روش دگردیسی ارزیابی پایداری (SAMA) برای برنامه ریزی موثر بر اساس تحمل شکست در شبکه ی کامپیوتری

شمیلا ابنزر- باسکاران

دپارتمان مهندسی و علوم کامپیوتر- دانشگاه کارونیا – کویمباتور – هند

کلید واژه ها:

چک پوینت گذاری

محاسبه شبکه ای

تحمل شکست

نرخ احیا

برنامه ریزی

چکیده

محاسبه ی شبکه ای (grid) اجازه ی هماهنگی و به اشتراک گذاری منابع کنترل شده و حل مسایل در سازمان های چند نهادی، دینامیک و مجازی را می دهد. علاوه بر آن تحمل شکست و برنامه ریزی وظیفه یک مسئله ی مهم برای شبکه های محاسباتی مقیاس بالا است زیرا طبیعت غیر قابل پیش بینی دارد. تکنیک هایی که به صورت عمده مورد استفاده قرار می گیرد برای درک تحمل شکست، چک پوینت گذاری های دوره ای است که به صورت دوره ای وضعیت کار را ذخیره می کند. اما یک فاصله زمانی چک پوینت گذاری نامناسب باعث تاخیر در اجرای کار می شود و خروجی کل را کاهش می دهد. با این نگرانی، مقاله ی حاضر تلاش می کند تا اطمینان حاصل کند که کارایی بهتری را در شبکه های کامپیوتری با سیستم های تحمل شکستی موثرتر و قابل اعتماد تر با استفاده از یک روش دگردیسی ارزیابی پایداری (SAMA) داریم. در اینجا استراتژی استفاده شده برای رسیدن به تحمل شکست به صورت تطبیق چک پوینت ها بر اساس وضعیت جاری و اطلاعات شکست قبلی در مورد منابع به صورت دینامیکی است که در مرکز اطلاعات نگهداری می شود. فرایند موثر برنامه ریزی می تواند با برنامه ریزی بر اساس تحمل خطل به دست بیاید که شامل تعیین نرخ انحراف تمام گره ها با استفاده از تعدادی محدودیت ارزیابی با پایداری بالا است. این باعث می شود که کار درون زمان ضرب الاجل با خروجی بهتر انجام شود و یک مسیر را برای قابل اعتماد کردن محیط شبکه می پیماید.

1. معرفی

در کل یک شبکه ی کامپیوتری یک طبقه بندی از منابع نرم و سخت افزاری است که دسترسی بدون رخنه و فراگیر را به پتانسیل های محاسباتی با تکنولوژی بالا فراهم می کند. به صورت وسیعی در محاسبات علوم، تکنولوژی، فرایند کسب و کار و مهندسی استفاده شده است. منابع در شبکه ممکن است به صورت جغرافیایی در چندین حوزه ی مدیریتی پیاده سازی شده باشند که می توانند بر روی بر سیستم عاملی مانند ویندوز، لینوکس یا یونیکس نصب شده باشند. با این کاربردها، کاربران ممکن است نیازمندیهای کامپیوتری و سایر منابع را در سیستم داشته باشند. عمدتا نقش سیستم شبکه می تواند در سه بخش دسته بندی شود. اول یک شکننده محاسبه برای تقسیم کارها به وظایف کوچکتر مورد استفاده قرار می گیرد. این باعث می شود که کار سریعتر به نتیجه برسد با پردازش بر روی چندین منبع موجود. دوم سرویس اطلاعات منابع شبکه (GRIS) که به منزله ی یک تابع است مانند نگهداری و کشف جدول منبع. سوم برنامه ریز وظیفه مسئول اعزام وظایف به منابع بر اساس الگوریتم برنامه ریزی است (1).

