مقابله با افت کیفیت پیش‌بینی کننده وضعیت آتی کار در محیط گرید محاسباتی

رضا قائمی[۱]–حسین سلامی[۲]–مهرداد جلالی[۳]

(۱) استادیار - گروه مهندسی کامپیوتر، واحد قوچان، دانشگاه آزاد اسلامی، قوچان، ایران

(۲) کارشناس ارشد - گروه مهندسی کامپیوتر، موسسه آموزش عالی فردوس، مشهد، ایران

(۳) استادیار - گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

تاریخ دریافت: ۱۳۹۶/۶/۶ تاریخ پذیرش: ۱۳۹۷/۶/۲۳

خلاصه: محیط‌های پردازشی توزیع شده مانند گریدهای محاسباتی، یکی از مهمترین بسترها برای رفع نیازهای پردازشی کاربران می‌باشند. این محیط‌ها دارای توان بالقوه‌ای برای پاسخگویی به نیازهای کاربران هستند اما مشکلات خاص خود را نیز به همراه دارند که از آن جمله می‌توان به مسئله خرابی کارها اشاره نمود. تلاش‌های متعددی برای چیره شدن بر این مسئله انجام شده است که به طور کلی می‌توان آنها را به دو دسته روش‌های سمت منابع و روش‌های سمت کار تقسیم نمود. هر دو دسته روش‌های ذکر شده با هدف دنبال نمودن رویکرد پیشگیرانه در برابر خرابی‌ها، به پیش‌بینی وضعیت منابع و یا کارها نیازمند می‌باشند. با اینحال با توجه به پویایی این محیط‌ها، مدل‌های ایجادشده به سرعت اعتبار خود را از دست داده و لذا نمی‌توانند کمک موثری به روش‌های یاد شده نمایند. در این نوشتار ابتدا با شناسایی دلایل کاهش کیفیت پیش‌بینی‌کننده‌ها در محیط گرید، راهکاری به منظور مقابله با آن ارائه شده و سپس راهکار مورد نظر در زمینه مقابله با خرابی کارها به کار گرفته شده است. نتایج آزمایشات بر روی دو محیط آزمایشی AuverGrid و Grid5000 نشان داد روش پیشنهادی افزایش کیفیت به ترتیب به میزان ۰/۰۲ و ۰/۰۶ را در این دو محیط باعث می‌گردد.

کلمات کلیدی: پیش‌بینی وضعیت آتی کار، گرید محاسباتی، خرابی کار.

Coping With the Loss of Quality of Job Future Predictors in Grid Computing Environments

Reza Ghaemi[1] – Hossein Salami[2] – Mehrdad Jalali[3]

(1) Assistant Professor – Dept. of Computer Engineering, Quchan Branch, Islamic Azad University, Quchan, Iran

r.ghaemi@iauq.ac.ir

(2) MSc - Ferdows Higher Education Institute, Mashhad, Iran

hosein.salami@gmail.com

(3) Assistant Professor – Dept. of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

dr_mehrdadjalali@yahoo.com

Abstract: Distributed processing environments, such as grids, are one of the most important platforms for meeting the user's processing needs. These environments have the potential to meet the needs of users, but they also have their own problems, including the failure of the jobs. Several attempts have been made to overcome this problem, which in general can be divided into two categories of resource side methods and job side methods. All these methods need some kind of prediction of the resources or jobs status in order to pursue a proactive approach to failures. However, due to the dynamics of these environments, the developed models quickly lose their quality and thus can not effectively help with the methods mentioned. In this paper, first, by identifying the reasons for reducing the quality of predictors in the grid environment, a solution has been proposed to deal with it, and then the proposed solution has been applied in the context of job failures. The results of experiments on the two experimental environments of AuverGrid and Grid5000 showed that the proposed method would increase the quality by 0.02 and 0.06 respectively in these two environments.

Index Terms: Job futurity prediction, grid computing, job failure.

نویسنده مسئول: رضا قائمی، استادیار، گروه مهندسی کامپیوتر، واحد قوچان، دانشگاه آزاد اسلامی، قوچان، ایران، rezaghaemi73@gmail.com.

۱- مقدمه

گریدها با دربرگرفتن تعداد زیادی از منابع، پتانسیل پردازشی قابل توجهی را برای فعالیت‌هایی که نیازمندی‌های پردازشی با مقیاس بالا دارند، نوید می‌دهند. با این حال این بستر هنوز نتوانسته این رویا را به طور کامل محقق نماید. علاوه بر ویژگی‌های خاص این بستر مانند مقیاس بالا، پویایی، ناهمگونی منابع و سیاست‌ها در عرضه‌ی منابع، که سبب پیچیدگی مدیریت آن می‌گردد، وجود بعضی مسائل باعث استفاده‌ی غیر بهینه از منابع در این محیط می‌گردد. یکی از این مسائل، مسئله‌ی خرابی کارهاست. خرابی کارها در بسیاری از بسترهای محاسباتی وجود دارد، اما این پدیده در گریدهای محاسباتی به صورت قابل توجهی دارای نرخ بالاتری است. برای نمونه در جدول ۱، نرخ کارهای خراب‌شده در چند محیط مختلف آورده شده است.

Table(1): The percentage of failures in some grids

جدول (۱): درصد خرابی در برخی از گریدها

نام محیط	درصد خرابی کارها
AuverGrid	٪۷
Tera Grid	٪۱۰-٪۴۵
Grid3	٪۲۷
LCG	٪۲۵-٪۳۳

دلایل متعددی برای خرابی کارها توسط محققان مختلف گزارش شده است. یک کار ممکن است با ماشین مشخصی سازگاری نداشته باشد، بیشتر کارها نیازمند CPU و یا سیستم عامل خاصی می‌باشند. برخی دیگر از خرابی‌ها ممکن است به سبب فعالیت‌های مدیریتی رخ دهد. برای نمونه، دسترسی یک کاربر به منبعی که قبلاً توانایی دسترسی به آن را داشته است، سلب می شود [۱]. خرابی یک کار ممکن است به سبب وجود یک اشکال در خود برنامه (کار) ارسالی کاربر رخ دهد. همچنین ممکن است که خرابی کار به سبب خرابی منبعی (منابعی) رخ دهد که بر روی آن (ها) در حال اجرا بوده است [۲]. دسته‌بندی‌های متعددی برای خرابی‌ها در محیط گرید موجود است که نمونه‌هایی از آنها در [۳]، [۴] و [۵] ارائه شده‌اند.

برخی از خرابی کارها، به سبب از دسترس خارج شدن منابع اجرا کننده‌ی آنها رخ می‌دهند. خرابی های سخت‌افزاری، مانند خرابی پرداز شگر اصلی و یا دیسک سخت، خرابی‌های نرم‌افزاری، مانند بروز خطاهای مهلک در نرم افزارهای سیستمی و مدیریتی همچون سیستم عامل و یا بسته‌های مدیریتی منبع، خرابی‌هایی که به سبب عوامل محیطی رخ می دهند، مانند قطع برق، خارج شدن منبع از محیط به صلاحدید مالک آن، و یا قطعی ارتباط با منبع به سبب قطعی ارتباطات شبکه‌ای، از جمله دلایلی است که سبب می‌شوند تا کارهایی که بر روی منبع (منابع) در حال اجرا بوده‌اند، دچار خرابی گردند. به چنین خرابی‌هایی، عوامل سمت منبع گوییم. مقابله با خرابی‌هایی که به سبب عوامل سمت منبع (منابع) رخ می‌دهند. به عنوان مسئله‌ی ایجاد اطمینان شناخته می‌شود. تلاش‌هایی که به منظور فراهم آوردن اطمینان در محیط گرید صورت گرفته‌اند، اکثرا بدنبال روش‌هایی برای ایجاد قابلیت تحمل خرابی بوده است. قابلیت تحمل خرابی، به معنای فراهم آوردن شرایطی است که در آن، اجرای کارها علیرغم بروز خرابی ها ادامه می یابد [۶]. برای نمونه، تکنیک check-pointing یکی از مهمترین تکنیک‌هایی است که برای ایجاد تحمل خرابی در اکثر محیط‌ها مورد استفاده قرار می گیرد.

