



دانشگاه آزاد اسلامی واحد نجف آباد



ششمین کنفرانس ملی مهندسی برق و سیستم‌های هوشمند

ششمین کنفرانس ملی مهندسی برق و سیستم‌های هوشمند - دانشگاه آزاد اسلامی واحد نجف‌آباد - ۱۰ و ۱۱ خرداد ۱۴۰۱

توازن بار سرور و کاهش زمان پاسخ‌دهی در شبکه‌های نرم‌افزار محور با استفاده از بهینه‌سازی ازدحام ذرات

فرزاد باقری^۱، سید محمود دانشور فرزنگان^{۲*}، آوید آوخ^۳

کارشناسی ارشد دانشکده مهندسی کامپیوتر، واحد نجف‌آباد، دانشگاه آزاد اسلامی، نجف‌آباد، ایران^۱

استادیار دانشکده مهندسی برق، واحد نجف‌آباد، دانشگاه آزاد اسلامی، نجف‌آباد، ایران^۲

bagheri.farzad@gmail.com^۱

m_daneshvar@pel.iaun.ac.ir^۲

aavokh@pel.iaun.ac.ir^۳

چکیده: با افزایش حجم ترافیک در شبکه‌های نرم افزار محور، توازن بار سرورها از نیازهای ضروری در این شبکه‌ها به شمار می‌رود. ایده اصلی توازن بار سرورها در شبکه نرم‌افزار محور جایگزینی سخت‌افزاری گران قیمت و ایستا با یک کنترل‌کننده و استفاده از زیر ساخت شبکه در توزیع تقاضاهای سرویس ورودی بر روی آرایه‌ای از سرورها است. اگرچه روش‌هایی برای این هدف ارائه گردیده‌اند که با کاهش هزینه و افزایش انعطاف پذیری توانسته‌اند کارایی توازن بار را بهبود بخشند ولی تحقیقات در این زمینه ادامه دارد. در این پژوهش از روشی مؤثر در جمع‌آوری زمان پاسخ هر سرور استفاده شده و یک الگوریتم فرامکاشف‌های پویا و آگاه از بار سرور، مبتنی بر بهینه‌سازی ازدحام ذرات به منظور توازن بار ارائه می‌گردد. بررسی نتایج نشان می‌دهد روش ارائه شده با توانایی تطبیق با شرایط بار مجموعه سرور و ایجاد تعادل بار، توانسته است تعداد تقاضاهایی پاسخ داده نشده با مهلت پاسخ ۴ ثانیه را تا ۲۴،۲۲ درصد نسبت به روش نوبت‌گردشی و تا ۱۱،۴۱ درصد نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ کاهش دهد. در نتیجه زمان پاسخ توسط روش ارائه شده کاهش یافته و تعداد تقاضای بیشتری در زمان مناسب پاسخ‌دهی می‌شوند و در نتیجه کارایی بهبود می‌یابد.

کلمات کلیدی: شبکه نرم‌افزار محور (SDN)، توازن بار، بهینه‌سازی ازدحام ذرات (PSO)

میزان بالایی از دسترسی‌های هم‌زمان و ترافیک داده به بیش از یک تجهیز برای پردازش می‌باشد که باعث بهبود توانایی پردازش سرور شده و زمان پاسخ به کاربر را کاهش می‌دهد [۳].

یک طبقه بندی برای توازن بار به صورت تکنیک‌های توازن بار سطح داده و تکنیک‌های توازن بار سطح کنترل است که تکنیک‌های توازن بار سطح داده خود شامل دو دسته توازن بار سرور و توازن بار لینک‌های ارتباطی است [۴]. در توازن بار سرور، تقاضای سرویس ورودی را روی آرایه‌ای از سرورهای تکرار شده توزیع و بدین صورت بار کاری سرورها را تقسیم می‌کنند. این کار یک مقدار ثابت از حداقل پهنای باند و تأخیر را برای خدمت به تمامی مشتری‌ها تضمین می‌کند و بهره‌وری افزایش می‌یابد [۵]. به منظور سبک کردن ترافیک سنگین جریان‌های شبکه و کاهش ریسک اعمال سربار به یک سرور تعداد زیادی از مراکز

۱-مقدمه

شبکه‌های سنتی مبتنی بر پروتکل اینترنت (IP^۱) پیچیده بوده و مدیریت آن‌ها سخت می‌باشد [۱]. برای غلبه بر چنین محدودیت‌هایی مفهومی به نام شبکه نرم افزار محور (SDN^۲) پیشنهاد گردید. SDN یک معماری شبکه است که کنترل شبکه از مکانیزم ارسال جدا گردیده است و بستر زیرساخت برای برنامه‌ها و سرویس‌های شبکه بصورت انتزاعی درآمده است.