علاوه بر این، معماری شبکه و خوشه برای کاربردهای موازی با تمرکز محاسباتی شهرت یافته است. بنابراین پیچیدگی زیرساخت شامل ذخیره ی انبوه، گره های محاسباتی و شبکه های در هم مرتبط، چالش های زیادی را با توجه به قابلیت اعتماد کلی سیستم فراهم می کند. در مقایسه با سایر محیط های توزیع یافته پیچیدگی شبکه معمولا از غیر هم جنس بودن منابع و مدیریت غیر متمرکز نشات می گیرد. این خواص به صورت مکرر باعث تفاوت های قوی در دسترسی می شود که خود مخصوصا به نرخ شکستی سیستم و منابع، سیاست های مدیریتی و نوسانات در بار سیستم بستگی دارند. مشخصا ویرایش های زمان اجرا در دسترسی سیستم می توانند به صورت قابل توجهی بر روی اجرای کار تاثیرگذار باشد. بنابراین برای یک گروه بزرگ از کارهای زمان گیر ، تاخیر و کمبود قابل قبول نیستند به این صرت تحمل شکست باید در نظر گرفته شود. یک مواجهه ی بزرگ در تامین تحمل شکست در یک محیط توزیع یافته وجود دارد در حال بهینه سازی زمان اجرای کار و استفاده از منابع. برای انجام آن، اقتباس دینامیک از تکنیک چک پوینت در این مورد استفاده شده است، و بر اساس تاریخچه ی شکستی منبع و وضعیت جاری از کار است که بر سربار چک پوینت غلبه می کند به عنوان چک پوینت گذاری دوره ای. و بنابراین، فرایند تحمل شکست را با خروجی بهتر شده به دست می آورد. شکل 1 نشان دهنده ی محیط عادی شبکه ای است که شامل تعدادی از گره های شبکه و یک گره کنترل است.

تحمل شکست باعث رسیدن به اعتماد پذیری سیستم می شود. اعتمادپذیری در ارتباط با تعدادی جنبه های کیفیت خدمات است که توسط سیستم ارایه می شود که ممکن است شامل ویژگی هایی مانند اعتمادپذیری و در دسترسی بودن باشد. اعتماد پذیری می گوید که یک سیستم می تواند به صورت مداوم بدون شکست کار کند. یک سیستم بسیار قابل اعتماد سیستمی است که بدون هیچ وقفه ای در مدت زمان نسبتا طولانی کار کند. در دسترس بودن گویای این است که یک سیستم فورا برای استفاده آماده است. تکنیک تحمل شکست به صورت عادتی برای افزایش نرخ در دسترس بودن و اعتمادپذیری در محاسبات استفاده می شوند. علاوه بر آن قابلیت اعتماد به عنوان احتمالی که سیستم به صورت مداوم به شکل عملیاتی در زمان 0 تا t باقی می ماند تعریف می شود. همچنین گفته می شود که اعتمادپذیری به صورت زیادی در ارتباط با نرخ های زمان متوسط شکست (MTBF) و زمان متوسط بین شکستها (MTBF) است. MTBF به عنوان زمان متوسط که سیستم می تواند تا هنگام وقوع یک شکست کار کند داده می شود در حالی که MTBFبه عنوان زمان متوسط بین دو شکستی متوالی است. اختلاف بین دو نرخ گفته شده به دلیل زمان مورد نیاز برای نو کردن سیستم است بعد از اینکه اولین شکست اتفاق افتاد. نمایش زمان متوسط برای تعمیر توسط MTTR، همچنین می گوید که MTBF=MTTF+MTTR. همچنین توضیح می دهد که تکرار چک پوینت های یک سیستم بر اساس MTTR آن است. شکل 2 یک مثال برای طبقه بندی اعتمادپذیری به گونه ای است که کارایی شبکه کامپیوتری بر اساس آن است.

علاوه بر آن، سیستم های براساس شبکه به ما برنامه ریزی تخمل شکستیی را می دهند با ارتباط دادن یک مجموعه از تکنیک های تحمل شکست برای تحمل شکستهای حیاتی در اجزای سیستم شبکه (3).

* تحمل شکست در هماهنگی مدل سازنده می تواند در دو سطح اندازه گیری شود. تحمل شکستی فضای تاپل یعنی مساله ی حمایت که فضا ناموفق نخواهد شد اگر شکستهایی در خود فضای تاپل وجود دارد و
* تحمل شکستی سطح کاربرد، که اطمینان حاصل می کند که رضایت کاربرد و تعدادی مشخصات اعتمادپذیری تامین شده است اگر بعضی از فرایندهای کاربردی خراب شوند.

برای تحمل شکستها در سیستم شبکه، کاربردهای موازی به صورت کلاسیک خود را با توانایی چک پوینت گذاری محاسبه ی خود یا وضعیت اجرایشان جهت پایدارسازی حافظه، مجهز می کنند. هنگامی که یک یا بیشتر از پردازش ها شکست بخورند و وظیفه ی خود را تمام نکنند، کاربرد می تواند از نزدیکترین چک پوینت شروع به کار دوبار بکند، به این ترتیب مقدار انجام دوباره ی محاسبات را که باید انجام شوند کاهش می دهد (4). با در نظر داشتن آن نگرانی ها، کار پیشنهاد شده بر روی فرایند برنامه ریزی موثر تمرکز می کند بر روی شبکه ی گره های توزیع شده که به مقدار زیادی تخمین شکست و تحمل را در بر می گیرد. مشوق اصلی این پروپوزال این است که اعتماد پذیری و کارایی را برای برنامه ریزی در شبکه ی محاسباتی فراهم کند.