بسیاری از ناهنجاریها مربوط به توسعه ناکارآمد نرم افزارهاست که منجر به بروز نشت حافظه، نخ‌های پایان نیافته، عدم آزادسازی قفل‌ها و.. می‌شوند و بر طبق [۷] به طور میانگین ۴۰٪ از مجموع کل ناهنجارها در سیستم‌های محاسباتی را به خود اختصاص می‌دهند. از این نوع عوامل خرابی کارها که از کارافتادگی منابع در بروز آنها دخالتی ندارد موارد دیگری مانند تخطی کار از میزان سهمیه‌ی اعطایی منبع، که توسط سیاست‌های آن منبع مشخص می‌گردد، انتظار بسیار زیاد در صف مدیر بارِ کاری، اتمام اعتبارنامه‌ی کاربر و لغو کردن اجرای کار توسط کاربر ارسال کننده در حین اجرا را اشاره نمود [۳]. در ادامه‌ی این نوشتار، چنین عواملی از خرابی کار را، *عوامل سمت کار* می‌نامیم. یکی از صدمات اصلی خرابی هایی که به سبب عوامل سمت کار ایجاد می‌شوند، هدر دادن زمان منابعی است که برای اجرا به آنها اختصاص داده شده‌اند [۸]. علاوه براین نشان داده شده است که کارهای خراب شده به طور میانگین میزان مصرف بسیار بیشتری از منابع محاسباتی را نسبت به کارهایی که به صورت صحیحی اجرا شده‌اند دارند [۹]. پیامد اصلی هدر رفت منابع، ایجاد تاخیر در آغاز اجرای سایر کارهایی است که در سیستم حضور دارند (افزایش زمان انتظار کارها). اگرچه تحقیقات گسترده‌ای در مورد روش‌های مقابله با خرابی‌هایی از کارها که به سبب عوامل سمت منبع رخ داده‌اند، انجام شده است، اما در مورد عوامل سمت کار، تلاش‌های به مراتب کمتری صورت گرفته است.

به طور کلی دو رویکرد برای مقابله با پدیده خرابی وجود دارد، (۱) واکنشی و (۲) پیشگیرانه. روش‌های واکنشی (مانند Checkpointing) عمل خاصی برای جلوگیری از بروز خرابی صورت نمی‌دهند بلکه پس از وقوع آن، اعمال مورد نظر خود را انجام می‌دهند، این در حالی است که در روش‌های پیشگیرانه، هدف، جلوگیری از بروز خرابی است. یکی از الزامات اصلی برای اکثر روش‌های پیشگیرانه، پیش‌بینی شرایط آتی (در مورد منابع و یا کارها) است. با این حال با توجه به طبیعت پویا و در حال تغییر گرید، مدلهایی که برای پیش‌بینی رفتارهای آن ایجاد می‌شوند به سرعت کارآیی خود را از دست داده و بلااستفاده می‌شوند. این امر سبب الزام به افزودن قابلیت تطبیق‌پذیری به چنین مدلهایی می‌گردد. در این نوشتار این هدف در قالب تطبیق‌پذیری به پیش‌بینی کننده وضعیت کار انجام پذیرفته است تا بدین طریق بر یکی از مشکلات اصلی در بستر گرید که میزان بالای خرابی کارها می‌باشد، فایق آییم. برای این منظور در ابتدا به شناسایی دلایل افت کیفیت پیش‌بینی کننده پرداخته و سپس راهکاری برای حفظ کیفیت پیش‌بینی کننده وضعیت کار ارائه شده است. در بخش بعد در ابتدا برخی روش‌های مورد استفاده برای

ایجاد پیش‌بینی کننده‌ها معرفی شده‌اند. در بخش سوم، به جزییات روش پیشنهادی خواهیم پرداخت و در بخش چهارم نتایج آزمایشات آن را بیان خواهیم نمود. نهایتاً در بخش انتهایی نتیجه‌گیری و کارهای آتی بیان خواهند شد.

۲- کارهای پیشین

تاکنون روش‌های متعددی برای پیش‌بینی وقایع پیشنهاد شده‌اند. این روش‌ها را می‌توان به طور کلی به دو دسته‌ی روش‌های داده‌گرا و مبتنی بر مدل تقسیم‌بندی نمود [۱۰]. روش‌های مبتنی بر مدل، مدلی تحلیلی و یا آماری را از سیستم استخراج نموده و سپس در صورت مشاهده‌ی انحراف از مدل، هشداری را صادر خواهند نمود. با توجه به مدل استخراج شده، می‌توان احتمال واقعه‌ی مورد نظر را در زمان خاصی در آینده تعیین نمود. در مورد خرابی منابع، مدل‌های آماری رایج مورد استفاده عبارتند از Weibull، Poisson و Logarithmic [۱۱]، و از جمله مدل‌های تحلیلی می‌توان به مدل مارکوف اشاره نمود. یکی از معایب این روش‌ها، عدم کارآیی آنها در محیط‌های بزرگ می‌باشد. دلیل این امر، پیچیدگی بالای محیط‌های با مقیاس بزرگ است.

در طرف مقابل، روش‌های داده‌گرا، با اعمال تکنیک‌های داده کاوی بر روی داده‌هایی که از گذشته‌ی سیستم به دست آمده است، الگوهای خرابی را فراگرفته و آنها را، بدون اینکه از قبل مدلی ایجاد کرده باشند، دسته‌بندی می‌نمایند. این روش‌ها سپس رفتار فعلی سیستم را بررسی می‌نمایند و در صورت مواجهه با رفتارهایی که مشابه با الگوهای فراگرفته‌شان برای خرابی باشد، اعلام هشدار خواهند نمود.

با توجه به رشد گرید و پیچیده‌تر شدن آن و از طرفی به علت طبیعت پویایی که در این محیط وجود دارد، رفتارهای آن کمتر در قالب مدل‌های ریاضی قابل بیان هستند [۱۰][۸][۱] و لذا روش‌های داده کاوی، در سال‌های اخیر، محبوبیت بالاتری میان پژوهشگران پیدا کرده‌اند، از اینرو در بخش بعدی به ذکر برخی پژوهش‌ها از کاربرد روش‌های داده کاوی در سیستم‌های توزیع شده و به خصوص گرید، پرداخته شده است.

۲-۱- استفاده از روش‌های داده‌کاوی برای انجام پیش‌بینی

نویسندگان در [۸]، داده کاوی را با توجه به بزرگ شدن و پیچیده شدن گرید، روش مناسب و کارآمدی جهت تشخیص و پیش‌بینی خرابی‌ها دانسته و تشخیص زود هنگام خطاها و نیز اجازه به کاربران در اتخاذ تصمیم‌های صحیح مبتنی بر دانش به دست آمده از کاوش داده‌ها را از جمله مزایای استفاده از متدهای داده کاوی عنوان نمودند.

شبکه‌های بیزین از جمله متدهایی است که استفاده از آن همواره مورد توجه محققان بوده است. به عنوان مثال آقای سانگ فو و همکارش در تحقیقشان [۱۲] با بهره‌گیری از شبکه‌های بیزین، به دنبال پیش‌بینی خطاهای سخت‌افزاری (همانند حافظه و پردازنده) و نرم‌افزاری (همانند سیستم عامل و سیستم فایل) در کلاسترها بوده‌اند. اطلاعات مربوط به تخصیص برنامه‌ها و داده‌های زمانی و مکانی سیستم، پارامترهای دخیل در مدل ساخته شده را تشکیل می‌دهند.

