

Hedwig

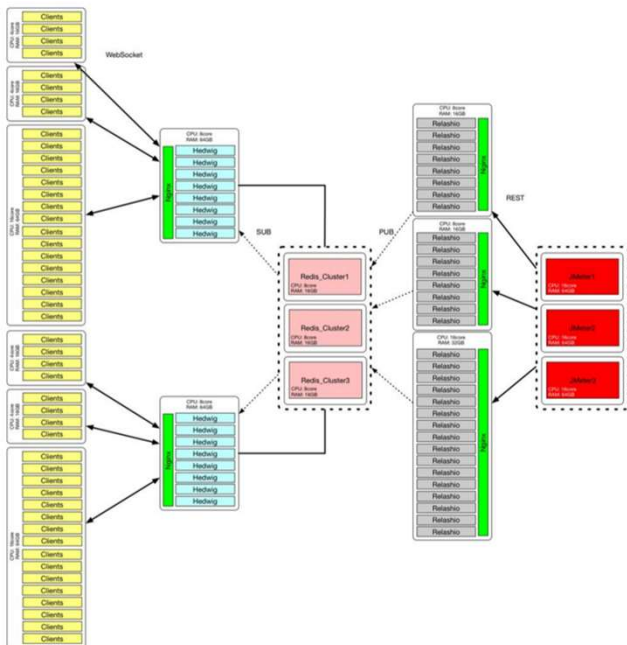


دانشجو: علی محمد فروتن نژاد
استاد راهنما: دکتر مهدی کارگهی
دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران



نتایج

- حداکثر تحمل هر Instance Relashio برابر 22K RPS
- حداکثر تعداد کانکشن هر Instance Hedwig برابر 60K
- حداکثر CPU مصرفی Redis Cluster برابر 20%
- عدم وجود پیغام رسیده نشده
- گسترش پذیری خطی هر دو سرویس hedwig و relashio



شرح مساله

در معماری نرم افزار یکی از مهم ترین مساله ها گسترش پذیری بر اساس رشد تعداد کاربرهاست که نیازمند شناسایی دقیق گلوگاه ها، آشنایی کامل با دامنه مساله و استفاده از ابزارهایی بهینه و کارا برای پاسخ به آن است. در حال حاضر یکی از مشکلاتی که در شرکت های نوپای کشور دیده می شود استفاده از متدولوژی های قدیمی برقراری ارتباط بین کلاینت ها و سرورهاست که باعث مشکلات زیر می شود

- بالا رفتن زمان پاسخ در حالتی که سیستم زیر فشار می رود
- بالا رفتن هزینه سخت افزار به دلیل عدم استفاده بهینه از آن
- سخت تر شدن کنترل Data Flow به دلیل معماری Monolithic
- بالاتر رفتن استفاده از پهنای باند به دلیل استفاده از JSON

معماری نرم افزار

طراحی معماری این سیستم سخت ترین کار در طول این پروژه بود. طی چندین مرحله معماری ساخته، تست و عوض شد. مهمترین اصلی که در این معماری در تلاش بودیم رعایت شود، گسترش پذیری سیستم بود. چرا که به طور معمول ۸ هزار درخواست در ثانیه به سمت سرورهای اسنپ می آید که با توجه به گسترش فعالیت شرکت پیش بینی می شود که تا ۱۵ هزار درخواست در ثانیه هم بالا رود. به همین دلیل معماری این سیستم طی چندین مرحله و با انجام تست های فراوان نهایی شد. در این معماری ۳ سرویس اصلی وجود دارند و از Redis به عنوان State Manager استفاده می شود.

جمع بندی

با توجه به معماری مورد بحث و همچنین تست های بیشماری که روی نرم افزار انجام شد می توان نتیجه گرفت که این سرویس قابلیت گسترش پذیری خطی دارد و می توان در محیط های عملیاتی واقعی از آن استفاده کرد.

مراجع اصلی

- <https://www.grpc.io>
- <https://medium.freecodecamp.org/million-websockets-and-go-cc58418460bb>
- https://docs.google.com/document/d/1mL9EF6qKrk7gbazY8bl_dvq3Pno2Xj_I_YShP40GLQE/edit