توازن بار از اولین کاربردهای پیش‌بینی شده برای شبکه‌های نرم افزار محور می‌باشد و الگوریتم‌ها و روش‌هایی برای این هدف ارائه گردیده‌اند [۲]. فناوری توازن بار می‌تواند از اتلاف غیرضروری منابع جلوگیری به عمل آورد. این فناوری، توزیع

^۲ Software-Defined Networking

^۱ Internet Protocol

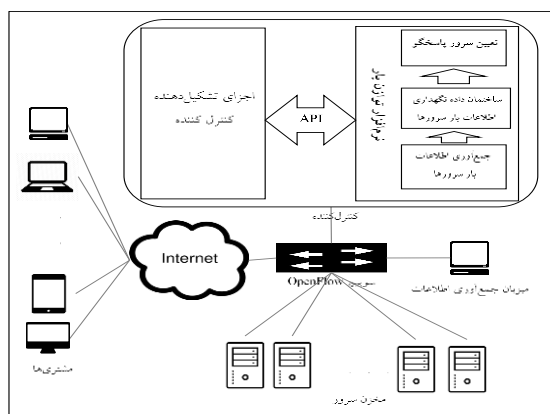
SDN با استفاده از زمان پاسخ واقعی هر سرور ارائه گردید [۶]. کارهای موجود اگرچه توانسته‌اند هزینه بالا را کاهش داده و انعطاف پذیری را بهبود بخشند ولی دارای نقاط ضعف و کاستی‌هایی می‌باشند.

اما توازن بار پویا یک مسأله چندجمله‌ای غیرقطعی کامل (NP-C) است [۹]، بنابراین زمان اجرای الگوریتم‌های موجود متناسب با ابعاد مسأله بصورت نمایی افزایش می‌یابد و از این رو می‌توان از روشهای فرامکاشفه‌ای برای حل این مسأله استفاده نمود.

۲- روش پیشنهادی

در این مقاله با استفاده از یک الگوریتم پویا، مبتنی بر بهینه‌سازی ازدحام ذرات (PSO^{۱۰}) سعی در توزیع بهینه بار سرور و ورودی شده است. در الگوریتم PSO هر ذره در جمعیت که معادل یک موجود در ازدحام می‌باشد دارای مکان و سرعت است و در یک فضای جستجوی چند بعدی حرکت می‌کند. در هر تکرار، هر ذره سرعت خود را بر اساس بهترین موقعیت خودش و موقعیت بهترین ذره در کل جمعیت تنظیم می‌نماید [۱۱].

در شکل ۱ معماری توازن کننده بار در شبکه SDN ارائه و نحوه استقرار نرم‌افزار توازن بار جهت پیاده‌سازی روش پیشنهادی نشان داده شده است.



شکل (۱) معماری توازن کننده بار SDN

در ساختار ارائه شده در شکل ۱ نرم‌افزار توازن بار در یک کنترل کننده مرکزی استقرار می‌یابد. این نرم‌افزار از دو بخش عملکردی

داده از منابع سخت افزاری خاص منظوره برای دستیابی به توازن بار و پشتیبانی از تعداد کاربر بیشتر استفاده می‌کنند [۶]. این تجهیزات توازن بار دارای معایبی از جمله ایجاد یک نقطه شکست، گران قیمت بودن، انعطاف پذیری پایین و نیاز به دانش فنی و مدیریتی خاص در زمینه پیکربندی تجهیزات تولیدکننده-ای خاص می‌باشند. ایده اصلی در مفهوم توازن بار سرورها در SDN جایگزینی مؤلفه سخت‌افزاری گران قیمت و تعریف شده بصورت ایستا برای توازن بار در خوشه سرور، با یک کنترل کننده این فلو^۳ و با استفاده از زیر ساخت شبکه محلی برای پیاده سازی راهکار توازن بار می‌باشد [۷]. SDN اجازه می‌دهد توازن کننده های بار را قابل برنامه ریزی نموده و در طراحی و اجرای استراتژی های توازن بار ایجاد انعطاف پذیری می‌کند [۸].