در [۱۳] از درخت‌های تصمیم، شبکه‌های بیزین، شبکه‌های عصبی و زنجیر مارکوف به عنوان تکنیک‌های مناسب جهت تعلیم و ایجاد مدل رفتاری سیستم نام برده شده است. نویسندگان این مقاله با ارائه‌ی یک سرویس و استفاده از تکنیک‌هایی همچون رأی‌گیری، ترکیب کردن و داوری، سعی کرده‌اند تا از ترکیب این متدها برای تشخیص و پیش بینی خرابی‌ها بهره بگیرند.

در [۱]، نویسندگان نشان دادند که می‌توان با استفاده از تکنیک‌های داده کاوی مانند درخت تصمیم و یا روش مبتنی بر قاعده، و اعمال آنها بر روی ویژگی‌های منابعی که اجرای کار بر روی آنها انجام شده است، ویژگی‌های مرتبط با موفقیت و یا خرابی کار را یافت. از ویژگی‌های یافت شده می‌توان به عنوان معیاری در انتخاب منابع مناسب، و یا تغییر پیکربندی منابع استفاده نمود. برای اثبات کارآیی روش پیشنهادی، نویسندگان در [۱۴] بر روی یک محیط واقعی، راهکار خود را مورد آزمایش قرار دادند. آنها از تکنیک درخت تصمیم‌گیریِ خاصی با نام HDDT برای کلاس‌بندی خرابی‌های رخ داده در کارها استفاده نمودند. با اعمال تکنیک فوق بر روی بارکاری‌های تک کاربره، نشان داده شد که می‌توان از درخت تصمیم ایجاد شده، نقاطی را یافت که می‌تواند در عیب‌یابی مشکل رخ داده موثر باشد. این نویسندگان در [۱۵]، همین روش را بر روی ویژگی‌ها (آرگومان‌ها)ی کار نیز اعمال نمودند تا از این طریق، تاثیر این ویژگی‌ها را در بروز خرابی بررسی نمایند.

سعادتفر و همکارانش در [۱۶] با تحلیل نمودن پرونده‌های ثبت وقایع AuverGrid به بررسی میان ویژگی‌های بارکاری و خرابی کارها پرداختند و سپس با بکارگیری شبکه‌های بیزین، توانستند با دقت ۹۶٪ خرابی کارها را در این محیط پیش‌بینی نمایند.

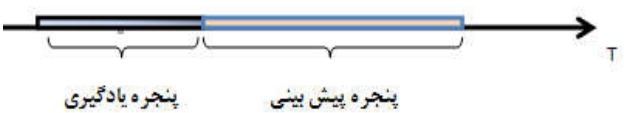
برخلاف روش‌های فوق، که عیب‌یابی را به صورت دستی صورت می‌دهند، لی و همکارانش [۳]، مقابله با خرابی‌ها را در سطح دلال منبع پیشنهاد نمود. ایشان با انجام تحلیل‌های آماری بر روی خرابی‌های رخ داده در گرید LCG، همبستگی‌های زمانی و فضایی را در بین آن‌ها گزارش نمودند. در واقع آنها نشان دادند که بیشتر خرابی‌ها به صورت انفجاری و در بازه‌های کوتاه مدت و در بعضی منابع خاص رخ می‌دهند. همچنین آنها نشان دادند که کاربران بعضی از سازمان‌های مجازی، سهم بیشتری را در خرابی‌های رخ داده، دارا می‌باشند. با توجه به این مشاهدات، دو استراتژی برای مقابله با این خرابی‌ها در سطح دلال منبع پیشنهاد گردید. در استراتژی اول، تاریخچه‌ی خرابی کارها توسط دلال منبع پیگیری می‌شود. این تاریخچه، برای هر سازمان مجازی و بر مبنای هر منبع صورت می‌پذیرد. از تاریخچه‌ی به دست آمده می‌توان استفاده نمود تا کارهای ارسالی از طرف کاربران یک سازمان مجازی را به سمت منابعی گسیل داد که در گذشته، رفتار بهتری (خرابی‌های کمتر) در قبال کارهای سازمان مجازی داشته‌اند. در استراتژی دوم، برای هر منبع، پارامتری با نام ظرفیت مؤثر تعریف شده است. ظرفیت موثر، حداکثر تعداد کارهایی از یک سازمان مجازی است که می‌تواند در یک منبع به طور همزمان در حال اجرا باشند. ایده‌ی این استراتژی این است که ممکن است در بعضی از منابع، برای هر

سـازمان مجازی، مقدار حداکثری برای تعداد کارهای در حال اجرا تعریف شده باشد و در صورتی که بیش از این مقدار، کار به منبع ارسال گردد، توسط منبع پذیرفته نشود.

روش‌های داده گرای فوق براساس داده‌هایی که برای تعلیم در اختیارشان قرار می‌گیرد (که نشاندهنده رفتار گذشته سیستم است) الگوهای خرابی را استخراج می‌نمایند، لذا در صورتی که شرایط فعلی محیط، متفاوت از شرایط دوره زمانی مورد استفاده برای آموزش باشد، الگوهای فراگرفته شده نامعتبر خواهند بود. این پدیده در مورد بستر گرید با توجه به دارا بودن طبیعت پویا و متغیر آن محتمل‌ترست. بعبارت دیگر الگوهای فراگرفته شده توسط روش‌های فوق در محیط گرید، به سرعت اعتبار خود را از دست داده و در عمل نمی‌توانند به شکل مفیدی روش‌های پیشگیرانه را یاری دهند. علیرغم بررسی جوانب متعدد در تحقیقات پیشین، هیچ یک از آنها به این مسئله توجهی نداشته‌اند لذا در این نوشتار، با بررسی انواع تغییرات در گرید و نحوه تاثیرگذاری آنها بر کیفیت پیش بینی کننده، چهارچوبی پیشنهاد شده است تا با افزودن قابلیت تطبیق‌پذیری به پیش‌بینی‌کننده، سعی در کاهش میزان تاثیرگذاری تغییرات محیطی در کیفیت پیش‌بینی کننده فراهم آید. در بخش بعدی، ابتدا دلایل موثر بر کاهش کیفیت پیش‌بینی کننده بررسی شده‌اند و سپس در بخش ۴، ساختار پیشنهادی سیستم ارائه شده است.

۳- بررسی دلایل افت کیفیت

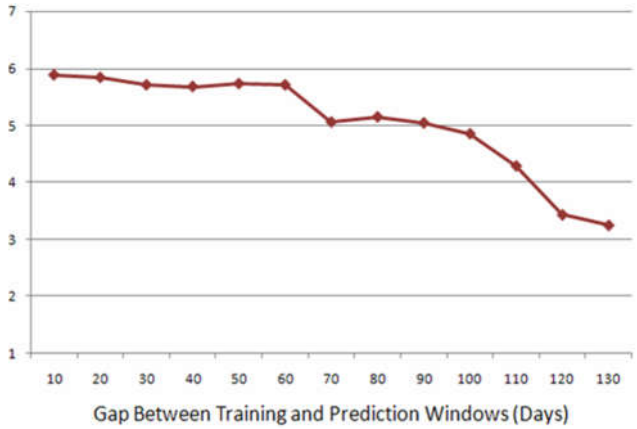
همانگونه که اشاره شد، روش‌های داده‌کاوی براساس داده‌هایی که برای تعلیم در اختیارشان قرار می‌گیرد الگوهایی را استخراج و براساس آنها مدلی را ایجاد می‌نمایند. در صورتی که داده‌های مورد استفاده برای آموزش، منعکس‌کننده شرایط فعلی محیط نباشند الگوهای فراگرفته شده نامعتبر خواهند بود و در نتیجه نتایج حاصل از مدل‌های ایجاد شده، کیفیت مناسب را نخواهند داشت. به منظور بیان دقیقتر این پدیده، مفهوم پنجره یادگیری (آموزش) و پنجره پیش‌بینی (آزمایش) را مورد استفاده قرار می‌دهیم. منظور از پنجره یادگیری، دنباله‌ای از N رکورد پیاپی $R_i\ (i = 1,2,\dots,N)$ از ویژگی‌های کارهای ورودی و نتیجه پایان کارهاست. مدل پیش‌بینی کننده با کاوش رکوردهای موجود در پنجره یادگیری ایجاد می‌گردد و سپس از آن برای پیش‌بینی وضعیت کارها در پنجره پیش‌بینی استفاده می‌نماییم. در شکل ۱، رابطه‌ی بین این دو مفهوم به تصویر کشیده شده است.



