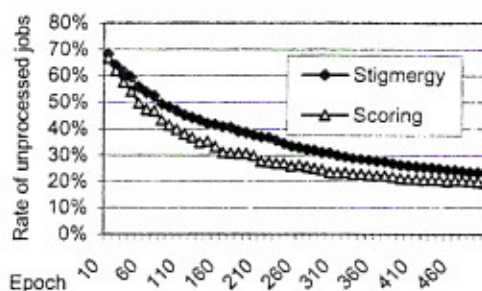
[9] J. Cao, D. J. Kerbyson and G. R. Nudd, "Use of agent-based service discovery for resource management in metacomputing environment," *Proceedings of 7th International Euro-Par Conference*, Manchester, UK, Lecture notes Science 2150, Springer Verlag, August 2001, pp. 882-886.

[10] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, Volume 1, No. 1, 1997, pp. 53-66.

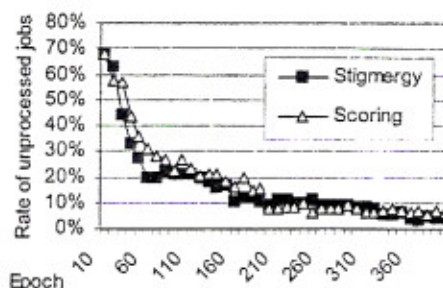
پردازش نشده برای الگوریتم امتیازدهی (۱۹/۴۱٪)، در حالیکه این مقدار برای الگوریتم نشانه گذاری ۲۳/۳٪ می باشد.

۵-۵- شبیه سازی پنجم

در این شبیه سازی خصوصیت تطبیق پذیری برای یک گرید مشابه گرید شبیه سازی ۴ مورد بررسی قرار گرفته است. الگوی توان پردازشی، در هر ۱۰ مرحله یک بار تغییر می کند. توان پردازشی دو گره که به صورت تصادفی انتخاب می شود، برابر ۵۰۰ و برای گره های دیگر برابر ۱ قرار داده می شود. شکل ۱۱ رفتار این دو الگوریتم را با هم مقایسه می کند. همانطور که مشاهده می شود الگوریتم امتیاز دهی دارای تطبیق پذیری تا حدودی بالاتر از الگوریتم نشانه گذاری است.



شکل ۱۰- همگرایی در شبکه با اتصالات زیاد



شکل ۱۱- تطبیق پذیری در شبکه با اتصالات زیاد

۶- نتیجه گیری

در این مقاله الگوریتم جدیدی مبتنی بر کلونی مورچه ها بنام امتیاز دهی، برای توازن بار در گریدهای وابانشی معرفی گردید و با دو الگوریتم تمام تصادفی و نشانه گذاری مقایسه گردید. با استفاده از شبیه سازی کامپیوتری نشان داده شد که روش امتیاز دهی چه در شبکه های با اتصالات کم و چه در شبکه های با اتصالات بالا، دارای قابلیت تطبیق پذیری و همگرایی بالاتری نسبت به دو الگوریتم نشانه گذاری و تمام تصادفی می باشد.



انجمن کامپیوتر ایران
Computer Society of Iran



مجموعه مقالات (جلد اول)

(مشمول بر مقالات فارسی)

۲۸ تا ۳۰ بهمن ماه ۱۳۸۲

دانشگاه صنعتی شریف

نقدیه

کنفرانس سالانه

انجمن کامپیوتر ایران



مرکز
مخابرات
و ارتباطات
پیشرفته