الگوریتم‌های به کار گرفته شده برای توازن بار به دو دسته ایستا و پویا تقسیم گردیده که توازن بار ایستا وضعیت بار سرور را در نظر نمی‌گیرند و برای شرایطی مناسب می‌باشند که وضعیت بار قابل پیش‌بینی و محاسبه است. در الگوریتم‌های توازن بار پویا، توازن کننده می‌تواند به صورت پویا تقاضاهای دسترسی را بر اساس شرایط بار سرورها توزیع و به سرور بهینه تخصیص دهد. الگوریتم مبتنی بر کمترین ارتباطات^۴، الگوریتم مبتنی بر کمترین بار^۵، الگوریتم پیش بینی کننده^۶ و الگوریتم مبتنی بر زمان پاسخ^۷ از الگوریتم‌های رایج توازن بار پویا می‌باشند [۷]. کارهای متفاوتی در زمینه توازن بار مبتنی بر پروتکل این فلو ارائه شده است؛ شانگ و همکارانش در سال ۲۰۱۳ با استفاده از جمع‌آوری اطلاعات سرور و محاسبه بار جمعی سرورها و انتخاب مجموعه‌ای از سرورها با بار سبک، به توازن بار در محیط SDN دست یافتند [۳]. در سال ۲۰۱۴، ژانگ و گوا با استفاده از یک الگوریتم آمارگیری و جمع‌آوری اطلاعات توازن بار در چارچوب SDN توانستند به حداقل تعداد اتصالات در هر سرور دست پیدا نمایند [۷]. در همین سال یک الگوریتم توازن بار هوشمند با استفاده از الگوریتم ژنتیک توسط چوو و همکارانش ارائه گردید [۹]. کائور و همکارانش یک راهکار توازن بار مبتنی بر الگوریتم نوبت گردشی^۸ ارائه داده و به مقایسه آن با الگوریتم تصادفی پرداختند [۱۰]. در سال ۲۰۱۷، یک الگوریتم توازن بار مبتنی بر معماری

^۷ Response-Time

^۸ Round-Robin

^۹ Nondeterministic Polynomial Time Complete

^{۱۰} Particle Swarm Optimization

^۳ OpenFlow

^۴ Least-Connection

^۵ Least-Loaded

^۶ Predictive

از بین سرورهای موجود می‌باشد. شکل ۴ نحوه نگاشت مسأله را نمایش می‌دهد.

تقاضای ۱ تقاضای ۲ تقاضای ۳ تقاضای ۴ تقاضای ۵

سرور ۳	سرور ۱	سرور ۲	سرور ۳	سرور ۲
--------	--------	--------	--------	--------

شکل (۴) نحوه نگاشت تقاضاهای توازن بار به ذرات PSO

گام اول در PSO، ایجاد جمعیت اولیه ذرات است که معمولاً مکان و سرعت اولیه هر ذره به‌طور تصادفی تولید می‌شود. مکان و سرعت اولیه هر ذره مطابق فرمول‌های ۳ و ۴ ایجاد می‌گردد.

$$x_{0i} = x_{min} + (x_{max} - x_{min}) \times rand \quad (3)$$

$$v_{0i} = v_{min} + (v_{max} - v_{min}) \times rand \quad (4)$$

برای به‌کارگیری بهینه‌سازی ازدحام ذرات در مسأله توازن بار، راهکاری لازم است که قادر باشد مکان‌های پیوسته فضای جستجو را به مقادیر گسسته درست و معتبر تبدیل و نگاشت نماید. به همین منظور قانون کوچک‌ترین مقدار موقعیت (SPV^{11}) برای استفاده از الگوریتم بهینه‌سازی ازدحام ذرات پیوسته در حل تمامی کلاس‌های مسائل با ماهیت گسسته توسعه پیدا کرد [۱۲].