شکل (۱): رابطه بین پنجره یادگیری و پیش‌بینی

Fig. (1): The relationship between learning and prediction window

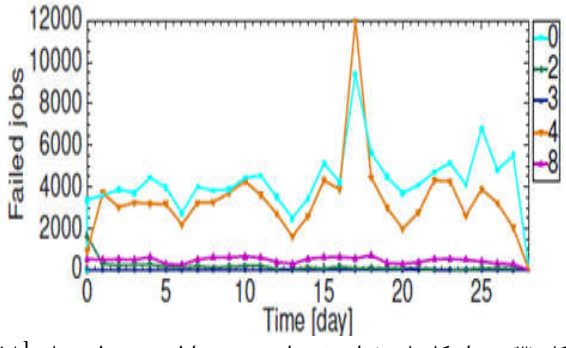
در صورتی که در پنجره آزمایش، شرایط تقریباً مشابه با پنجره یادگیری باشد، پیش‌بینی کننده قادر به انجام پیش‌بینی‌هایی با کیفیت بالا خواهد بود. با این حال بروز تغییرات در شرایط محیطی اجتناب ناپذیر است. این تغییرات ممکن است به صورت تدریجی در محیط رخ دهند و لذا منجر به بروز پدیده *سالخوردگی* برای پیش‌بینی کننده گردد. در [۱۷] بمنظور بررسی این پدیده، کیفیت پیش‌بینی کننده با به دست آوردن دقت آن در پنجره‌های پیش‌بینی که به تدریج از پنجره یادگیری، فاصله داده می‌شدند سنجش شد. مطابق انتظار، همانگونه که در شکل ۲ نیز نمایش داده شده است، دور شدن از پنجره یادگیری منجر به کاهش کیفیت پیش‌بینی کننده می‌گردد.



شکل (۲):کاهش کیفیت پیش‌بینی کننده در اثر سالخوردگی[۱۷]

Fig. (2): Reduced predictor quality due to aging [17]

علاوه بر پدیده سالخوردگی، تغییرات لحظه‌ای شدید نیز می‌توانند منجر به کاهش کیفیت پیش‌بینی کننده شود. در [۳] با انجام تحلیل‌های آماری بر روی خرابی‌های رخ داده در گرید LCG، همبستگی‌های زمانی و فضایی گزارش شد. نویسندگان نشان دادند که بیشتر خرابی‌ها به صورت انفجاری و در بازه‌های کوتاه مدت و در بعضی منابع خاص رخ می‌دهند. در [۱۸] نیز افزایش قابل ملاحظه کارهای خراب شده در بازه زمانی کوتاه گزارش شده است. در شکل ۳، شماره‌های ۰، ۲، ۳، ۴ و ۸، دسته بندی های مختلف از کارها را براساس اولویت آنها نشان می‌دهد. همانگونه که مشاهده می شود، در بازه زمانی روز ۱۶ تا ۱۹، تعداد زیادی از کارهای با اولویت ۰ و ۴ به صورت ناگهانی دچار اجرای ناموفق شده‌اند.

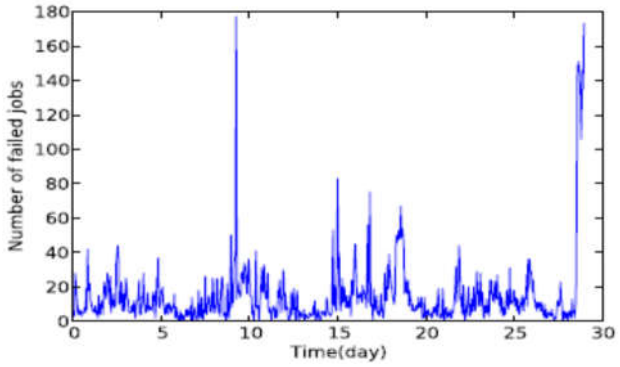


شکل (۳): تعداد کارهای خراب شده از هر دسته اولویت در طی زمان [۱۸]

Fig. (3): The number of failed jobs from each priority category over time [18]

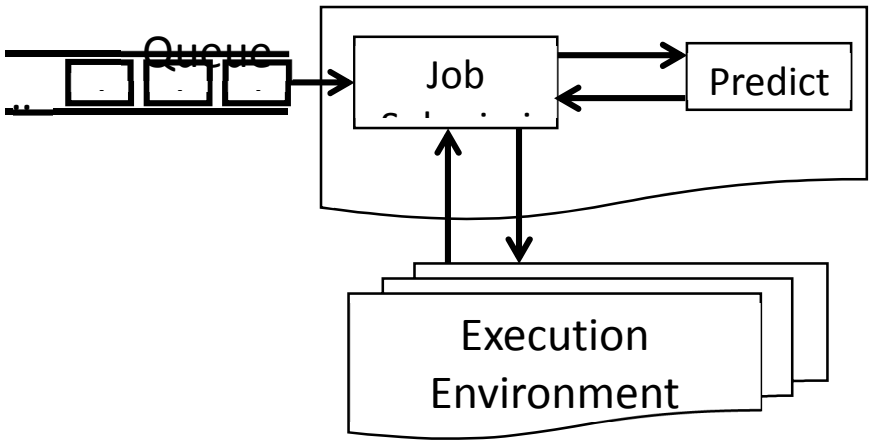
در [۱۹] نویسندگان به تحلیل پرونده‌های ثبت بارکاری کلاسترهای Google که شامل ۱۲۰۰۰ گره در طول دوره یک ماهه بوده است، پرداختند. در شکل (۴) تعداد کارهای خراب شده در طی دوره یکماهه بررسی شده نمایش داده شده است. همانگونه که مشاهده می‌شود، وقوع خرابی‌ها در برخی از بازه‌های زمانی، به صورت ناگهانی افزایش یافته است.

این نوع تغییرات لحظه‌ای غالباً سبب بی‌اعتبار شدن مدل‌های فراگرفته شده توسط پیش‌بینی‌کننده از پنجره یادگیری که چنین تغییرات لحظه‌ای را دربرنگرفته‌اند خواهد شد. طبیعتاً فارغ از اینکه کدامیک از دو عامل فوق سبب کاهش کیفیت پیش‌بینی کننده شود، می‌توان با انجام بازآموزی با استفاده از داده‌هایی که منعکس کننده شرایط فعلی، و احتمالاً آینده نزدیک محیط می‌باشند، کیفیت پیش‌بینی کننده را افزایش داد. با اینحال تعیین اینکه این داده‌ها را در چه بازه‌ای بایستی جستجو نمود، مسئله مهمی است که در بخش بعد بدان خواهیم پرداخت.



شکل (۴): تعداد کارهای خراب شده در گره‌های مربوط به کلاستر Google [۱۹]

Fig. (4): The number of failed jobs in Google Cluster nodes [19]



شکل (۵): ساختار سیستم پیشنهادی

Fig. (5): The proposed system structure

۴- تشریح عملکرد روش پیشنهادی

در این بخش با ارائه و تشریح ساختار سیستم پیشنهادی، به نحوه تعیین بازه زمانی به منظور انتخاب داده‌های مناسب برای تعلیم پیش‌بینی کننده خواهیم پرداخت. در شکل (۵) ساختار سیستم پیشنهادی نمایش داده شده است که مشابه با ساختار ارائه شده در [۱۷]، علاوه بر محیط اجرایی و صف کارهای ارسالی، شامل مدیر ارسال کار و پیش بینی‌کننده است.

