

بررسی و ارزیابی سرور های استریمینگ و لبه (edge) و خودکار کردن فرآیند تست و استقرار پیکربندی سرورها



دانشجو: محمد امین فاضلی
استاد راهنما: دکتر خامس پناه
دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران

اقدام دیگری که در راستای بهبود caching علاوه بر تغییر معماری گفته شده انجام شده، مربوط به sharding می باشد. این اقدام باعث شد در سرورهای کش ما cache hit ratio به طور قابل توجهی افزایش یابد. در این روش، به جای ارسال چرخشی (round robin) درخواستها به همه سرورها، درخواستها با استفاده از هشینگ ثابت (consistent hashing) به سرورهای خاصی هدایت می شوند، که این کاهش تجدید کش و افزایش نسبت هیت کش را به دنبال داشت. تقسیم بندی محتوا بین سرورهای مختلف امکان نگهداری بخش بیشتری از کل محتوا در کش را فراهم می کند.

برای تضمین صحت محتوای ویدیویی، یک اسکریپت پایتون ساخته شده است که فایل های M3U8 و فهرست های فرعی و فایل های TS را برای تضمین بارگذاری بدون خطا و دسترس پذیری بررسی می کند. همچنین استفاده از Black Box Exporter به تضمین دسترسی مداوم محتوا کمک می کند، همچنین با ارسال خودکار درخواست به فایل ها برای اطمینان از دسترس پذیری، نظارت مداوم و واکنش سریع به مشکلات را فراهم می آورد.

نتایج و جمع بندی

- مزایای معماری جدید:** معماری جدید با تفکیک سرورهای ترانسکود و کش، مقیاس پذیری را بهبود می بخشد و به مدیریت بهتر ترافیک کمک می کند. این تغییر افزایش قابلیت اطمینان را به دنبال دارد، زیرا خرابی در یک بخش، تأثیر کمتری بر بخش های دیگر می گذارد. علاوه بر این، عملکرد بهینه سازی شده با تنظیم جداگانه سرورها، کاهش فشار بر سرورهای ترانسکود از طریق کش کردن محتوای پردرخواست و افزایش نسبت هیت کش را ممکن می سازد.
- مزایای شاردینگ:** استفاده از شاردینگ در سرورهای کش cache hit ratio را به طور قابل توجهی افزایش داده است. این روش با تخصیص دقیق محتوا به سرورهای خاص، محتوای بیشتری را برای مدت زمان طولانی تری در کش نگه می دارد و درخواست های کمتری به سرورهای VOD ارسال می شود. این امر نیاز به تجدید کش را کاهش داده و به توزیع متوازن بار بین سرورها کمک می کند، که به نوبه خود به میزان قابل توجهی عملکرد سیستم را بهبود می بخشد. با استفاده از این معماری و استفاده از تکنیک شاردینگ توانستیم نسبت کش هیت را از حدود ۷۰-۶۰ درصد به بالای ۹۰-۸۵ درصد در همه سرور های استریم برسانیم که این تغییر بزرگ تأثیر شگرفی در میزبانی کردن ترافیک بیش تر و همچنین افزایش قابل اتکا بودن سیستم گذاشته است.
- مزایای چرخه جدید CI/CD:** معماری جدید چرخه CI/CD با بهره گیری از اتوماسیون از طریق Ansible و سایر ابزارها، سرعت توسعه را افزایش داده است. این معماری مقیاس پذیری را بهبود بخشیده، خطاهای انسانی را کاهش داده، ردیابی و مستندسازی را بهبود می بخشد و اعتبارسنجی خودکار کیفیت کد و پیکربندی ها را قبل از استقرار فراهم می آورد. این امکانات فرآیندهای دستی زمان بر را حذف کرده و به تیم ها امکان می دهد به نیازهای در حال تغییر با سرعت و کارایی بیشتری پاسخ دهند.

تغییر از معماری قدیمی به معماری جدید به معنای مقابله با چالش های رو به رشد کاربران و ترافیک بوده و اگرچه با افزایش پیچیدگی و هزینه ها همراه است، اما مزایای قابل توجهی از جمله بهبود مقیاس پذیری، قابلیت اطمینان و عملکرد را به همراه دارد. این تغییرات ضروری برای حفظ رقابت پذیری و ارائه خدمات با کیفیت به کاربران در دنیای مدرن است.

مراجع اصلی

- J. Humble و D. Farley, "Continuous Delivery: Reliable Software Releases through Build Test and Deployment Automation", 2010.
- B. Beyer, C. Jones, J. Petoff, و N. R. Murphy, "Site Reliability Engineering: How Google Runs Production Systems", 2016.
- "Nginx HTTP Server", Nginx Inc, Technical documents: open source.