۳-۲- محاسبه برازش مکان هر ذره

برای محاسبه برازش هر ذره از رویکرد استفاده شده در روش مبتنی بر کوتاهترین زمان پاسخ [۶] استفاده می‌شود. بدین صورت برای برازش هر ذره، تاریخچه زمان پاسخ هر سرور در آرایه‌ای در کنترل‌کننده ذخیره می‌گردد. پس از آن در هنگام محاسبه برازش هر ذره مطابق گام‌های زیر عمل می‌نماییم.

• **گام اول:** در این گام مطابق فرمول ۶ بالاترین زمان پاسخ و مطابق فرمول ۷ پایین‌ترین زمان پاسخ در بین ذرات جستجو و یافت می‌شود.

$$T_{Max} = \{ T_{1,0} \cdot T_{2,0} \cdot \dots \cdot T_{n,0} \} \quad (6)$$

$$T_{Min} = \{ T_{1,0} \cdot T_{2,0} \cdot \dots \cdot T_{n,0} \} \quad (7)$$

در فرمول‌های ۶ و ۷، $T_{i,j}$ زمان پاسخ برای ذره i در زمان j است که $T_{i,0}$ به معنی زمان پاسخ در حال حاضر می‌باشد.

• **گام دوم:** بر اساس T_{Max} و T_{Min} به دست آمده، $|T_{Max} - T_{Min}|$ محاسبه می‌گردد. اگر $\lambda < |T_{Max} - T_{Min}|$ آنگاه $T_{i,0}$ به عنوان برازش هر ذره i در نظر گرفته می‌شود و در غیر این صورت گام سوم اجرا می‌گردد.

اصولی تشکیل گردیده است. بخش جمع‌آوری اطلاعات زمان پاسخ سرورهای مستقر در مخزن سرور که توسط میزبان جمع‌آوری اطلاعات ارسال گردیده را دریافت و در آرایه زمان پاسخ هر سرور ذخیره می‌نماید. بخش دیگر تعیین‌کننده سرور پاسخگو است که طبق راه‌کاری بهترین سرور برای پاسخگویی به تقاضای وارده را تعیین و مشخص می‌کند تقاضای HTTP وارد شده توسط کدام سرور خوشه پاسخ داده شود.

۲-۱- الگوریتم توازن بار

مکان و سرعت اولیه هر ذره به‌طور تصادفی مقداردهی اولیه می‌شود. هر ذره مقادیر $pbest$ که بیانگر بهترین مکان شناسایی شده توسط ذره و $gbest$ که بیانگر بهترین مکان شناسایی شده توسط تمامی ذرات از ابتدای اجرای الگوریتم تا لحظه فعلی است را می‌شناسد و در هر تکرار سرعت و موقعیت ذره به‌روزرسانی می‌شود. پارامترهای مورد استفاده در این الگوریتم و معانی آن‌ها در جدول ۱ نشان داده شده است.

در شکل ۳، شبه کد الگوریتم توازن بار PSO قابل مشاهده است.

```

P = Particle_Initialization();
For itr = 1 to itr_Max
  For each particle p in P do
    Fp = Fitness(p)
    If Fp is better than Fitness(pBest)
      pBest = p;
    End;
  End;
  gBest = best p in P;
  For each particle p in P do
    Calculate particle p velocity according equation
    Calculate particle p position according
    equation
  End;
End;

```

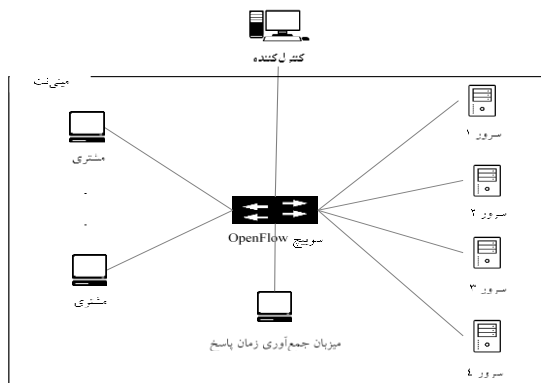
شکل (۳) شبه کد الگوریتم توازن بار PSO

۲-۲- نمایش و ایجاد ذرات

برای حل و اجرای مسأله توازن بار توسط این الگوریتم لازم است ابتدا بین مسأله توازن بار و اجزای الگوریتم PSO نگاشتی صورت می‌گیرد، سپس الگوریتم اجرا می‌گردد. برای نگاشت مسأله توازن بار سرور برای الگوریتم PSO یک روش معمول، نمایش ذره به‌عنوان یک بردار $n \times 1$ است که n نمایش‌دهنده تعداد تقاضاهای پیش فرض به‌عنوان ذرات و مقدار نسبت داده شده به هر ذره اندیس یک سرور