پیش‌بینی کننده به صورت دوره‌ای براساس ویژگیهای نگهداری شده از کارهای ارسالی به همراه نتایج پایانی اجرای آنها و با پیروی از روش پیشنهادی در [۱۶] که با بکارگیری شبکه بیزین به تعلیم پیش‌بینی‌کننده پرداخته‌اند، ایجاد (و بازآموزی) می‌شود. ویژگیهای مورد استفاده برای تعلیم پیش‌بینی‌کننده در [۱۶] علاوه بر ویژگیهای خام کارها، برخی ویژگیهای ترکیبی را نیز در بر می‌گیرد. لازم به ذکرست که با توجه به اینکه تمرکز اصلی در این نوشتار، افزودن قابلیت تطبیق پذیری به پیش‌بینی‌کننده است، می‌توان از روش‌های رایج دیگری نیز که برای تعلیم پیش‌بینی‌کننده وجود دارد، استفاده نمود.

مشابه با [۱۷] مدیر ارسال کار، تصمیم‌گیری در مورد پذیرش و یا عدم پذیرش کار دریافتی از صف کارها را برعهده دارد. این تصمیم گیری با توجه به احتمال خرابی کار که به وسیله رایزنی با پیش‌بینی کننده حاصل می‌گردد، صورت می‌گیرد. علاوه بر این، مدیر ارسال کار وظیفه نظارت بر عملکرد پیش بینی‌کننده و همچنین تصمیم گیری در مورد انجام بازآموزی پیش بینی کننده را نیز بر عهده دارد. برای این منظور، در پایان هر پنجره آزمایش، اندازه‌گیری میزان کیفیت پیش بینی کننده انجام خواهد گرفت.

طبیعتاً بدون توجه به عاملی که سبب کاهش کیفیت پیش‌بینی کننده شده است، می‌توان با انجام بازآموزی با استفاده از داده‌های مناسب، کیفیت پیش‌بینی کننده را برای پنجره‌های آزمایش بعدی افزایش داد. این داده‌ها در واقع پنجره یادگیری را برای بازه زمانی بعدی تشکیل خواهند داد با اینحال تعیین بازه انتخاب آنها یکی از مسایلی است که نیازمند توجه بیشتری است. همانگونه که در بخش قبل مشاهده شد تغییرات محیطی به دو صورت تدریجی و یا ناگهانی بروز می‌نمایند و بازه زمانی برای انتخاب داده‌های مناسب در هر یک از این دو حالت با دیگری

متفاوت است. اگر تغییرات محیطی به صورت تدریجی رخ دهند، بازه زمانی طولانی‌تری را دربرخواهند گرفت (سالخوردگی) و لذا بمنظور پوشش آنها بایستی اندازه بزرگتری برای پنجره یادگیری در نظر گرفته شود. در طرف مقابل، تغییرات ناگهانی و لحظه‌ای، دوره حیات کوتاهتری دارند و یادگیری الگوهای آنها توسط پیش‌بینی کننده نیازمند بازه یادگیری کوچکتری است.

به منظور بیان دقیقتر روش پیشنهادی، عملکرد سیستم به بازه (پنجره)های زمانی مساوی با اندازه d تقسیم می‌شود. هر یک از این پنجره‌ها به ترتیب شماره‌گذاری می‌شوند بگونه‌ای که اولین پنجره دارای شماره 0، دومین پنجره دارای شماره 1 و... می‌باشد. در هر یک از پنجره‌ها، تعدادی کار توسط مدیر ارسال کار، بررسی و در مورد پذیرش و یا عدم پذیرش آنها تصمیم گیری می‌شود. در پایان هر پنجره، در صورتی که دقت پیش بینی کننده کمتر از حد آستانه α شود، مدیر ارسال کار بایستی بمنظور بازآموزی پیش بینی کننده اقدام نماید. دقت پیش بینی کننده در یک پنجره (W_i)، با توجه به درستی پیش بینی‌های صورت گرفته برای کارهای بررسی شده در همان پنجره مشخص می‌شود. به صورت دقیقتر فرض کنید که p_j $(0 \le p_j \le 1)$ نمایانگر احتمال خرابی کار j باشد که توسط پیش بینی کننده در اختیار مدیر ارسال کار قرار داده شده است. میزان دقت پیش بینی کننده در پیش بینی وضعیت اجرای این کار را با A_j نشان می‌دهیم و آنرا به صورت زیر محاسبه می‌نماییم.

$$A_j = \begin{cases} 1 - p_j & if\ job\ j\ succeeded \\ p_j & if\ job\ j\ failed \end{cases} \quad (۱)$$

لذا دقت پیش‌بینی کننده برای پنجره W_i به صورت زیر محاسبه می‌شود.

$$\frac{\sum_{j \in W_i} A_j}{|W_i|} \quad (۲)$$

در رابطه فوق، $|W_i|$ تعداد کارهای موجود در پنجره W_i را مشخص می‌نماید.

در صورتی که دقت پیش‌بینی کننده کمتر از α باشد، مدیر ارسال کار بازه زمانی برای بازآموزی را که در واقع تعداد پنجره‌های مورد استفاده برای بازآموزی می‌باشد را از رابطه (۳) محاسبه می‌نماید:

$$m \times (c - k) \quad (۳)$$

در رابطه فوق k، شماره آخرین پنجره‌ای است که مدیر ارسال کار اقدام به بازآموزی پیش‌بینی کننده کرده است. پارامتر c، شماره پنجره فعلی را مشخص می‌کند و m پارامتری بمنظور تعیین میزان دخیل نمودن گذشته سیستم در آموزش پیش‌بینی کننده است. همانگونه که قبلا اشاره شد، تغییرات ناگهانی در بازه کوتاهی رخ می‌دهند و لذا بمنظور پوشش آنها نیاز به بازه کوتاهتری است در حالی که تغییرت تدریجی پوشش زمانی طولانی‌تری را نیاز دارند. واضح است که این مشاهده با استفاده از رابطه (۳) در نظر گرفته شده است بگونه‌ای که هر چقدر تغییرات دقت پیش‌بینی‌کننده شیب تندتری داشته باشند، مدیر ارسال کار بازه کوتاهتری را برای بازآموزی آن استفاده خواهد نمود، اما در صورتی که تغییرات به صورت تدریجی رخ داده باشند، بازه طولانی‌تری برای انجام بازآموزی پیش‌بینی کننده مورد استفاده قرار می‌گیرد. در بخش بعد با معرفی دیتاست‌های مورد استفاده، به بیان نتایج آزمایشات پرداخته شده است.

۵- آزمایشات و نتایج

در این بخش به بیان جزییات پیاده‌سازی روش پیشنهادی، معرفی دیتاست‌های مورد استفاده برای آزمایش‌های، نمایش و تحلیل آنها خواهیم پرداخت.

۵-۱- دیتاست مورد استفاده

در این نوشتار از مجموعه داده‌هایی از آرشیو بارکاری گرید [۲۰] که مربوط به سه محیط AuverGrid، DAS2 و Grid5000 است استفاده نموده‌ایم. این آرشیو، پرونده‌های ثبت وقایع مختلفی از گریدهای شناخته شده با محتوای بیش از ۲۰۰۰ کاربر که دارای حدود ۷میلیون کار در بازه ۱۳ ساله است و شامل ۱۰۰۰۰ منبع می‌باشند را جمع‌آوری نموده است. منابع داده‌ای این پرونده‌های ثبت وقایع از مدیران منابع محلی (مانند PBS)، مدیران منابع گرید (مانند Globus GRAM) تا مدیران منابع سمت کاربر و سازمان مجازی (مانند Condor Schedd) گسترده شده است.