مقدمه و خلاصه

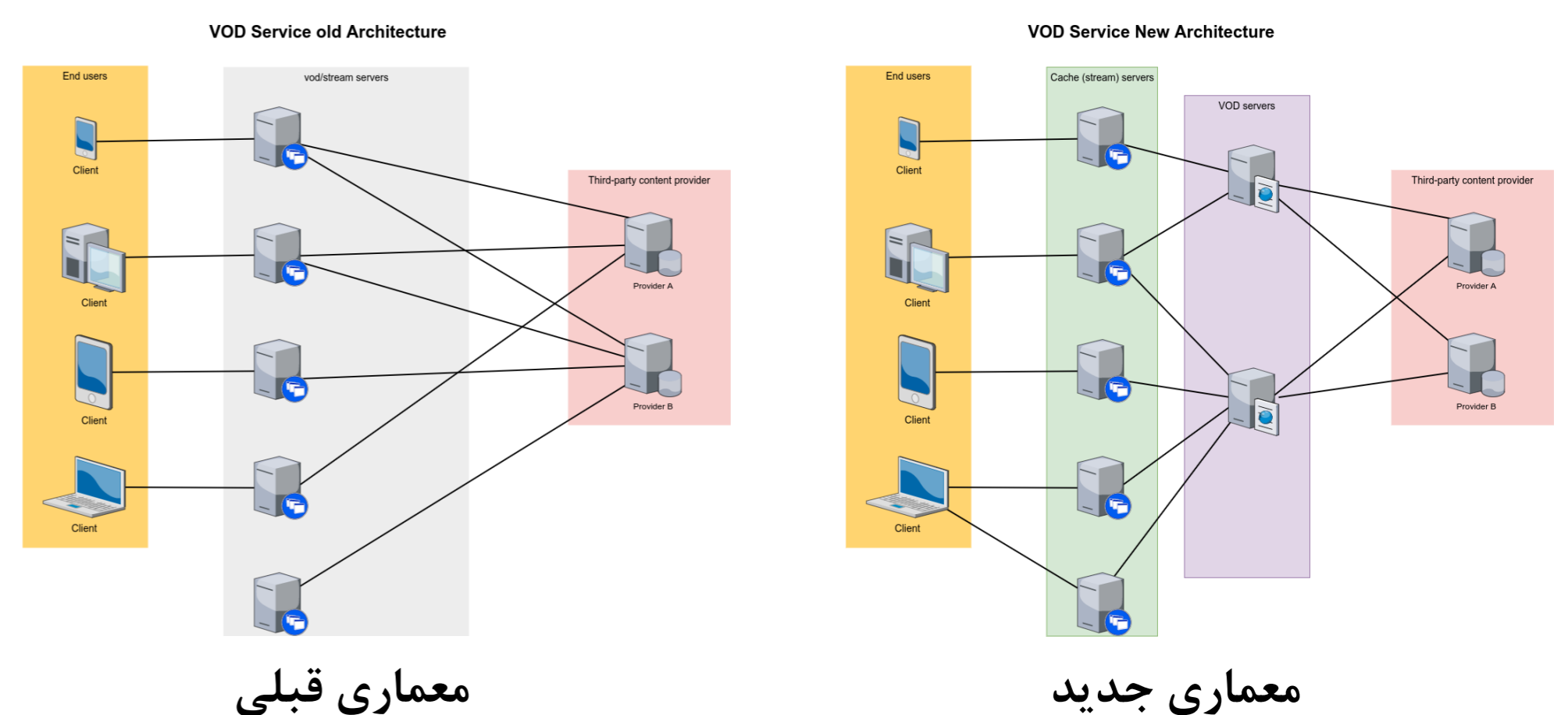
ما در مایکت (اپ استور ایرانی) یک بخش فیلم و سریال داریم که در اپ قابل مشاهده است. این سرویس VOD از سال ۱۴۰۰ شروع به کار کرد. ابتدا به راحتی جوابگوی نیاز کاربران بود ولی بعد از مدتی با افزایش چشمگیر کاربران روبرو شدیم.

طبق آخرین آمار رسمی هم اکنون میزبان حدود ۷ هزار فیلم و سریال برای بیش از ۵ میلیون کاربر هستیم و توانستیم از مرز ۵۰۰ میلیون دقیقه پخش عبور کنیم. مطمئناً وضعیت فعلی بدون بررسی و بازنگری در معماری و تلاش برای بهبود این سرویس امکان پذیر نبوده و با تلاش همکارانم هم اکنون این سرویس در این مقیاس توانایی ارائه خدمت دارد.

در این پروژه سعی شده اقداماتی که برای بهبود این سرویس انجام شده با تمرکز بر قسمت خودکارسازی فرآیند تست و استقرار پیکربندی سرور ها و اطمینان از صحت محتوای ارائه شده را بیان کنیم که شامل بخش هایی مثل تست و استقرار خودکار، بهبود عملکرد caching، استفاده از sharding برای توزیع بار بهتر و عملکرد بهتر caching، تنظیم پارامتر های سیستم عامل و وب سرور برای سرو بهتر محتوا و اطمینان از درستی محتوا دریافتی کاربر می باشد.

مدل پیشنهادی و پیاده سازی

تغییر معماری سیستم های فناوری اطلاعات معمولاً به منظور بهبود عملکرد، افزایش قابلیت اطمینان و مقیاس پذیری انجام می شود. در این مورد، تغییر معماری سیستم از یک ساختار تک سروری که هم کدگذاری ویدیویی و هم کش محتوا را انجام می داد به معماری جدیدی با تفکیک وظایف به سرورهای کدگذار ویدیویی و سرورهای کش انتقال یافته است.



بهبود دیگر مربوط به پیکربندی وب سرورها می باشد. در گذشته پیکربندی های Nginx در یک مخزن نگهداری می شدند و برای اعمال تغییرات یا استقرار جدید روی سرورها، نیاز به اتصال دستی به سرور از طریق SSH، کپی پیکربندی جدید و راه اندازی مجدد Nginx بود. این فرآیند زمان بر و خطاپذیر بوده و کارایی تیم را کاهش می داد، زیرا هر تغییر نیاز به دخالت دستی داشت.

اکنون به ازای هر سرویس یک مخزن جدا برای پیکربندی های Nginx وجود دارد که به عنوان submodule در مخزن اصلی Ansible قرار گرفته است تا به روزرسانی ها را دنبال کند و هنگام اجرای اسکریپت های Ansible پیکربندی را بگیریم و اعمال کنیم.