Performance & Scalability Plan

در پروژهMetrogo ، عملکرد و مقیاس‌پذیری سیستم صرفاً یک دغدغه فنی نیست، بلکه مستقیماً با بقای کسب‌وکار، تجربه کاربر و اعتماد ذی‌نفعان گره خورده است. یک محصول حوزه حمل‌ونقل شهری، به‌ویژه مترو، باید در شرایط اوج استفاده، بحران‌های ناگهانی، افزایش هم‌زمان کاربران و تراکنش‌ها و محدودیت‌های زیرساختی، همچنان قابل اتکا باقی بماند. این سند با همین نگاه تدوین شده است: پاسخ به این سؤال کلیدی که آیا رشد کاربران و تراکنش‌ها می‌تواند سیستم Metrogo را از پا درآورد یا خیر، و اگر پاسخ منفی است، با چه هزینه و چه طراحی‌ای این پایداری تضمین می‌شود.

درMetrogo ، عملکرد سیستم به‌عنوان بخشی از وعده ارزشی محصول تعریف شده است. کاربر انتظار دارد اطلاعات مسیر، زمان‌بندی، پرداخت و اعلان‌ها در کوتاه‌ترین زمان و بدون اختلال در دسترس باشند. هر تأخیر، قطعی یا رفتار غیرقابل پیش‌بینی مستقیماً به کاهش اعتماد و ترک محصول منجر می‌شود. بنابراین عملکرد خوب نه یک «بهینه‌سازی بعداً»، بلکه یک الزام از روز اول است.

نگاه Metrogo به SLI و SLO

در طراحی عملکردMetrogo ، ابتدا شاخص‌های قابل اندازه‌گیری عملکرد یا همان SLI تعریف شده‌اند. این شاخص‌ها به‌گونه‌ای انتخاب شده‌اند که هم از دید کاربر معنادار باشند و هم از دید تیم فنی قابل پایش و کنترل. مهم‌ترین این شاخص‌ها شامل دسترس‌پذیری سرویس، زمان پاسخ درخواست‌ها، نرخ خطا در عملیات کلیدی و پایداری در شرایط بار بالا هستند.

بر اساس این شاخص‌ها، اهداف سطح سرویس یا SLO تعیین شده‌اند. برای مثال، دسترس‌پذیری سیستم به‌گونه‌ای تعریف شده که در بازه‌های زمانی مشخص، سرویس اصلی Metrogo با درصد بالایی در دسترس کاربران باشد. این عدد صرفاً یک ادعای بازاریابی نیست، بلکه بر اساس توان زیرساخت، معماری سیستم و هزینه‌های قابل تحمل برای استارتاپ انتخاب شده است Metrogo .آگاهانه بین هزینه و سطح خدمت تعادل برقرار کرده و به‌جای وعده‌های غیرواقعی، اهدافی تعریف کرده که قابل دستیابی و پایدار باشند.

در موردlatency ، Metrogoبه‌طور مشخص روی تجربه کاربر تمرکز دارد. زمان پاسخ برای عملیات حیاتی مانند جستجوی مسیر، دریافت زمان حرکت قطارها و ثبت پرداخت باید در محدوده‌ای باقی بماند که کاربر احساس «سیستم زنده و سریع» داشته باشد. این موضوع به‌صورت کمی تعریف شده و مبنای پایش مداوم قرار گرفته است.

ظرفیت‌سنجی و پیش‌بینی رشد

Metrogoاز ابتدا با فرض رشد تدریجی اما پیوسته طراحی شده است. ظرفیت‌سنجی سیستم تنها بر اساس وضعیت فعلی انجام نشده، بلکه سناریوهای رشد کاربران، افزایش تراکنش‌ها در ساعات اوج، و رویدادهای خاص شهری نیز در نظر گرفته شده‌اند. این سناریوها به تیم اجازه می‌دهند بفهمد سیستم در چه نقطه‌ای به محدودیت نزدیک می‌شود و چه منابعی باید قبل از رسیدن به آن نقطه اضافه شوند.

در این پروژه، ظرفیت‌سنجی صرفاً به معنای «چند کاربر هم‌زمان» نیست. بلکه شامل بررسی مصرف منابع در لایه‌های مختلف سیستم است؛ از پایگاه داده گرفته تا سرویس‌های واسط وAPI ها. این تحلیل کمک می‌کند مشخص شود کدام بخش‌ها زودتر اشباع می‌شوند و کدام بخش‌ها فضای رشد بیشتری دارند.