محیط DAS-2 که یکی از محیطهای انتخابی در این نوشتار است، سیستمی توزیع شده شامل ۴۰۰ پردازشگر که در ۵ دانشگاه از کشور هلند قرار گرفته است می‌باشد. این محیط، بستری آزمایشی برای اهداف تحقیقاتی است که بارکاری آن از انواع مختلفی از برنامه‌های کاربردی شامل کارهای تک پردازشگری تا کارهای Grid MPI یا IBIS تشکیل شده است. Grid'5000، بستر گرید تجربی شامل ۹ سایت توزیع شده جغرافیایی است که در فرانسه قرار دارد. هر سایت متشکل از یک یا چندین کلاستر با مجموع ۱۵ کلاستر است. پرونده‌های ثبت وقایع در این بستر از شروع پروژه Grid'5000 تا نوامبر ۲۰۰۶ به وسیله تمامی زمانبندهای دسته‌ای (OAR) که Grid5000 را مدیریت می‌نمایند ثبت شده‌اند. پرونده‌های ثبت وقایع محیط سوم مربوط به AuverGrid است. AuverGrid گریدی چندسایته است که بخشی از پروژه EGEE می‌باشد. این گرید میان افزار LCG را بعنوان زیرساخت گرید بکارگرفته است. پرونده‌های ثبت وقایع ثبت شده مربوط به این محیط توسط مدیران منابع از ۵ کلاستر حاضر در AuverGrid جمع‌آوری شده است.

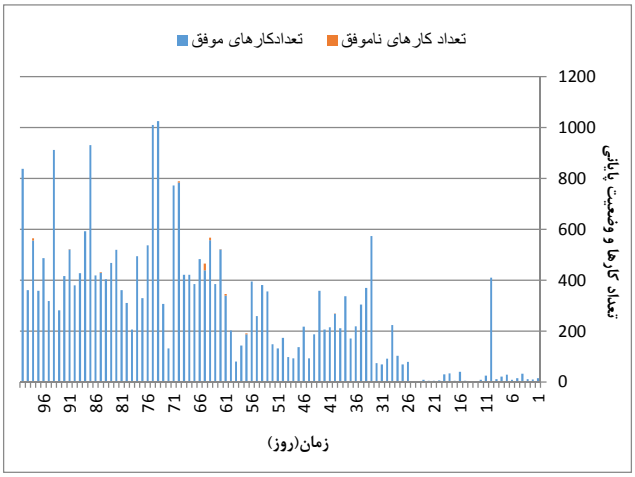
در آزمایشهای انجام گرفته، تنها به داده‌های مربوط به ۱۰۰ روز ابتدایی این سه محیط اکتفا نموده‌ایم. در جدول (۲) برخی ویژگی‌های داده‌های مربوط به کارهای ارسالی به این سه محیط نمایش داده شده است. همانگونه که مشاهده می‌شود، نسبت کارهای با اجرای ناموفق در DAS2 بسیار کم، در AuverGrid نسبتاً کم و در سیستم Grid5000 نسبتاً زیاد می‌باشد (بیش از نیمی از کارهای ارسالی شاهد اجرای ناموفق بوده‌اند).

Table (2): Some specifications of the dataset being used

جدول (۲): برخی مشخصات دیتاست‌های مورد استفاده

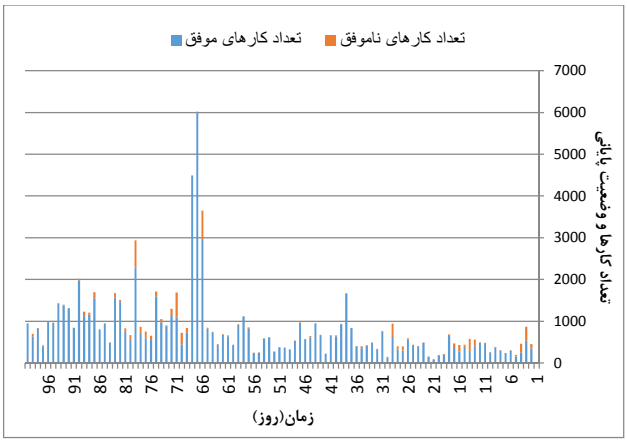
	تعداد کل کارها (۱۰۰روز ابتدایی)	تعداد کارهای با اجرای ناموفق	درصد کارهای با اجرای ناموفق	تعداد کل روزها
DAS2	۲۷۸۴۲	۶۶	۰/۰۰۲	۶۲۰ روز
AuverGrid	۸۴۳۴۰	۶۳۷۱	۰/۰۷۵	۳۶۱ روز
Grid5000	۵۶۴۶	۲۹۰۸	۰/۵۱	۸۵۰ روز

در نمودارهای (۶) تا (۸)، توزیع زمانی کارهای ارسالی این ۳ سیستم به همراه نسبت اجراهای ناموفق در ۱۰۰روز ابتدایی آنها نمایش داده شده است.



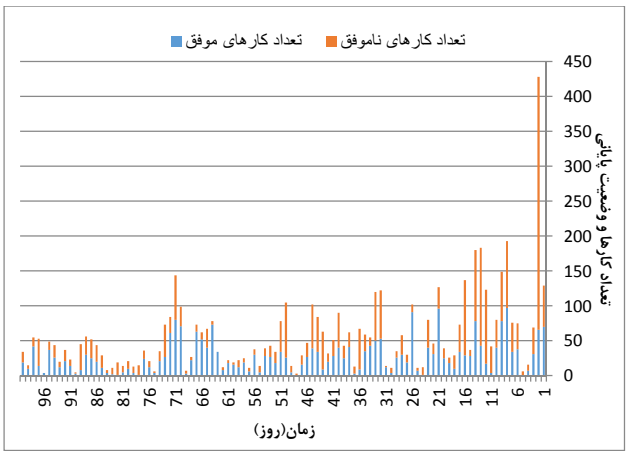
شکل (۶): توزیع زمانی کارهای ارسالی به DAS2

Fig. (6): Distribution of job submitted to DAS2



شکل (۷): توزیع زمانی کارهای ارسالی به AuverGrid

Fig. (7): Distribution of job submitted to AuverGrid

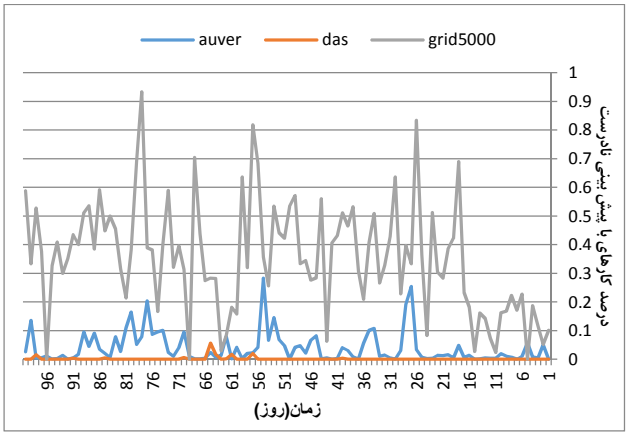


شکل (۸): توزیع زمانی کارهای ارسالی به Grid5000

Fig. (8): Distribution of job submitted to Grid5000

۵-۲- بررسی عملکرد پیش‌بینی کننده ایستا

در ابتدا به بررسی عملکرد پیش‌بینی کننده‌ای که به صورت ایستا تعلیم دیده است می‌پردازیم. بدین منظور پیش‌بینی کننده با استفاده از داده‌های ۱۵ روز ابتدایی هر یک از سه محیط DAS2، AuverGrid و Grid5000 تعلیم داده شده است و سپس وضعیت پایانی کارهایی که در کل بازه ۱۰۰ روزه این سه محیط قرار دارند تعیین شد. در صورتی که پیش‌بینی کننده، احتمال خرابی یک کار را بیش از ۵۰ درصد تعیین کند، مدیر ارسال کار آن را ناموفق و در غیر این صورت موفق در نظر گرفته است. درصد پیش‌بینی‌های نادرست توسط پیش بینی کننده حاصل در سه محیط فوق به ترتیب برابر با ۰/۰۰۱۴، ۰/۰۴ و ۰/۳۵۳ به دست آمد. در نمودار زیر نیز توزیع زمانی این پیش‌بینی‌های نادرست در طی ۱۰۰ روز مورد بررسی در سه محیط به نمایش گذاشته شده است.



