دانشگاه صنعتی شریف

مقدمه‌ای بر توزیع بار و الگوریتم‌ها

امیرحسین روان نخجوانی | ۴۰۰۱۰۴۹۷۵

عبدالصمد حقیری | ۹۸۱۰۵۷۲۷

بهار ۱۴۰۳

فهرست مطالب

I. چکیده

با بزرگ شدن مقیاس سیستم‌ها نیاز به راه‌حلی پویا برای توزیع بار روی منابع مختلف داریم، به صورتی که قابلیت اطمینان و حاضر بودن سیستم را افزایش دهد، تا بتواند در زمان‌های اوج ترافیک بدون خرابی به همهٔ درخواست‌ها در زمان معقول پاسخ دهد. این توزیع می‌تواند در لایه‌های مختلفی رخ دهد که معمولا در لایه‌های ۴ و ۷ مدل OSI رخ می‌دهد. سیاست‌های مختلف توزیع بار در ابزارهای مختلف پیاده‌شده‌اند که کافیست ما ازین ابزارها در سیستم خود استفاده کنیم.

II. کلیدواژه‌ها

توزیع بار، مقیاس‌پذیری، NginX، توسعه ابری

III. مقدمه

سیستم‌ها محدودیت زیادی در توسعه عمودی[1] دارند و پس از زیاد شدن کاربرها بعد از نقطه‌ای باید به صورت غیرقابل اجتنابی به سراغ توسعه افقی[2] برویم. خوشبختانه با پیشرفت مفاهیم توسعه ابری[3] این امر، امری قابل انجام است. با افزایش تعداد سرویس‌دهندگان یک سیستم سوالی حیاتی‌ای پیش می‌آید: **چگونه درخواست‌های ورودی را بین سرویس‌دهندگان پخش کنیم؟** این سوالی است که مفاهیم توزیع بار[4] به آن پاسخ می‌دهند. در سیستم‌های توزیع‌شده[5]، محاسبات ابری و خدمات وب، توازن بار با مدیریت و توزیع کارآمد درخواست‌های ورودی، تجربیات یکپارچه‌ای را برای کاربران فراهم می‌کند.

هدف اصلی توازن بار بهینه‌سازی استفاده از منابع، افزایش توان عملیاتی، کاهش زمان پاسخ و جلوگیری از بارگذاری بیش از حد هر منبع است. با اطمینان از اینکه هیچ منبعی به تنهایی بار زیادی را تحمل نمی‌کند، توازن بار عملکرد و قابلیت اطمینان[6] کلی برنامه‌ها و خدمات را بهبود می‌بخشد.

---

[1] Vertical Scaling
[2] Horizontal Scaling
[3] Cloud Native
[4] Load Balancing
[5] Distributed Systems
[6] Reliability

۱۷. تاریخچه

۱) ظهور پردازنده‌ها

در ابتدا با ظهور پردازنده‌ها مفاهیم توزیع بار تنها در زمینه تقسیم عملیات‌ها بین هسته‌های یک پردازنده بود تا پردازنده سرعت و بهره‌وری بیشتری داشته باشد.

۲) انفجار اینترنت

با به وجود آمدن اینترنت مهم‌ترین موضوع نحوه اتصال سرورهای مختلف به یکدیگر برای تشکیل یک شبکه بود. این موضوع حتما منجر به سوال مورد نظر ما می‌رسد، چرا که تقسیم بسته‌های اینترنت چالشی جدایی‌ناپذیر از مفاهیم شبکه‌ها است. در این دوره حضور الگوریتم‌های پیشرفته‌تر باعث پیشرفت چشم‌گیری در زمینهٔ توزیع بار شد.

۳) عصر وب

با به موفقیت رسیدن کسب‌وکارهای اینترنتی[۷] اهمیت قابلیت توسعه‌پذیری[۸] یک سیستم چند برابر شد. سرورها و سایت‌های خرید آنلاین باید توانایی مدیریت تعداد درخواست‌های بالا در ساعات اوج ترافیک را بدون تاخیر چشم‌گیر جواب می‌داند. در این دوره کتاب‌ها و مقالات زیادی در زمینهٔ قابلیت اطمینان و توسعه سیستم‌های نرم‌افزاری تولید شد.

۷. انواع توزیع‌بار

توزیع بار می‌تواند در لایه‌های مختلفی انجام شود.

۱) مبتنی بر DNS

این توزیع مبتنی بر توزیع ترافیک بر اساس پاسخ‌های DNS است که مشتریان را به آدرس‌های IP مختلف هدایت می‌کند. در حالی که این روش راهی ساده برای توزیع بار فراهم می‌کند، فاقد جزئیات دقیق و قابلیت حساب‌کردن بار سرور در زمان واقعی است.