باقی مانده ی این مقاله به این گونه سازماندهی شده است. بخش 2 در مورد جزئيات کارهای مرتبط صحبت می کتدف. بخش 3 روش دگرگونی ارزیابی پایداری را ارایه می دهد که در افزایش کارایی برنامه ریزی و اعتمادپذیری موثر است. بخش 4 نتايج تجربی را ارایه می دهد و بخش 5 با اشاره به کارهای آینده مقاله را پایان می دهد.

2. کارهای مرتبط

به دلیل اینکه برنامه ریزی در شبکه ی کامپیوتری یک مشکل حل نشده باقی‌مانده است، تحقیق در این حوزه هنوز هم یک فرایند در جریان است. یک تکنیک که به نام استراتژی برنامه ریزی تحمل شکست در بعضی کارها پیشنهاد شده است (۴) برای شبکه‌های کامپیوتری. فرایند متشکل از روش تحمل شکستی براساس تکرار با مکانیزم برنامه‌ریزی کار MTTR است.

چک پوینت گذاری کار براساس حالت شکستی منبع در این الگوریتم تحمل شکستست. کارهای اصلی به این صورت هستند:

* مکانیزم تحمل شکست برای شبکه ی کامپیوتری
* استراتژی برنامه‌ریزی براساس MTTR
* به کار بردن مکانیزم بازیافت چک پوینت
* پیاده‌سازی ها با مجموعه ابزار Globus انجام شده‌اند

با ادامه ی آن، یک سیستم جدید برای هماهنگی شفاف بازشروع-چک پوینت متعلق به کاربردهای شبکه ی توزیع شده در خوشه های محاسباتی استفاده شده است. آن‌ها چک پوینت ZapC را معرفی کرده‌اند که به صورت موثری عملیات را به صورت موازی در چندین گره محاسباتی بازشروع کرده است (۶). علاوه برآن کاربرد معیار بازتکرار-چک پوینت برای بازبرنامه ریزی به دست آمده است. مؤلف تحقیق را به طرفی برده است که کارهای موثری را ارایه دهد که نتایج بهتری داشته باشد.

یک روش دیگر که در این مقاله ارایه شده است (۷) توضیح دهنده ی حالات مختلف به دست آمده از یک وظیفه است در حالی که در حال به دست آمدن است. گفته شده است زوجیت بین منابع ممکن از شکستها و جلوه‌های ضربه‌ای آن‌ها در یک محیط شبکه ای در هنگام جمع شدن با عدم قطعیت داده‌های حالت منبع و وضعیت‌های عملیات خدمات، گویای این است که مدلهای معین تخمین شکست ممکن است غیرموثر و نامناسب باشد. هرچند، حالات متفاوتی که در سیستم تحمل شکست درگیر بوده‌اند عبارتند از حالت پایدار، حالات آسیب پذیر، حالات شکست و حالت نگهداری. همراه با آن‌ توضیحات، یک کار (۸) نشان دهنده ی یک روش برای عملیات و اعتمادپذیری برای آزمایش توزیع‌های مختلف زیروظایف در یک شبکه با استفاده از یک ساختار درخت بوده است که همراه با در نظر گرفتن ارتباطات درونی شکست و وابستگی‌های داده است. همچنین در این مقاله گفته شده است که فرایند هنوز نیاز به بهسازی دارد و باید دینامیک های ترخ شکستی متغیر با زمان را استفاده کند. یک روش متفاوت در مقاله ی (۹) پیشنهاد شده بود برای جمع آوری پایدار چک پوینت های جهانی در یک محیط سیستم توزیع شده. الگوریتم چک پوینت هایی را می‌سازد که باید تحت یک چک پوینت استوار جهانی باشد. این یک فرایند را باری می‌دهد که چک پوینت پایه ی برنامه‌ریزی شده ی ترتیبی را در یک زمان برنامه‌ریزی شده استفاده کند. مطابق یک تلاش دیگر (۱۰)، برنامه‌ریزی کار در شبکه به کمال خود رسیده است، با در نظر گرفتن توافق سطح خدمات (SLA). علاوه بر آن، فرایند برای مصرف مؤثر منابع مشترک است.