¹¹ Smallest Position Value

سوییچ اپن فلو پس از دریافت اولین بسته تقاضا از مشتری در جدول جریان خود برای پیدا کردن قانونی که بتواند بسته دریافتی را با آن انطباق دهد و مطابق آن عمل نماید به جستجو می پردازد و از آنجا که قانونی در جدول جریان پیدا نمی کند از قانون پیش فرض جدول جریان یعنی ارسال بسته به سمت کنترل کننده استفاده کرده و بسته را با استفاده از پیغام PacketIn به سمت کنترل کننده ارسال می کند.



شکل (۵) توپولوژی شبکه

۴- ارزیابی نتایج

برای ارزیابی، یک مهلت از سوی مشتری معادل ۴ ثانیه برای تقاضاهای HTTP در نظر گرفته می شود که چنانچه زمان پاسخ به تقاضا بیشتر از این مهلت پاسخ گردد ارتباط ایجاد شده برای تقاضا از سوی مشتری قطع می شود پس لازم است مهلت خاتمه ای برای غلبه بر این مشکل ایجاد گردد و به اندازه کافی بزرگ باشد که از دید کاربر چنین زمان پاسخی بالا و قابل قبول نباشد. پس از تنظیم مهلت پاسخ معادل ۴ ثانیه، تعداد کل تقاضاهایی که زمان آن ها از این مهلت گذشته است و پاسخ داده نشده اند نیز محاسبه و درصد تقاضاهای پاسخ داده نشده نسبت به کل تقاضاها برای هر روش محاسبه می شود.

به منظور ارزیابی روش با افزایش نرخ تقاضاهای ارسال درپنج نرخ ۶۰، ۷۰، ۸۰، ۹۰ و ۱۰۰ به مقایسه روش پیشنهادی مبتنی بر الگوریتم PSO با روش نوبت گردشی و روش توازن بار مبتنی بر کوتاه ترین زمان پاسخ [۶] می پردازیم. روش نوبت گردشی به عنوان نماینده روش های ایستا در نظر گرفته شده است که پیش تر نشان داده شده است که نسبت به روش تصادفی عملکرد بهتری دارد [۱۰].

• **گام سوم:** در این گام انحراف استاندارد زمان پاسخ برای هر ذره با در نظر گرفتن تعداد m داده تاریخی ذخیره شده برای هر سرور مطابق فرمول ۸ محاسبه می گردد و به عنوان برازش هر ذره در نظر گرفته می شود.

$$S_i = \sqrt{(T_{i,0} - \bar{T})^2 + (T_{i,1} - \bar{T})^2 + \dots + (T_{i,m-1} - \bar{T})^2} \quad (8)$$

در فرمول ۸، \bar{T} نمایش دهنده متوسط درصد استفاده CPU برای تعداد m داده تاریخی می باشد. همچنین S_i نشان دهنده انحراف استاندارد داده های تاریخی برای هر ذره i است.

۳- پیاده سازی روش پیشنهادی

در جدول ۳ پارامترهای شبیه سازی نشان داده شده است. در جدول ۴ مقادیر عددی پارامترهای الگوریتم PSO نشان داده شده است.

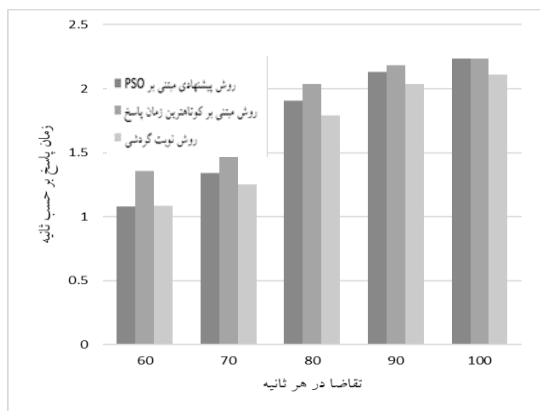
جدول (۳) پارامترهای شبیه سازی

پارامتر	توضیحات
تعداد سرور	۴ سرور
زمان شبیه سازی	۱۰۰ ثانیه
تعداد تکرار هر آزمایش	۱۰ مرتبه
بار تقاضا	با پاسخ ۰/۰۰۴، ۰/۵۲۴، ۱، ۰/۲ مگابایت
زمانی بین ارسال تقاضاها	۱ ثانیه
مهلت پاسخ	۴ ثانیه