---

[7] E-Commerce
[8] Scalability

۲) مبتنی بر سرور

این روش از توزیع‌کننده‌های بار فیزیکی یا مجازی برای هدایت ترافیک به سرورهای متعدد استفاده می‌کند. این توازن‌کننده‌ها می‌توانند تصمیمات زمان واقعی بر اساس بار فعلی، سلامت سرور و معیارهای عملکرد بگیرند.

۳) مبتنی بر برنامه

این روش در سطح برنامه عمل کرده و وظایف یا درخواست‌های خاص را بین نمونه‌ها یا خدمات مختلف توزیع می‌کند. این روش به ویژه برای معماری‌های میکروسرویس‌ها مفید است که خدمات مختلف وظایف خاصی را مدیریت می‌کنند.

VI. الگوریتم‌های مختلف توزیع بار

هر سیستم توزیع بار با یک سیاست خاص عمل می‌کند که تعیین می‌کند بار چگونه باید تقسیم‌بندی شود. در ادامه به معرفی برخی از معروف‌ترین الگوریتم‌های توزیع‌بار می‌پردازیم.

۱) چرخشی (Round Robin)

این الگوریتم از طریق لیستی از سرورها چرخیده و هر درخواست جدید را به سرور بعدی در لیست اختصاص می‌دهد. این روش ساده است و در صورتی که سرورها ظرفیت‌ها و بارهای مشابهی داشته باشند، به خوبی کار می‌کند.

۲) چرخشی وزن‌دار (Weighted Round Robin)

این روش یک نسخه پیشرفته‌تر از روش چرخشی است که به هر سرور بر اساس ظرفیتش یک وزن اختصاص می‌دهد. سرورهای با ظرفیت بالاتر درخواست‌های بیشتری دریافت می‌کنند.

۳) کمترین اتصالات (Least Connections)

این الگوریتم ترافیک را به سروری هدایت می‌کند که کمترین تعداد اتصالات فعال را دارد. این روش در محیط‌هایی که بار سرورها می‌تواند به طور قابل توجهی متفاوت باشد، مؤثر است.

۴) هش IP (IP Hashing)

این الگوریتم از یک تابع هش بر روی آدرس IP مشتری استفاده می‌کند تا تعیین کند که کدام سرور باید درخواست را مدیریت کند. این روش تضمین می‌کند که همان IP مشتری به همان سرور هدایت می‌شود، که به ماندگاری نشست[9] کمک می‌کند.

۵) کمترین زمان پاسخ (Least Response Time)

این روش درخواست‌ها را به سروری هدایت می‌کند که کمترین زمان پاسخ را دارد، و عملکرد را برای برنامه‌های حساس به زمان بهینه می‌کند.

سیاست‌های مختلفی برای این کار وجود دارد که هر کدام نقاط ضعف و قوت مخصوص خود را دارند. انتخاب سیاست درست می‌تواند روی عملکرد سیستم تاثیر مهمی بگذارد. پیاده‌سازی درست این الگوریتم‌ها چالش‌های زیادی را در بر دارد، اما امروزه ابزارهای مختلفی وجود دارند که می‌توانند این کار را در لایه‌های مختلف انجام دهند که در بخش‌های بعدی معرفی انجام می‌شود.

VII. پیاده‌سازی توزیع بار

قبل از پیاده‌سازی فنی لازم است که درک درستی از سیستم‌ها و نیازهای آن داشته باشیم تا بتوانیم در انتخاب ابزارهایی که برایمان توزیع بار را ممکن می‌کند، بهینه عمل کنیم. با فرض اینکه نیازهای سیستمی به خوبی درک شده‌اند به سراغ معرفی ابزارها می‌رویم.

۱) ابزارهای سخت‌افزاری

دستگاه‌هایی از فروشندگانی مانند F5 Networks و Cisco عملکرد و ویژگی‌های قدرتمندی ارائه می‌دهند اما می‌توانند گران و پیچیده برای مدیریت باشند.

۲) ابزارهای نرم‌افزاری

راه‌حل‌هایی مانند Nginx، HAProxy و Traefik گزینه‌های انعطاف‌پذیر و مقرون به صرفه‌ای فراهم می‌کنند. آن‌ها می‌توانند به صورت محلی یا در محیط‌های ابری مستقر شوند.

۳) ابزارهای ابری

[9] Session

خدمات ارائه‌شده توسط ارائه‌دهندگان ابر مانند AWS، Azure و Google Cloud راه‌حل‌های توازن بار قابل مقیاس و مدیریت‌شده‌ای ارائه می‌دهند که به‌طور یکپارچه با دیگر خدمات ابری ادغام می‌شوند.