به کار بردن ساختار تقسیم و غلبه کمک می‌کند تا یک مکانیزم را بسازد که یک سربار کمتر دارد و کمتر از تکنیک های رایج مانند قراردادن چک پوینت مورد استفاده قرار گرفته است و در مقاله (۱۱) داده شده است. این کار مقدار کار زاید انجام شده بعد از یک توقف یک یا بیشتر گره را به حداقل می رساند. کار اشاره شده در (۱۲) یک بررسی را در مورد اطمینان از اعتمادپذیری سیستم شبکه ارایه می دهد. مؤلف تلاش‌هایی را مورد تأکید قرار داده است که بر روی مکان های جدای کاربردی از سیستم‌های شبکه تمرکز دارد.

* اعتمادپذیری در برابر سخت‌افزار محاسباتی، نرم‌افزار و ساختار داده دارد.
* اعتمادپذیری خدمات مدیریتی و زیرساخت ها
* اعتمادپذیری شبکه‌های کامپیوتری برای انتقال داده و پیغام رسانی

علاوه برآن، مقاله شامل قسمت‌های مختلف از شکست است، روش‌های تشخیص شکست و روشهای بازیابی با استفاده از چک پوینت ها است. پردازش چک پوینت و بازیابی (CPR) یک روش کار است که در مقاله (۱۳) ارایه شده است، مدیریت ثبت ماهرانه ی معاملات مورد اصلی در سربار چک پوینت است. بعضی از موارد خاص در سایر کار ها (۱۴) داده شده‌اند و بر روی استراتژی های پیشرفته ی برنامه‌ریزی در مورد محاسبه ی شبکه ای تمرکز کرده است. علاوه بر آن کار دیگری (۱۵) مفهوم توسعه ی یک سیستم مقاوم در برابر شکست را برای یک محیط شبکه ی محاسباتی با تعیین تاریخچه ی رخداد شکست، تعداد چک پوینت ها و غیره توضیح داده است. همان‌طور که در مقاله گفته شده است، محدودیت اصلی کار پیاده‌سازی سرویس تکرار چک پوینت با به دست آوردن تکرار در یک حالت سریعتر است.

تحمل شکست به عنوان شکست منابع (۱۶) تعیین می شوپ. استراتژی استفاده شده برای به دست آوردن تحمل شکست با اقتباس نسبی چک پوینت هایی است که بر روی تاریخچه ی اطلاعات شکست و حالت کنونی منابع تکیه دارند. اما هنوز یک ضرورت برای بهسازی روش برای اصلاح کارایی و اعتمادپذیری در برنامه‌ریزی وظیفه در ابر (cloud) وجود دار. با دنبال کردن آن، یک مقاله ی بررسی دیگر(۱۷) مسایل متفاوت در تحمل شکست در شبکه را توضیح داده است. برای به دست آوردن سطح بالایی از دسترسی و اعتمادوذیری، زیرساخت شبکه باید کاملاً در برابر شکست مقاوم باشد. به دلیل اینکه شکست در منابع به صورت کشنده ای بر روی اجزای کار تأثیر می گذارد، سرویس تحمل شکست برای ارضاء کردن محدودیت‌های کیفیت خدمات در شبکه محاسباتی حیاتی است. تکنیک های به صورت عام استفاده شده برای ایجاد تحمل شکست، تکرار و چک پوینت گذاری کار است. علاوه بر آن، مدل برنامه ریزی قابل اعتماد و کارار در مقاله ی (18) ارایه شده بود، براساس اعتبار و در دسترس بودن منابع شبکه. با اعتمادپذیری بهتر شده، یک کاهش بزرگ در احتمال شکست و فریب در استفاده از درخواست های کاربری خدمات وجود دارد. مکانیزم چک پوینت گذاری همچنین برای تحمل شکست مورد استفاده قرار گرفته است. بهسازی ممکن است مجموعه ای از اجبارهای کیفیت خدمات مرتبط با کاربر را در نظر بگیرد. الگوریتم بازبرنامه ریزی در کار دیگری (19) پیشنهاد شده است که یک امکان بسیار بااهمیت دارد و به عنوان یک الگوریتم جدید شناخته می شود، به دلیل اینکه می تواند با آرایه ی بزرگی از روش های ابتکاری بازربرنامه ریزی مورد استفاده قرار گیرد. همچنین ممکن است از روش های بیشتر بازبرنامه ریزی استفاده کند که طبقه بندی آن در ارتباط با موقعیت گره در گراف و مکانیزم تحمل شکستیی است که می تواند مورد استفاده قرار گیرد. روشهای تطبیق پذیر برنامه ریزی کار با تحمل شکست پیشنهاد شده اند(20). به دلیل اینکه الگوریتم های برنامه ریزی کار بسیاری در شبکه وجود داشت، بیشتر آنها بر روی برنامه ریزی کار با دانه بندی کوچک تمرکز کرده اند و دیگران تنها بر روی روش های برنامه ریزی کار های سبک وزن تنها مترکز هستند. در مورد وظایف بزرگتر و منابع بزرگتر، تنها چند الگوریتم برنامه ریزی در دسترس هستند. همچنین گفته شده است که برنامه ریزی کار در یک محیط شبکه ی پویا و ناپایدار مورد تحمل است و همچنین زمان اجرای کلی برای انجام تمام کارهای تخصیص داده شده در شبکه را کاهش می دهد.اما هیچ چک پوینتی برای بازیابی شکست مورد استفاده قرار نگرفته است. یک نمایش از چک پوینتهای امنیتی در مقاله ی (21) نمایش داده شده است.