برنامه Load Test وStress Test

Metrogoبرای اطمینان از عملکرد سیستم، برنامه مشخصی برای آزمون بار و آزمون فشار دارد. این آزمون‌ها نه‌تنها برای تأیید وضعیت فعلی سیستم، بلکه برای کشف نقاط ضعف پنهان طراحی شده‌اند. در این تست‌ها، رفتار واقعی کاربران شبیه‌سازی می‌شود؛ از ورود هم‌زمان کاربران در ساعات اوج گرفته تا ثبت پرداخت‌های متوالی و دریافت اعلان‌ها.

هدف از این تست‌ها صرفاً رسیدن به یک عدد نیست، بلکه مشاهده رفتار سیستم در شرایط غیرعادی است. تیم Metrogo به‌دنبال پاسخ این سؤال است که وقتی بار از حد انتظار عبور می‌کند، سیستم چگونه شکست می‌خورد. آیا به‌صورت تدریجی کند می‌شود یا به‌طور ناگهانی از دسترس خارج می‌شود؟ پاسخ به این سؤال نقش مهمی در طراحی راهکارهای مقیاس‌پذیری دارد.

شناسایی گلوگاه‌ها(Bottlenecks)

یکی از نقاط قوت رویکرد Metrogo این است که فرض نمی‌کند سیستم بی‌نقص است. گلوگاه‌های بالقوه از ابتدا شناسایی و مستندسازی شده‌اند. برای مثال، پایگاه داده می‌تواند در سناریوهای خاص به نقطه فشار برسد، یا سرویس‌های وابسته خارجی ممکن است در ساعات اوج پاسخ‌گو نباشند.

به‌جای نادیده گرفتن این گلوگاه‌ها، تیم Metrogo آن‌ها را به‌عنوان ریسک‌های شناخته‌شده پذیرفته و برای هرکدام راهکار کاهش ریسک تعریف کرده است. این راهکارها شامل بهینه‌سازی کوئری‌ها، کش‌گذاری، تفکیک سرویس‌ها و در صورت لزوم بازطراحی بخشی از معماری است.

استراتژی مقیاس‌پذیری

مقیاس‌پذیری در Metrogo به‌صورت تدریجی و مبتنی بر نیاز واقعی طراحی شده است. سیستم به‌گونه‌ای ساخته شده که امکان افزایش ظرفیت بدون توقف سرویس فراهم باشد. این مقیاس‌پذیری تنها به افزایش منابع سخت‌افزاری محدود نمی‌شود، بلکه شامل بهبود طراحی نرم‌افزاری و کاهش وابستگی‌های غیرضروری نیز هست.

رویکرد Metrogo این است که ابتدا ساده‌ترین و کم‌هزینه‌ترین راهکارها اجرا شوند و تنها در صورت نیاز، به سمت راهکارهای پیچیده‌تر حرکت شود. این رویکرد از رشد هزینه‌های غیرضروری جلوگیری می‌کند و با ماهیت یک استارتاپ در مراحل اولیه هم‌خوانی دارد.

تأثیر رشد بر هزینه‌ها

یکی از بخش‌های کلیدی این سند، تحلیل رابطه بین رشد کاربران و هزینه‌های سیستم است . Metrogoبه‌صورت شفاف بررسی کرده که افزایش هر واحد بار چه تأثیری بر هزینه زیرساخت دارد. این تحلیل به تیم کمک می‌کند تصمیم‌های آگاهانه بگیرد و رشد را به‌گونه‌ای مدیریت کند که منجر به فشار مالی ناگهانی نشود.

در این پروژه، هزینه به‌عنوان بخشی از طراحی عملکرد در نظر گرفته شده است، نه نتیجه‌ای ناخواسته. این دیدگاه باعث می‌شود رشد سیستم قابل پیش‌بینی و قابل کنترل باقی بماند.