جدول (۴) مقادیر عددی پارامترهای الگوریتم PSO

پارامتر	توضیحات
تعداد تکرار حلقه PSO	۱۰۰ مرتبه
تعداد ذرات	۱۰ ذره
ضرایب شتاب C_1 و C_2	۲
x_{min}	۱
x_{max}	۴
v_{min}	۰
v_{max}	۱
ω_{min}	۰/۴
ω_{max}	۰/۹
λ	۰/۱

توپولوژی و مدل سیستم توسعه داده شده در محیط اپن فلو مطابق شکل ۵ است. روند کار بدین صورت است که برنامه نرم افزاری بر روی مشتری ها قرار دارد که در طی زمان شبیه سازی تقاضای HTTP به سمت سوییچ ارسال می کند. در



شکل (۷) زمان پاسخ بر اساس نرخ تقاضای ارسالی

با توجه به شکل ۷، روش پیشنهادی نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ توانسته است زمان پاسخ را در نرخ ۶۰ تقاضا در ثانیه، ۰،۲۷ ثانیه، در نرخ ۷۰ تقاضا در ثانیه ۰،۴۲ ثانیه، در نرخ ۸۰ تقاضا در ثانیه ۰،۱۳ ثانیه و در تعداد ۹۰ تقاضا در ثانیه ۰،۰۵ ثانیه کاهش داشته است. همچنین با افزایش تعداد تقاضا این بهبود کاهش یافته و در تعداد ۱۰۰ تقاضا در ثانیه، متوسط زمان پاسخ در هر دو روش تقریباً یکسان است که دلیل این امر رسیدن متوسط زمان پاسخ به بیشینه خود در این بازه است.

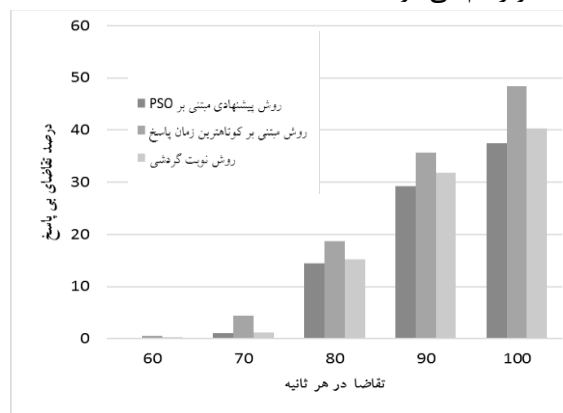
در مقابل روش نوبت گردشی وضعیت بدین گونه نیست و زمان پاسخ در این روش تفاوت اندکی با زمان پاسخ در روش پیشنهادی دارد و این تفاوت به ترتیب برای ۶۰ تقاضا در ثانیه زمان پاسخ تقریباً مساوی، در ۷۰ تقاضا در ثانیه ۰،۰۸ ثانیه، در ۸۰ تقاضا در ثانیه ۰،۱۱ ثانیه، در ۹۰ تقاضا در ثانیه ۰،۰۹ ثانیه و در ۱۰۰ تقاضا در ثانیه برابر ۰،۱۲ ثانیه بوده و این مقادیر کم و عمدتاً به دلیل سربار محاسباتی در روش پیشنهادی است.

شکل ۸ نمایش‌دهنده زمان پاسخ هر یک از سرورهای مجموعه سرور است، در این شکل محور عمودی نمایش‌دهنده زمان پاسخ و محور افقی نمایش‌دهنده شماره هر سرور می‌باشد که برای هر روش متوسط زمان پاسخ هر سرور در تعداد ۷۰ تقاضا در ثانیه به صورت ثابت برای تمامی روش‌ها و در تعداد ۱۰ تقاضا در ثانیه و در طی زمان اجرای ۱۰۰ ثانیه محاسبه و رسم می‌شود.