شکل (۹): توزیع زمانی میزان پیش‌بینی‌های نادرست توسط پیش‌بینی‌کننده ایستا

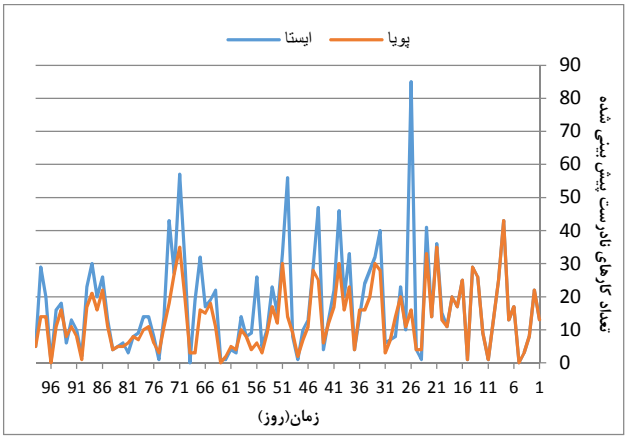
Fig. (9): Time Distribution of incorrect predictions by static predictor

طبیعی است که با توجه به آموزش پیش‌بینی کننده براساس داده‌های ۱۵روز ابتدایی، در هر سه محیط، بهترین عملکرد به دست آمده، در این بازه رخ داده باشد اما پس از این بازه، به تدریج شاهد کاهش کیفیت می‌باشیم. همانگونه که مشاهده می‌شود در محیط DAS2 تغییر خاصی

در میزان کیفیت پیش‌بینی‌کننده پس از بازه اولیه حاصل نمی‌گردد. این امر نشان‌دهنده‌ی پایداری نسبی شرایط در این محیط است که سبب معتبر باقی ماندن مدلی شده که براساس داده‌های اولیه ایجاد گردیده است. در دو محیط AuverGrid و بخصوص Grid5000 پیش‌بینی‌های نادرست پس از بازه ۱۵ روز اول به وضوح نمایان است که نشاندهنده‌ی این است که در آنها پویایی بسیار بیشتری در شرایط نسبت به محیط DAS2 وجود دارد. یکی دیگر از نکات قابل مشاهده در این نمودار، عدم یکنوایی کاهش کیفیت است بگونه‌ای که در برخی بازه‌های زمانی، عملکرد پیش‌بینی‌کننده تقریباً به خوبی بازه‌ی اولیه محیط (بازه‌ای که براساس آن تعلیم دیده است) بوده است. این مشاهده، تاییدی بر این موضوع است که برخی شرایط به صورت دوره‌ای در طی حیات سیستم رخ خواهند داد. برای مثال همانگونه که در شکل فوق مشاهده می‌شود در محیط AuverGrid در بازه روز ۲۱ تا ۲۵، ۶۶ تا ۶۹ و ۹۰ تا ۹۵ و در محیط Grid5000 در برخی روزها (نه به صورت پیوسته) مانند روز ۲۴، ۴۳، ۵۹ تا ۶۴ و ۹۵ پیش‌بینی‌کننده کیفیتی مشابه با بازه ۱۵ روزه اول را به نمایش گذاشته است.

۵-۳- بررسی عملکرد پیش‌بینی‌کننده پویا

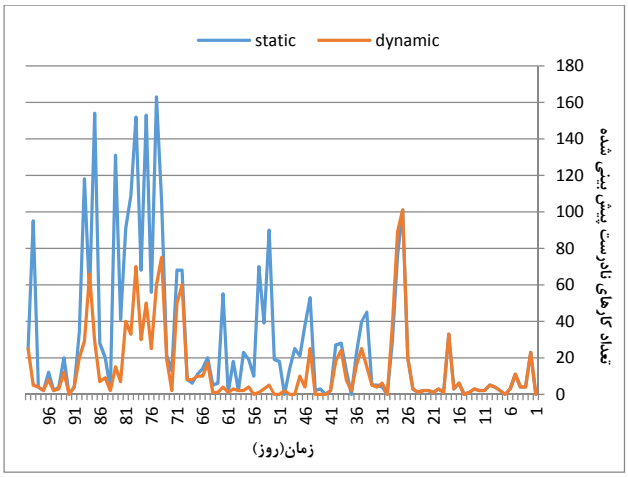
در ادامه به بررسی عملکرد سیستم که با پیروی از روش تشریح شده در بخش قبل ایجاد شده است پرداخته‌ایم. بیان این نکته الزامی است که اندازه پنجره را در آزمایشات مورد استفاده به صورت تجربی برابر با یک روز (d=1) و همچنین مقدار پارامتر m (میزان دخیل نمودن گذشته سیستم در انجام بازآموزی) را برابر با ۱۵ در نظر گرفتیم. پس از انجام آزمایشات، در مورد محیط DAS هیچگونه آموزش مجددی صورت نگرفت که با توجه به پایداری آن دور از انتظار نبود. در محیط AuverGrid و همچنین Grid5000 تمامی دفعاتی که کاهش کیفیت مشاهده شد به سبب تغییرات لحظه‌ای بوده است.



شکل (۱۰): توزیع زمانی تعداد کارهای نادرست پیش بینی شده در محیط Grid5000 توسط پیش‌بینی‌کننده ایستا و پویا

Fig. (10): Time Distribution of the number of incorrect prediction in the Grid5000 environment by static and dynamic predictors.

نمودار نمایش داده شده در شکل ۱۰ نتیجه اجرای روش پیشنهادی بر روی محیط Grid5000 را نشان می‌دهد که بهبود حاصل شده در عملکرد پیش‌بینی‌کننده به سبب انجام بازآموزی در ۲۰ نقطه مختلف به دست آمده است. در محیط AuverGrid با توجه به پایداری بیشتری که نسبت به محیط Grid5000 دارد، تعداد بازآموزی‌ها کمتر و محدود به سه نقطه ۲۷، ۷۵ و ۸۹ می‌باشد. به طور کلی میزان افزایش کیفیت در محیط Grid5000 در حالت پویا نزدیک به ۶٪ بیشتر از حالت ایستا بوده است و این میزان برای محیط AuverGrid در حالت ایستا از ۰/۰۴ به ۰/۰۲ در حالت پویا تغییر یافت.



شکل (۱۱): توزیع زمانی تعداد کارهای نادرست پیش بینی شده در محیط AuverGrid توسط پیش‌بینی‌کننده ایستا و پویا

Fig. (11): Time Distribution of the number of incorrect prediction in the AuverGrid environment by static and dynamic predictors

۵-۴- بررسی میزان سربار پیش‌بینی‌کننده پویا

همانگونه که در بخش قبل مشاهده شد، بازآموزی پیش‌بینی‌کننده سبب افزایش کیفیت پیش‌بینی‌های انجام شده توسط آن می‌شود. با این حال این افزایش به قیمت تحمیل سربار زمانی و مکانی برای انجام بازآموزی‌ها و همچنین نگهداری سابقه اجرایی کارهای اجرا شده در محیط به دست آمده است. در این بخش، سربار زمانی تحمیلی برای بازآموزی‌های پیش‌بینی‌کننده در سه محیط DAS2، AuverGrid و Grid5000 گزارش شده است. لازم به ذکرست که این نتایج، میانگین ۱۰ بار اجرا بر روی سیستمی با پردازشگر intel core i5 با ۴ گیگابایت حافظه اصلی و برروی سیستم‌عامل ویندوز ۷ حاصل شده است. در محیط DAS2 به سبب عدم بازآموزی تفاوت بسیار جزیی در دو حالت ایستا و پویا مشاهده شد که افزایش حاصل به سبب نظارتی است که توسط مدیر ارسال کار بر عملکرد پیش‌بینی‌کننده در حالت پویا انجام می‌شود. در AuverGrid مدت زمان اجرا در حالت پویا نسبت به حالت ایستا ۱۰٪ درصد بیشتر است اما در Grid5000 به سبب تعداد بازآموزی‌های نسبتا زیاد، مدت زمان اجرا ۷۵٪ بیشتر بوده است.