برنامه ریز شبکه ی توزیع شده ی قابل اعتماد (rdgs) توسعه داده شده است (21) که نرخ برنامه ریزی موفق وظایف مرکب را افزایش می دهد. مولف پارامترهایی مانند اولویت و ضرب الاجل را برای وظایف مرکب با محاسبات و ارتباطات بالا در نظر گرفته است. شامل کردن الگوریتم های کشف منبع به عنوان مهاجرت بعدی وظایف در اجرا داده شده است.

از نظر جنبه های دیگر، مهاجرت وظیفه ی براساس نگاشت ثابت اولیه، در مقاله ی (23) توضیح داده شده است. به این ترتیب آن ها به کارایی بالا در شبکه ی محاسباتی رسیده اند. با اینکه مکانیزم خروجی را به حداکثر می رساند، یک نیاز برای در نظر گرفتن تاخیر بیشتر و وضعیت محیطی وجود دارد. علاوه بر این، در تمرکز بر روی نزدیکی چک پوینت در تحمل شکست، یک چک پوینت افزایشی با دانه بندی کوچک در کار دیگری (24) معرفی شده است. این می تواند اندازه فایل چک پوینت را به صورت قابل توجهی بدون سربار کاهش دهد. با سوار شدن زمان بین چک پوینت ها، اختلاف حجم فایل چک پوینت بی دو شما برای بسیاری از کارها بسیار کوچک است.

1. مکانیزم پیشنهاد شده
2. مدل شبکه ی داده شده در این مقاله شامل گره های محاسباتی به صورت جغرافیایی پیاده شده، منابع و تعدادی از خدمات برای اجرای موثر وظایف است که باید از مراکز فرایند تخصیص شوند. به عنوان هدف غالب این مقاله برای برنامه ریزی وظایق برای گره های شبکه که می توانند به صورت کارایی وظایف داده شده را انجام دهند، تحمل شکستی گره ها بیشتر براساس تاریخچه ی قبلی آن در نظر گرفته می شود. علاوه بر آن، اقتباس تکنیک چک پوینت برای افزایش اعتمادپذیری بر محیط شبکه است. داده های چک پوینت در طول کل فرایند پایدار می شوند. خدمت دهنده ی اطلاعات (is) منابع و اطلاعات مربوط به وضعیت کار را توسط فرایند چک پوینت گذاری و برنامه ریز کار جمع آوری و نگهداری می کند. تاریخچه ی اطلاعات در مورد هر منبع را امتداد می دهد. علاوه بر آن توسط این روش پایداری و اعتمادپذیری هر گره پخش شده تحت یک محیط شبکه با یک نرخ شکست و مفروضات چک پوینتی در فرایند برنامه ریز کار ماهر، آزمایش می شود.

3.1. تخمین نرخ شکست

براساس تاریخچه ی قبلی توانایی اجرای گره های شبکه، نرخ شکست در اینجا تخمین زده می شود. به ما اجازه دهید با این فرضیات و مشتقات به جلو رویم.

Fi = تعدداد شکست / (زمان اجرای کل)، I = {1,2,3,….N} به گونه ای که N تعداد گره ها در شبکه است

T = زمان کل گرفته شده برای اجرا در یک گره



با گرفتن لوگاریتم:



مشتق گیری از t:



با این نرخ اعتمادپذیری، معادلات مشتق شده از هر و همه ی گره ها در زمان t تعیین می شود. بنابراین اعتمادپذیری گره (NRt) در زمان به این صورت داده می شود:



به گونه ای که f نمایش دهنده ی نرخ شکست است. با دنبال کردن آن، زمان متوسط برای گرفتن شکست یا شانس شکست (F) از هر گره با این تخمین زده می شود:



این حالت زمانی اتفاق میافتد که ماشین به صورت غیرمنتظره ای دچار شکست می شود. علاوه براین داده شده است که، ارزیابی بر اساس زمان متوسط بین دو شکست متوالی است. در تطابق با آن، اگر یک شکست رخ دهد، زمان تکمیل یک کار خاص از آن نرخ مورد انتظار بیشتر خواهد بود.