مطابق شکل ۸، روش پیشنهادی نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ توانسته است زمان پاسخ سرورهای مجموعه سرور را بهبود دهد و برای سرور اول، زمان پاسخ روش پیشنهادی نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ ۰،۴۳ ثانیه، در سرور دوم ۰،۳۷ ثانیه، در سرور سوم ۰،۴۶ ثانیه و در سرور چهارم ۰،۱۵ ثانیه کاهش داشته است.

در تمامی آزمایش‌های انجام شده زمان اجرا ۱۰۰ ثانیه در نظر گرفته شده است و برای محاسبه پارامترهای وابسته ذکر گردیده و رسم شکل آن‌ها برای هر روش، هر آزمایش ده مرتبه تکرار شده است.

در این شبیه‌سازی سرورها شرایط یکسان دارند و بدون هیچگونه نابرابری بار در بین سرورها آزمایش انجام می‌گیرد. در شکل ۵-۱ درصد تقاضاهای پاسخ داده نشده یا به عبارتی دیگر تقاضاهایی که زمان پاسخ آن‌ها بالای ۴ ثانیه بوده است برای هر روش نمایش داده می‌شود. در این شکل محور عمودی نمایش‌دهنده درصد تقاضاهایی که زمان پاسخ آن‌ها بالای ۴ ثانیه بوده و پاسخ داده نشده‌اند نسبت به کل تقاضاهای ارسالی می‌باشد و محور افقی نمایش‌دهنده تعداد تقاضاهای ارسالی در ثانیه است. برای هر روش تعداد تقاضاهای پاسخ داده نشده در ۱۰ تکرار آزمایش و در طی زمان اجرای ۱۰۰ ثانیه محاسبه و رسم می‌شود.



شکل (۶) درصد نرخ تقاضاهای بی پاسخ بر اساس نرخ تقاضای ارسالی

طبق شکل ۶ برای تعداد ۸۰ تقاضا در ثانیه، روش پیشنهادی نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ ۴،۲۷ درصد و نسبت به روش نوبت گردشی ۰،۸۰ درصد، در تعداد ۹۰ تقاضا در ثانیه، نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ ۶،۳۸ درصد و نسبت به روش نوبت گردشی ۲،۵۰ درصد و در تعداد ۱۰۰ تقاضا در ثانیه، این روش نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ ۱۰،۹۸ درصد و نسبت به روش نوبت گردشی ۲،۸۳ درصد بهبود داشته است.

شکل ۷ نمایش‌دهنده زمان پاسخ برای تقاضاهایی است که زمان پاسخ آن‌ها زیر ۴ ثانیه است، در این شکل محور عمودی نمایش‌دهنده زمان پاسخ و محور افقی نمایش‌دهنده تعداد تقاضاهای ارسالی در ثانیه می‌باشد که برای هر روش متوسط زمان پاسخ در ۱۰ تکرار آزمایش و در طی زمان اجرای ۱۰۰ ثانیه محاسبه و رسم می‌شود.

[1] Benson, T. Akella, A. and Maltz, D.A. "Unraveling the Complexity of Network Management," NSDI, pp. 335-348, Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI'09, Berkeley, CA, USA, 2009.

[2] Kreutz, D. Ramos, F.M. Verissimo, P.E. Rothenberg, C.E. Azodolmolky, S. and Uhlig, S. "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, pp. 14-76, 2015.

[3] Shang, Z. Chen, W. Ma, Q. and Wu, B. "Design and implementation of server cluster dynamic load balancing based on OpenFlow," Awareness Science and Technology and Ubimedia Computing (iCAST-UMEDIA), IEEE International Joint Conference on, pp. 691-697, 2013.

[4] Hamdan, M. Hassan, E. Abdelaziz, A. Elhigazi, A. Mohammed, B. Khan, S. Vasilakos, A.V. and Marsono, M.N. "A comprehensive survey of load balancing techniques in software-defined network," *Journal of Network and Computer Applications*, vol. 174, p. 102856, 2021.

[5] Koerner, M. and Kao, O. "Optimizing openflow load-balancing with I2 direct server return," Network of the Future (NOF), IEEE 4th International Conference on the, pp. 1-5, 2013.