درMetrogo ، نگاه به عملکرد سیستم صرفاً محدود به زمان پاسخ یا میزان مصرف منابع نیست، بلکه عملکرد به‌عنوان «رفتار سیستم در طول زمان و تحت فشار واقعی» تعریف می‌شود. بسیاری از سیستم‌ها در شرایط آزمایشگاهی عملکرد مناسبی دارند، اما زمانی که با رفتار واقعی کاربران، نوسانات ترافیک شهری و رویدادهای پیش‌بینی‌نشده مواجه می‌شوند، به‌سرعت دچار افت کیفیت می‌شوند. این دقیقاً همان سناریویی است که Metrogo از ابتدا برای آن آماده شده است.

یکی از واقعیت‌های حوزه حمل‌ونقل شهری این است که بار سیستم یکنواخت نیست. کاربران در ساعات خاصی از روز، مانند ابتدای صبح یا پایان ساعات کاری، به‌صورت هم‌زمان به سیستم مراجعه می‌کنند. این الگوی انفجاری ترافیک، اگر به‌درستی مدیریت نشود، می‌تواند حتی سیستم‌هایی با میانگین بار پایین را از کار بیندازد. به همین دلیل، طراحی عملکرد Metrogo بر اساس «پیک بار» انجام شده، نه میانگین استفاده.

نگاه عمیق‌تر به دسترس‌پذیری(Availability)

درMetrogo ، دسترس‌پذیری صرفاً به معنای روشن بودن سرور نیست. سیستم ممکن است از نظر فنی در دسترس باشد، اما اگر پاسخ‌دهی آن به حدی کند شود که کاربر نتواند از آن استفاده کند، از دید تجربه کاربر عملاً غیرقابل دسترس محسوب می‌شود. به همین دلیل، تعریف دسترس‌پذیری در این پروژه با شاخص‌های عملکردی دیگر گره خورده است.

Metrogoدسترس‌پذیری را در سطح قابلیت‌های کلیدی محصول تعریف می‌کند. برای مثال، اگر کاربر نتواند اطلاعات مسیر یا زمان حرکت را دریافت کند، حتی اگر سایر بخش‌های سیستم فعال باشند، تجربه شکست‌خورده تلقی می‌شود. این نگاه باعث می‌شود تیم به‌جای تمرکز صرف بر uptimeزیرساخت، بر در دسترس بودن واقعی ارزش محصول تمرکز کند.

Latencyبه‌عنوان بخشی از تجربه سفر

در پروژهMetrogo ، latencyفقط یک عدد فنی نیست، بلکه بخشی از تجربه سفر کاربر است. تأخیر در دریافت اطلاعات می‌تواند منجر به تصمیم‌های اشتباه، از دست دادن قطار یا افزایش استرس کاربر شود. بنابراین، کاهش latency نه یک بهینه‌سازی لوکس، بلکه بخشی از مسئولیت محصول در قبال کاربر است.

این پروژه latency را در سطوح مختلف بررسی می‌کند؛ از زمان پاسخAPI ها گرفته تا تأخیر در ارتباط با سرویس‌های وابسته. این بررسی چندلایه به تیم اجازه می‌دهد بفهمد تأخیر از کجا ناشی می‌شود و آیا با افزایش بار، این تأخیر به‌صورت خطی افزایش می‌یابد یا به‌صورت ناگهانی جهش می‌کند.

تحلیل رفتار سیستم در شرایط ناپایدار

یکی از بخش‌های مهم برنامه عملکردMetrogo ، بررسی رفتار سیستم در شرایط ناپایدار است. این شرایط شامل قطع یا کندی سرویس‌های خارجی، افزایش ناگهانی بار یا حتی خطاهای انسانی در عملیات است. هدف این تحلیل‌ها این است که مشخص شود سیستم چگونه به خطا واکنش نشان می‌دهد.

Metrogoبه‌دنبال طراحی سیستمی است که در مواجهه با مشکل، به‌جای فروپاشی کامل، به‌صورت کنترل‌شده کیفیت خود را کاهش دهد. این مفهوم که به آن «شکست «graceful گفته می‌شود، نقش مهمی در حفظ اعتماد کاربران دارد. کاربر ممکن است کندی جزئی را بپذیرد، اما قطعی کامل معمولاً منجر به ترک محصول می‌شود.

ظرفیت‌سنجی پویا به‌جای ایستا

در بسیاری از پروژه‌ها، ظرفیت‌سنجی به‌صورت ایستا و در ابتدای کار انجام می‌شود و سپس به فراموشی سپرده می‌شود Metrogo .این رویکرد را ناکارآمد می‌داند. در این پروژه، ظرفیت‌سنجی به‌عنوان یک فرآیند پویا تعریف شده که باید با رشد محصول و تغییر رفتار کاربران به‌روزرسانی شود.