3.2 چک پوینت گذاری آغازین

به جای ثبت رویدادها، چک پوینت گذاری بر روی ذخیره ی گاه و بیگاه حالت محاسبات بر روی حافظه ی ثابت تکیه دارد. اگر یک شکست اتفاق بیافتد، محاسبات از یکی از وضعیت های ذخیره شده ی قبلی شروع می شود. به دلیل اینکه محاسبات توزیع یافته است، فضای مصالحه ی بر روی سناریوهای چک پوینت گذاری محلی و جهانی و هزینه ی بازیابی منتج شده ی آن ها باید در نظر گرفته شود. بنابراین چک پوینت گذاری بر اساس روشها به شیوه ای متفاوت است که فرایندها باهم هماهنگ می شوند و در منبع یک وضعیت پایدار جهانی، به گونه ای که حالت پایدار جهانی می تواند یا در زمان بازیابی برگشتی یا در زمان چک پوینت گذاری باشد. شکل 3 در ادامه نشان دهنده ی دسته بندی شکست با توجه به قابلیت بازیابی است.

عملیات چک پوینت گذاری بر روی یک منبع، اجرای یک کار تک به صورت دیاگرامی در شکل 4 نمایش داده شده است که در پایین توضیح داده شده است:

فاز 1. کار به منبع (r) تحویل داده شده است، بعد از اجرای یک بازه ی زمانی I، اجرای کار بر روی یک منبع فعال یک درخواست چک پوینت را می سازد.

فاز 2. برای هر منبع، فرایند آخرین زمان شکست را از سرویس دهنده ی اطلاعات می گیرد، هنگامی که هیچ شکستی روی نداده است، آخرین زمان شکست با زمان شروع سیستم آغاز می شود.

فاز 3. درخواست چک پوینت گذاری ساخته شده با کار، توسط برنامه ریز تخمین زده می شود. فرض شده است که منبع ثابت است و چک پوینت برای جلوگیری از سربار حذف می شود.

فاز 4. برای جلوگیری از حذف تعداد زیادی چک پوینت یک تعداد حداکثری از محدودیت های حذفی باید مورد تصمیم قرار گیرد. بنابراین این روش سربار چک پوینت را با حذف چک پوینت های قابل جلوگیری کاهش می کند.

بنابراین چک پوینت گذاری آغازین یک قابلیت را برای ذخیره ی یک حالت کار حد واسط ایجاد می کند. هنگامی که یک کار طولانی مدت نیازمند ذخیره ی بعضی حالت های حد میانی برای محافظت از شکست های گره هاست، این مفید خواهد بود. سپس در احیای یک گره شکست خورده، یک کار باید چک پوینت های ذخیره شده را بارگزاری کند و از جایی که کار را ترک کرده است، ادامه دهد.

3.3 ارزیابی پایداری گره

در این فرایند، پایداری گره های شبکه با بعضی محدودیت ها مانند تعداد شکست ها (n)، زمان مورد انتظار فرایند، زمان طی شده ی یک فرایند و نرخ بازیابی، ارزیابی می شود. علاوه بر این، نرخ افت و خیز (FR) یک نقش مهم را در ارزیابی پایداری و اعتمادپذیری گره های شبکه برای برنامه ریزی کار کارا را دارد. مدل پیشنهاد شده رویداد شکست یک گره شبکه در توزیع پواسن آسان می کند. فرض کنید i = {1و2و3و....N} باشد، به گونه ای که N تعداد گره ها است، هر کدام یک توزیع پواسن را دنبال می کنند و همچنین به عنوان متغیرهای تصادفی در نظر گرفته می شوند.

با توجه به این حالت، در نظر بگیرید که تابع توزیع D(x)، به گونه ای که x نرخ بازیابی است. اینجا توضیح نرخ بازیابی (x) در زمان گرفته شده توسط یک گره برای کامل کردن یک کار خاص داده می شود، بعد از شکست که نرخ شانس fi را در نظر می گیرد. برای محاسبه ی درست، توزیع ارلنگ در نظر گرفته شده است. این یک نوع عمومی از توزیع آماری است که در ارتباط با توزیع بتا است و به صورت مداوم در فرایندی برمی خیزد که گره ها دارای زمان های انتظار مرتبط هستند. علاوه براین تابع توزیع را برای به دست آوردن نتایج محاسبه ی عدد حقیقی عمومی سازی می کند. به این صورت داده می شود:



