

شروع

اولش خیلی ساده شروع کردم و یه پروژه جنگو ساختم و فقط یه اپ بهش اضافه کردم به اسم posts. بعد سعی کردم کم کم نیازمندی‌ها رو پوشش بدم. مدل‌هام Post و Rating هستن. هر Rating یه کلید خارجی به پست و یه کلید خارجی به یوزر داره. اولش خیلی ساده هر پست title و content داشت. هر rating هم به جز کلید خارجی‌ها مقدار رای رو نگه می‌داشت. توی این حالت برای این که تعداد کاربرها و میانگین امتیازهارو دربیارم باید برای هر پست کوئری می‌زدیم هر بار. این کوئریه هم چیز سنگینی بود. مخصوصا این که وقتی صفحه لود میشه به ازای همه‌ی پست‌ها باید این کوئری رو بزنیم و توی لود صفحه‌ی اول اصلا خوب کار نمی‌کرد. رای‌های بر اساس یوزر/پست یونیک بودن و اینجوری کسی رای تکراری نمی‌تونست ثبت کنه. از طرفی وقتی رای می‌داد آپدیتش می‌کردم. تا اینجا همه‌ی نیازمندی‌های اولیه ساپورت می‌شد. ولی روی لود بالا خوب کار نمی‌کرد.

تعداد زیاد رای برای یک پست

برای این که توی لود بهتر عمل کنم اومدم جمع و تعداد رای‌های هر پست رو توی هر پست نگه داشتم. با این که دیتای ریداندنت بود ولی با این دی‌نرمالایزیشن پرفرمنس از On به O1 رسید. فقط باید حواسم می‌بود که موقع آپدیت رای‌ها و یا رای جدید این مقادیر رو آپدیت نگه دارم که اینکانسیستنسی پیش نیاد. یه مورد مهم دیگه هم این بود که همه‌ی این موارد باید اتمیک انجام می‌شدن که باز هم جلوی اینکانسیستنسی گرفته بشه. الان مشکل تعداد زیاد رای برای یه پست و محاسبه‌ی سنگین اون بخش حل شد.

امتیازات هیجانی

برای حل مشکل امتیازات هیجانی هم روش EMA زدم.

$$\text{smoothed_rating}(t) = \alpha \times \text{arithmetic_avgt} + (1 - \alpha) \times \text{smoothed_rating}(t-1)$$

اینجوریه که هنوز به میانگین عادی نیاز داریم ولی تاثیر هیستوری توش رو می‌شه زیاد و کم کرد. اگه آلفا رو کم بذاریم تاثیر هیجانات تا حد خوبی هندل میشه.

این روش هم خیلی اورهد زمانی نداره و هم خیلی اوردهد حافظه.

یه فیلد به تبیل پست یه فیلد smoothed_rating اضافه کردم. برای آپدیتش هم یه پراسسی چند دقیقه یه بار اجرا میشه و امتیاز همه‌ی پست‌هارو آپدیت می‌کنه. اینجوری موقع خوندن اورهدی نداریم. آلفا رو هم گذاشتم ۰.۱ که تاثیر اتفاقات لحظه‌ای و هیجانی کمتر بشه.

برای بهتر کردنش هم میشه آلفا رو داینامیک کرد. مثلاً توی حالت‌هایی که هیجان تشخیص داده شد آلفا کوچک‌تر باشه.

لود چند هزار تایی

برای لود چند هزار توی ثانیه هم تصمیم داشتم که دیتابیس رو شارژ کنم. خیلی تلاش کردم ولی موفق نشدم. از اونجایی که پست‌ها کاملاً از هم مستقل‌ان میشه پست‌هارو توی دیتابیس‌های مختلف ریخت مثلاً بر حسب آیدی. فقط باید حواسمون باشه رای‌های مربوط به هر پست توی همون شارژ باشه.

مانیتورینگ

توی این راه به طور مداوم پرفرمنس رو با locust مانیتور می‌کردم. به ۵۰۰ rps رسید سیستم. مثلاً برای تست روش هندل هیجانات اومدم یه لود انداختم که ۲۰۰ تا یوزر رای‌شون به یه پست رو صفر کنن. یه بار هم برای ۵ این کار رو کردم. اینجوری بود که میانگین پایین می‌اومد ولی حول و حوش میانگین قبلی می‌موند و به صفر نزدیک نمی‌شد. لذت‌بخش‌ترین بخش چالش اینجاش بود.