VIII. چالش‌ها و مفاهیم پیشرفته‌تر

۱) چک سلامت (Health Check)

اگر یک نمونه سرویس‌دهنده در فرایند پاسخ به یکی از درخواست‌ها به مشکلی بخورد باید بار به صورت هوشمند به سراغ بقیهٔ نمونه‌ها برود، برای این کار توازن‌کننده‌های بار به طور منظم سلامت سرورها را با ارسال درخواست‌های دوره‌ای بررسی می‌کنند. اگر یک سرور این چک‌های سلامت را نگذرد، به طور موقت از گروه سرورها حذف می‌شود تا زمانی که بازیابی شود. با این کار توزیع‌کننده اطمینان حاصل می‌کند که تنها سرورهای سالم ترافیک دریافت می‌کنند، و قابلیت اطمینان کلی را بهبود می‌بخشد.

۲) ماندگاری نشست (Sticky Sessions)

برخی برنامه‌ها نیاز دارند که درخواست‌های یک کاربر به طور مداوم به همان سرور ارسال شوند. توازن‌کننده‌های بار می‌توانند با استفاده از روش‌هایی مانند هش IP یا کوکی‌ها نشست‌ها را پیگیری کنند. چیزی که اهمیت دارد Stateless[10] بودن برنامه‌هاست. اگر برنامه‌ای اینگونه نباشد افزایش قابلیت اطمینان و درستی برنامه امری به مراتب سخت‌تر است.

۳) توازن بار جهانی سرور (GSLB)

سیاست‌های توزیع بار تنها بر اساس میزان درخواست یا نوع درخواست عمل نمی‌کنند بلکه باید بر اساس موقعیت جغرافیایی درخواست‌کننده نیز تصمیم بگیرند. برای برنامه‌هایی با یک پایگاه کاربر جهانی، GSLB ترافیک را بین مراکز داده متعدد در مناطق جغرافیایی مختلف توزیع می‌کند. این روش تأخیر را بهبود می‌بخشد و قابلیت جایگزینی را برای نمونه‌های دارای مشکل فراهم می‌کند.

٤) یکپارچه‌سازی با مقیاس‌پذیری خودکار (Auto-scaling Integration)

[10] برای اطلاعات بیشتر به سراغ twelve-factor apps بروید.

محیط‌های ابری مدرن اجازه می‌دهند که منابع بر اساس تقاضا به صورت پویا مقیاس‌پذیر شوند. توازن‌کننده‌های بار می‌توانند با گروه‌های مقیاس‌پذیری خودکار یکپارچه شوند تا در صورت نیاز سرورها را اضافه یا حذف کنند، و اطمینان حاصل کنند که منابع به طور کارآمد استفاده می‌شوند و صرفه‌جویی در هزینه‌ها انجام می‌شود. اخیرا با پیشرفت در زمینه یادگیری ماشین روش‌هایی برای یادگیری میزان منابع مورد نیاز برای زمان‌ها و شرایط مختلف نیز به وجود آمده.

IX. نتیجه‌گیری

توازن بار یک مفهوم اساسی در محاسبات مدرن است که امکان عملکرد کارآمد و قابل اطمینان سیستم‌ها و برنامه‌های توزیع‌شده را فراهم می‌کند. با توزیع هوشمندانه بارهای کاری بین منابع متعدد، توازن بار از دسترسی بالا و مقیاس‌پذیری اطمینان می‌دهد. با وجود الگوریتم‌ها و استراتژی‌های پیاده‌سازی متنوع، توازن بار یک راه‌حل چندمنظوره است که می‌تواند برای برآورده کردن نیازهای محیط‌های محاسباتی مختلف سفارشی‌سازی شود.

X. منابع

1. Bostrom, Nick. "Superintelligence: Paths, Dangers, Strategies." Oxford University Press, 2014.
2. Tanenbaum, A. S., & Van Steen, M. (2016). Distributed Systems: Principles and Paradigms. Pearson.
3. Hohpe, G., & Woolf, B. (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley.
4. “AWS Elastic Load Balancing” Amazon Web Services. https://aws.amazon.com/elasticloadbalancing/
5. “Nginx Load Balancing” Nginx. https://www.nginx.com/resources/glossary/load-balancing/
6. “HAProxy Documentation”. HAProxy Technologies. https://www.haproxy.org/documentation.html
7. “Load Balancing Algorithms and Techniques”. TechTarget. https://www.techtarget.com/searchnetworking/definition/load-balancing

8. “Global Server Load Balancing Explained”. Cloudflare. https://www.cloudflare.com/learning/cdn/glossary/global-server-load-balancing-gslb/
9. “Session Persistence”. F5 Networks. https://techdocs.f5.com/en-us/bigip-15-1-0/big-ip-local-traffic-managers-ltm-concepts/persistence-concepts.html