این رویکرد پویا باعث می‌شود تصمیم‌های زیرساختی بر اساس داده‌های واقعی گرفته شوند، نه حدس و گمان. همچنین به تیم کمک می‌کند نقاط بحرانی را پیش از تبدیل شدن به بحران شناسایی کند.

Load Testبه‌عنوان ابزار یادگیری، نه تأیید

درMetrogo ، آزمون بار تنها برای اثبات «کار کردن سیستم» انجام نمی‌شود. این آزمون‌ها به‌عنوان ابزار یادگیری طراحی شده‌اند. تیم با اجرای تست‌ها، رفتار سیستم را مشاهده می‌کند و از نتایج آن برای بهبود معماری و تنظیمات استفاده می‌کند.

نتایج این تست‌ها مستند می‌شوند و به‌عنوان مرجع تصمیم‌گیری‌های بعدی مورد استفاده قرار می‌گیرند. این مستندسازی کمک می‌کند تیم از تکرار اشتباهات گذشته جلوگیری کند و دانش عملکرد سیستم در طول زمان حفظ شود.

مدیریت گلوگاه‌ها در مسیر رشد

Metrogoمی‌پذیرد که با رشد سیستم، گلوگاه‌های جدیدی ظاهر خواهند شد. به‌جای تلاش برای حذف تمام گلوگاه‌ها از ابتدا، تمرکز پروژه بر شناسایی زودهنگام و مدیریت تدریجی آن‌هاست. این رویکرد واقع‌بینانه با منابع محدود یک استارتاپ هم‌خوانی دارد.

هر گلوگاه شناسایی‌شده همراه با سناریوهای رشد و راهکارهای احتمالی مستند می‌شود. این مستندات به تیم کمک می‌کند در زمان نیاز، تصمیم سریع و آگاهانه بگیرد.

مقیاس‌پذیری به‌عنوان انتخاب آگاهانه

درMetrogo ، مقیاس‌پذیری نه به‌عنوان هدف نهایی، بلکه به‌عنوان ابزار پشتیبانی از رشد تعریف شده است. این پروژه به‌جای پیاده‌سازی راهکارهای پیچیده از ابتدا، به‌دنبال ساخت سیستمی است که بتواند به‌صورت مرحله‌ای و بر اساس نیاز واقعی رشد کند.

این انتخاب آگاهانه باعث می‌شود سیستم هم قابل نگهداری بماند و هم هزینه‌های آن تحت کنترل باشد. مقیاس‌پذیری در Metrogo به‌معنای آماده بودن برای رشد است، نه الزام به رشد بی‌محابا.

هزینه عملکرد و مقیاس‌پذیری

یکی از نگرانی‌های اصلی در پروژه‌های مبتنی بر زیرساخت، افزایش ناگهانی هزینه‌ها در اثر رشد است Metrogo .این موضوع را به‌صورت شفاف بررسی کرده و تلاش کرده است رابطه بین بار سیستم و هزینه را قابل پیش‌بینی کند.

این شفافیت به تیم اجازه می‌دهد رشد را به‌گونه‌ای برنامه‌ریزی کند که هم پاسخ‌گوی نیاز کاربران باشد و هم فشار مالی غیرقابل مدیریت ایجاد نکند.

جمع‌بندی

این سند نشان می‌دهد که Metrogo عملکرد و مقیاس‌پذیری را نه به‌عنوان یک موضوع فنی صرف، بلکه به‌عنوان ستون اصلی اعتماد کاربران و پایداری کسب‌وکار می‌بیند. تعریف شفاف SLI وSLO ، ظرفیت‌سنجی واقع‌بینانه، برنامه تست بار، شناسایی گلوگاه‌ها و تحلیل هزینه رشد، همگی نشان‌دهنده بلوغ تیم در مواجهه با رشد هستند.

Metrogoبا این رویکرد ثابت می‌کند که رشد کاربران و تراکنش‌ها سیستم را از پا درنمی‌آورد، بلکه با طراحی آگاهانه و مهندسی‌شده، به فرصتی برای تقویت محصول و کسب‌وکار تبدیل می‌شود.