۶- نتیجه‌گیری و کارهای آتی

روش‌های داده‌کاوی براساس داده‌هایی که برای تعلیم در اختیار آنها قرار می‌گیرد، الگوهایی را استخراج و براساس آنها مدلی را ایجاد می‌نمایند. مسلماً در صورتی که الگوهای استخراجی، قواعدی همیشگی را دربربگیرند، انتظار ما عملکرد مناسب روش‌های مذکور است. اما اگر داده‌های در اختیار قرار گرفته برای تعلیم، نماینده مناسبی از کل داده‌ها نباشند و یا پویایی محیط به حدی باشد که الگوهای فراگرفته شده به تدریج نامعتبر شوند، این روش‌ها نیازمند بازسازی مجدد با داده‌های جدیدتر خواهند بود. در این پژوهش راهکاری به منظور مقابله با پدیده افت کیفیت پیش‌بینی‌کننده‌ها که نقش مهمی در روش‌های پیشگیرانه مقابله با خرابی کارها ایفا می نمایند، ارائه گردید. با توجه به مشاهدات انجام شده در چند محیط، علاوه بر عامل سالخوردگی، تغییرات لحظه‌ای نیز بعنوان عامل تأثیرگذار در کاهش کیفیت پیش‌بینی‌کننده‌ها معرفی شد. به منظور مقابله با این دو عامل، مولفه‌ای با عنوان مدیر ارسال کار افزوده شد که عملکرد پیش‌بینی‌کننده را نظارت کرده و در صورت مشاهده کاهش در میزان دقت پیش‌بینی کننده، نسبت به انجام بازآموزی اقدام می‌نماید. نتایج حاصل از آزمایشات بر دو محیط نسبتا پویا نشان داد که افزوده شدن این مولفه و انجام بازآموزی از میزان افت کیفیت پیش‌بینی‌کننده می‌کاهد، اما با این حال، به سبب انجام این بازآموزی‌ها سربار زمانی و مکانی نیز به سیستم تحمیل می‌گردد.

یکی از مشاهدات در مورد محیط‌های مورد آزمایش، تناوبی بودن شرایط در آنهاست بدین معنی که الگوهایی که در گذشته مناسب بوده‌اند، در آینده نیز مجدداً قابل بکارگیری خواهند بود. در نوشتار حاضر به این امر توجهی نشده است، در واقع هرگاه مدل فراگرفته شده فعلی عملکرد نامناسبی از خود به نمایش می‌گذارد، به طور کامل کنار گذاشته شده و مدل جدیدی براساس داده‌های جدید ایجاد می‌گردد. به منظور کاهش دادن سربار زمانی می‌توان با به دست آوردن الگوهای مربوط به تکرار شدن شرایط در محیط، از مدل‌هایی که در گذشته ایجاد شده‌اند نیز استفاده مجدد نمود. این موضوع را می‌توان بعنوان یکی از مسایلی که در آینده بدان پرداخته شود بیان نمود.

سپاسگزاری

این مقاله مستخرج از یک طرح پژوهشی است که حامی مالی آن دانشگاه آزاد اسلامی واحد قوچان می باشد. بدینوسیله از دانشگاه آزاد واحد قوچان به سبب حمایت مالی تقدیر و تشکر می گردد.

References

[1] D.A. Cieslak, D. Thain, N.V. Chawla, "Short paper: Troubleshooting distributed systems via data mining", Proceeding of the IEEE/HDPC, pp. 309-312, Paris, France, June 2006.
[2] A.N. Duarte, F. Brasileiro, W. Crine, J.A. Filho, "Collaborative fault diagnosis in grids through automated tests", Proceedings of the IEEE/AINA, Vol. 1, pp. 69-74, Vienna, Austria, April 2006.
[3] H. Li, D. Groep, L. Wolters, J. Templon, "Job failure analysis and its implications in a large-scale production grid", Proceeding of the IEEE/SCIENCE, pp. 27–27, Amsterdam, The Netherlands, The Netherlands, Dec. 2006.
[4] K. Neocleous, M.D. Dikaiakos, V. Fragopoulou, E. Markatos, "Failure management in grids: The case of the EGEE infrastructure", Parallel Processing letters (World Scientific Publishing), Vol. 17, No. 4, pp. 391-410, 2007.
[5] D. Zeinalipour-Yazti, H. Papadakis, C. Georgiou, M.D. Dikaiakos, "Metadata ranking and pruning for failure detection in grids", Parallel Processing Letters, Vol. 18, No. 3, pp. 371-390, Sep. 2008.
[6] C. Dabrowski, "Reliability in grid computing systems", Proceeding of the CCPE, Special issue on Open Grid Forum Vol. 21, No. 8, pp. 927–959, June 2009.
[7] A. Pellegrini, P. Di Sanzo, D.R. Avresky, "A machine learning-based framework for building application failure prediction models", Proceeding of the IEEE/IPDPSW, pp. 1072-1081, Hyderabad, India, May 2015.
[8] L. Shrinivas, J.F. Naughton, "Issues in applying data mining to grid job failure detection and diagnosis", Proceeding of the IEEE/HDPC, Boston, pp. 239-240, June 2008.
[9] Y. Yuan, Y. Wu, Q. Wang, G. Yang, W. Zheng, "Job failures in high performance computing systems: A large-scale empirical study", Computers and Mathematics with Applications, Vol. 63, No. 2, pp.365–377, Jan. 2012.
[10] J. Gu, Z. Zheng, Z. Lan, J. White, E. Hocks, B.H. Park, "Dynamic meta-learning for failure prediction in large-scale systems: A case study", Proceeding of the IEEE/ICPP, pp. 157-164, Portland, OR, USA, Sep. 2008.
[11] P. Garraghan, P. Townend, J. Xu, "An empirical failure-analysis of a large-scale cloud computing environment", IEEE/HASE, pp. 113-120, Miami Beach, FL, USA , Jan. 2014.
[12] S. Fu, C-Z. Xu, "Exploring event correlation for failure prediction in coalitions of clusters", Proceedings of the IEEE/ACM, pp. 1–12, Reno, NV, USA, USA, Nov 2007.
[13] Z. Lan, P. Gujrati, Y. Li, Z. Zheng, R. Thakur, J. White, "A fault diagnosis and prognosis service for teragrid clusters", Proceeding of the IEEE/GRID, Tsukuba, Japan, June 2007.
[14] D. A. Cieslak, N. V. Chawla, and D. L. Thain. "Troubleshooting thousand of jobs on production grids using data mining techniques", Proceeding of the IEEE/ACM, pp. 217-224, 2008.
[15] N.V. Chawla, D. Thain, R. Lichtenwalte, D.A. Cieslak, "Data mining on the grid for the grid", Proceeding of the IEEE/IPDPS, pp. 1-8, Miami, FL, USA, April 2008.
[16] H. Saadatfar, H. Fadishei, and H. Deldari, "Predicting job failures in AuverGrid based on workload log analysis", New Generation Computing, Vol. 30, No. 1, pp. 73-94, Jan. 2012.
[17] H. Saadatfar, H. Deldari, "A job submission manager for large-scale distributed systems based on job futurity predictor", International Journal of Grid and Utility Computing, Vol. 5, No. 1, pp. 50-59, Dec. 2014.

[18] Rosa, Andrea, Lydia Y. Chen, W. Binder, "Predicting and mitigating jobs failures in big data clusters", Proceeding of the IEEE/ACM, pp. 221-230, Shenzhen, China , May 2015.
[19] X. Chen, C.D. Lu, K. Pattabiraman, "Failure analysis of jobs in compute clouds: A google cluster case study", Proceeding of the IEEE/ ISSRE, pp. 167-177,. Naples, Italy, Dec. 2014.
[20] A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, D.H.J. Epema, "The grid workloads archive", Journal of Future Generation Computer Systems, Vol. 24, No. 7, pp. 672-686, Feb. 2008.