علاوه بر آن، مساله ی اصلی تخمین نرخ افت و خیز زمان مورد انتظار برای تکمیل در هنگامی است که محدودیتهای شکست ثابت است یا با توجه به زمان تکمیل اجازه داده می شود. مفروضاتی که باید در نظر گرفته شود در محاسبه ی نرخ افت و خیز به این صورت هستند:

فرض کنید که x1و x2 نرخ های بازیابی مورد انتظار با توجه به انجام کار برای انتظار داشتن زمان پردازش و زمان پردازش طی شده باشد. در ادامه، t1 و t2 احتمال مورد انتظار برای زمان تکمیل هستند، f نرخ شکست و n1 و n2 تعداد شکست های روی داده در هنگام پردازش است که به روشی که پیشتر گفته شد به دست می آید. با این مفروضات، ارزیابی بعدی مطابق با توزیع ارلنگ روی می دهد.



واضح است که شکست ممکن است افزایش یابد در حالی که تعداد چک پوینت ها افزایش می یابد (یعنی) شکست ها ممکن است فراتر از چک پوینت ها باشند. بنابراین معادله ی ادامه یافته X2 بالاتر از یک خواهد بود که تحت تاثیر توزیع ارلنگ اینچنینی می شود:



نرخ افت و خیز با حل کردن معادله ی (7) و (8) به دست می آید. به این ترتیب داده می شود:



نرخ افت و خیز تمام گره ها در یک محیط شبکه ای به این صورت به دست می آید. با استفاده از نتایج محاسبه، گره ها رده بندی می شوند.

3.4. تابع رده بندی

بعد از محاسبه ی نرخ افت و خیز تمام گره ها، رده بندی ممکن است اتفاق بیافتد با توجه به آن. فرایند ایجاد رده بندی شامل تعیین تابع رده بندی Rبر اساس ترخ افت و خیز یک گره است. FR از زمان پردازش یک کار با توجه به تاریخچه ی قبلی بارهای کاری به دست می آید. هرچند شامل انتخاب گره ها با نرخ افت و خیز پایین است که کارایی کلی شبکه را افزایش می دهد.

از مجموعه ی گره های N، گره های با پایداری بالا و افت و خیز پایین به دست می آیند، تابع رده بندی به کار برده می شود. این برنامه ریزی موثر را پشتیبانی می کند. بنابراین برنامه ریزی توسط برنامه ریز برای کارهای تخصیص داده شده از مجموعه ی فرایندها به اجرا در می آید. شکل 5 نشان دهنده ی جریان کلی کار پیشنهاده داده شده است.

بنابراین، ثابت شده است که مدل یک تخمین خوب برای رفتار شبکه ی کامپیوتری است، در حالی که زمان های شبیه سازی کوتاهی دارد، پیچیدگی نسبی محاسباتی پایینی دارد و نتایج ماهرانه ای را به دست می آورد. اکنون، فرض شده است که مدل پیشنهاد شده کاملا قابل اعتماد است و تحمل خطا می تواند به صورت موثری به دست آید با استفاده از چک پوینت گذاری و تخمین نرخ افت و خیز.

1. نتایج تجربی

برای تامین سند برای مکانیزم پیشنهاد شده، نتایج با استفاده از برنامه ی GridSim ایجاد می شوند. علاوه بر این، برای مقایسه ی کارایی روشهای ابتکاری پذیرفته شده برای چک پوینت گذاری، مدل شکست سیستم و بارهای کاری واقعی مورد استفاده قرار می گیرد. در این استراتژی شبیه سازی، پارامترهای مدل به گونه ای مقدار دهی اولیه می شوند که نمایش دهنده ی یک سیستم شبکه با بار بسیار سنگین باشند که دارای یک الگوی رسیدن کار هستند. برای شبیه سازی، 400 شغل و 200 منلع پردازشگر در نظر گرفته شده اند. نیازمندی های منابع برای کارها بین 1000 و 3000 MIPS (میلیون عملیات بر ثانیه)در نظر گرفته شده اند. نمایش دهنده ی یک کار است که نیازمند اجرا توسط یک منبع پردازشی است که حداقل به اندازه ی نیازمندی های منابع توانایی داشته باشد.