[6] Zhong, H. Fang, Y. and Cui, J. "LBBSRT: An efficient SDN load balancing scheme based on server response time," *Future Generation Computer Systems*, Elsevier, vol. 68, pp. 183-190, 2017.

[7] Zhang, H. and Guo, X. "SDN-based load balancing strategy for server cluster," Cloud Computing and Intelligence Systems (CCIS), IEEE 3rd International Conference on, pp. 662-667, 2014.

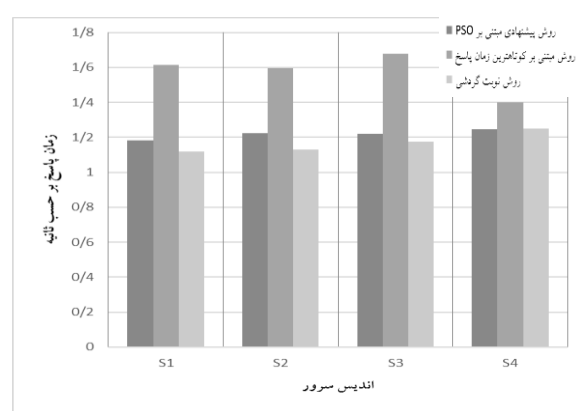
[8] Semong, T. Maupong, T. Anokye, S. Kehulakae, K. Dimakatso, S. Boipelo, G. and Sarefo, S. "Intelligent load balancing techniques in software defined networks: A survey," *Electronics*, vol. 9, p. 1091, 2020.

[9] Chou, L.-D. Yang, Y.-T. Hong, Y.-M. Hu, J.-K. and Jean, B. "A genetic-based load balancing algorithm in openflow network," *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, Springer, pp. 411-417, 2014.

[10] Kaur, S. Kumar, K. Singh, J. and Ghumman, N.S. "Round-robin based load balancing in Software Defined Networking," Computing for Sustainable Global Development (INDIACom), IEEE 2nd International Conference on, pp. 2136-2139, 2015.

[11] Kennedy, J. and Eberhart, R. "Particle swarm optimization," Neural Networks, IEEE International Joint Conference on, pp. 1942-1948, 1995.

[12] Fatih Tasgetiren, M. Liang, Y.-C. Sevcli, M. and Gencyilmaz, G. "Particle swarm optimization and differential evolution for the single machine total weighted tardiness problem," *International Journal of Production Research*, vol. 44, pp. 4737-4754, 2006.



شکل (۸) زمان پاسخ هر سرور در تعادل بار اولیه

در مقابل روش نوبت گردشی وضعیت بدین گونه نیست و زمان پاسخ سرورها در این روش به دلیل تعداد تقاضای ارسالی کم و عدم پر شدن صف تقاضای سرورها تقریباً مساوی زمان پاسخ در روش پیشنهادی است.

۴- نتیجه‌گیری

در این پژوهش روشی پویا مبتنی بر الگوریتم فرامکاشفای PSO به منظور توازن بار سرور ارائه گردید. هدف از ارائه این روش کاهش زمان پاسخ به مشتری بود که برای این منظور یک نرم افزار پیاده‌سازی و در کنترل‌کننده مستقر گردید. همچنین به منظور جمع‌آوری اطلاعات بار سرور برای استفاده در الگوریتم توازن بار، راه‌کاری ارائه و پیاده‌سازی گردید.

میزان بهبود ایجاد شده در زمان پاسخ توسط روش ارائه گردیده در مقایسه با روش نوبت گردشی و روش مبتنی بر کوتاه‌ترین زمان پاسخ توسط پارامترهای زمان پاسخ و درصد تقاضاهای پاسخ داده نشده ارزیابی گردید. با توجه به نتایج به دست آمده روش پیشنهادی می‌تواند تعداد تقاضاهایی که زمان پاسخ آن‌ها بیش از حد تعیین شده ۴ ثانیه است را در شرایط تعادل بار بین ۴,۲۷ تا ۱۰,۹۸ درصد نسبت به روش مبتنی بر کوتاه‌ترین زمان پاسخ و ۰,۸۰ تا ۲,۸۳ درصد نسبت روش نوبت گردشی کاهش دهد.

۵- مراجع