علاوه برآن، حجم کار برای هر کار به صورت تصادفی بین 300000 و 400000میلیون عملیات ایجاد می شوند و سرعت منبع پردازشی به صورت دلخواه بین 1000 تا 3000MIPS ایجاد می شود. 4 سطح از گره های شبکه برای هر 500MIPS وجود دارد. عوامل جامع شبیه سازی در ادامه در جدولی داده می شوند:

در اینجا، نرخ افت و خیز به عنوان اختلاف بین زمان پردازش مورد انتظار و زمان پردازی اضافی توضیح داده می شود. با این فرضیات، ارزیابی کار پیشنهاد شده در ادامه انجام می شود. شکل 6 در ادامه نشان دهنده ی ارتباط میانی بین فاصله زمانی متوسط چک پوینت و تعداد کارهایی است که به صورت موفق به اجرا در آمده اند.

واضح است از گراف که هنگامی که تعداد کارهای افزایش می یابد، یک افزایش در مقدار فاصله زمانی چک پوینت ها به وجود می آید. ارزش ها به صورتی تنظیم شده اند که با یک ذخیره ی موثر وضعیت های اجرا برای بازیابی سریع منطبق باشد، در حالی شکست ها در حال رخ دادن هستند. با دنبال کردن آن، شکل 7 در پایین نشان داده است که ارتباط بین کارها و تعداد چک پوینت ها به چه صورت است. به صورت واضح، تعداد کارها به صورت مستقیم با تعداد چک پوینت های ایجاد شده تناسب دارد. شکل 8 معرفی شده در بالا یک مثال برای انطباق گره ها و نرخ متوسط افت و خیز است. در اینجا، نرخ افت و خیز از معادله ی (9) محاسبه می شود.

(شکل ها و جداول در این قسمت قرار می گیرند)

FR بر اساس نرخ بازیابی گره ها، زمان پردازش مورد انتظار و زمان پردازش طی شده تعیین می شود. گراف به طرف بالا اشاره می کند هنگامی که یک افزایش در تعداد کارهای در آینده پردازش شده توسط یک گره خاص در شبکه اتفاق می افتد. علاوه بر آن پایداری گره ها با تعدادی محدودیت های ارزیابی پایداری که شامل پارامترهای بعدا اشاره شده ارزیابی می شود که در تعیین FR درگیر هستند.

به عنوان هدف این کار، تحمل خطا به صورت اصلی بر روی برنامه ریزی مار متمرکز است. بعد از محاسبه ی نرخ افت و خیز، رده بندی گره ها برای برنامه ریزی موثر انجام می شود. گره های با نرخ افت و خیز پایین در ابتدا رده بندی می شوند که توسط گره های دارای یک نرخ افت و خیز بالاتر دنبال می شوند. براساس نتایج رده بندی، کارهای درون مجموعه ی فرایندها برنامه ریزی می شوند. شکل 9 فاش کننده ی مقایسه ی بین دقت برنامه ریزی STSFTR (1) کنونی (برنامه ریزی وظایف بر اساس مناسب ترین منبع تحمل خطا) و مدل پیشنهاد شده است. از شکل مشخص است که مدل پیشنهاد شده در برنامه ریزی کارها کاراتر است. این یک مدرک برای کارایی روش SAMA ی استنتاج شده است.

1. نتیجه گیری و کارهای آینده

تحمل خطا در تمام محیط های توزیع شده یک مساله ی حیاتی است. در اینجا، مساله تحمل خطا توسط شکست در انجام کار مورد اشاره قرار گرفته است. روش SAMA به صورت پویا شامل تخمین نرخ خطا، چک پوینت گذاری اولیه و ارزیابی پایداری با استفاده از محدودیت هایی مانند تعداد شکست، زمان طی شده، زمان مورد انتظار و نرخ بازیابی است. در ادامه، نرخ افت و خیز تعیین می شود و براساس آن رده بندی گره ها انجام می شود با هدف برنامه ریزی موثر. بنابراین کار پیشنهاد شده تحمل در برابر خطا را توسط اقتباس قوی از تناوب چک پوینت مطابق با تاریخچه ی قبلی از زمان اجرای کار و اطلاعات شکست، به دست می آورد و به صورت فراوانی سربار چک پوینت را کاهش می دهد و خروجی کلی را افزایش می دهد. هرچند، فرایند هزینه ی مهاجرت را کاهی می دهد، اعتمادپذیری گره های شبکه را اصلاح می کند و برنامه ریز کار ار هدایت می کند تا کارها را از مجموعه های فرایند به طرف گره ها به یک شکل ماهرانه تخصیص دهد.

با توجه به بهسازی های آینده، ما تصمیم داریم که بر روی در نظر گرفتن شکست های مختلف مانند، شکست های سخت افزاری، شکست های ارتباطی کار کنیم و یک چارچوب بهینه سازی شده ی برنامه ریزی را توسعه دهیم.