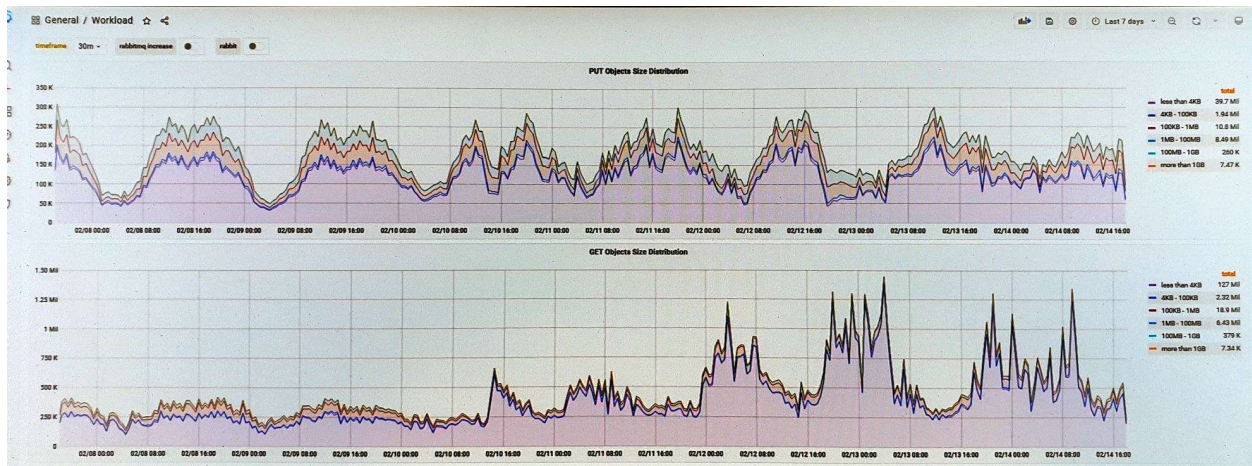


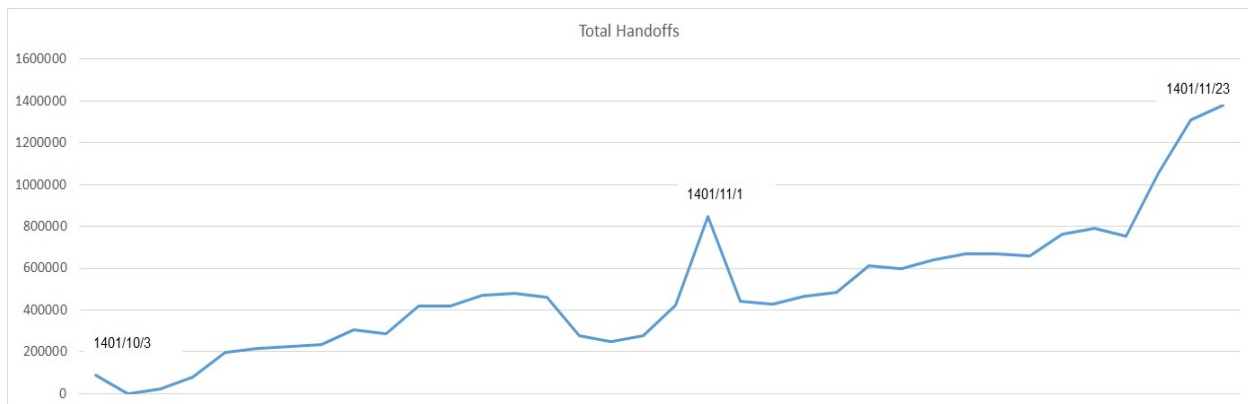
بنام خدا

پیوست فنی سمفونی ۲

در حال حاضر کلاستر با الگوی بار کاری زیر در حال کار است. حدود ۷۵ درصد از داده ها را ریز فایل ها (زیر ۱ مگابایت) تشکیل می دهد که در مجموع حدود یک درصد حجم کلاستر را به خود اختصاص داده اند. نرخ زیاد ورود ریز فایل ها به سامانه از یک طرف سبب اشباع توان IOPS دیسک های HDD در کل سامانه شده و حدود ۸۰ درصد پهنای باند دیسک ها را بلااستفاده باقی می گذارند، و از طرفی دیگر اثر زیادی بر مصرف منابع سرورهای حساب و کیسه که حاوی فراداده هستند می گذارد و ایجاد سازگاری آنها را در ۵ نسخه با تاخیر بسیار زیاد مواجه می کند. اشباع شدن منابع سامانه سبب استفاده از منابع رزرو به منظور تحمل اضافه بار می شود که خود را در قالب ایجاد handoff نشان می دهد. با توجه به الگوی بار سینوسی با تناوب یک روزه، چنانچه ناپایداری در دوره های تناوب یک روزه به گونه ای برطرف نشود که handoff ها بجای خود برگردند و سرورهای فراداده با همدیگر سازگار شوند، در نهایت سبب ایجاد داده های تاریک خواهد شد و در نتیجه آن ظرفیت حجمی کلاستر را به تدریج مصرف می کند. علاوه بر موارد فوق، به دلیل اشیاء شدن منابع سروهای کیسه، حذف اشیاء منقضی شده نیز با تاخیر همراه خواهد شد که آزاد شدن ظرفیت حجمی کلاستر توسط سرویس object_expirer را با تاخیر همراه کرده است.



وضعیت بار کاری PUT و GET بر اساس توزیع سایز اشیاء در سامانه سمفونی از تاریخ ۱۹ الی ۲۵ بهمن



آمار افزایش تعداد پارتیشن های رزرو در سامانه سمفونی از ابتدای دیماه ۱۴۰۱

خلاصه نیازمندی های شناسایی شده برای سمفونی ۲

- بهار ۱۴۰۲

- ایجاد رینگ جدید جهت کاهش نرخ تکرار کیسه ها از ۵ به ۳ جهت تسریع در عملیات سازگاری
- توسعه ماشین حساب عمومی جهت محاسبه LOM متناسب با مقیاس بارکاری
- ارتقای سامانه ELK (اصلاح زیرساخت، ارتقای نسخه ELK، افزایش ظرفیت انبارش سامانه به حداقل سه ماه)
- توسعه نشانگرهای سطح آستانه کارایی و ظرفیت سامانه (IOPS/BW)
- ارایه راهکار خروج لاگ بصورت گمنام شده
- استقرار MonsterAI_v0.1

- تابستان ۱۴۰۲

- پیاده سازی قابلیت چابک (CAS) جهت کش ریز فایل ها و کاهش بار کاری دیسک های HDD
- ارتقای نسخه هیولا
 - بهبود در عملکرد توازن بار در اتصالات بین کلاینت و پراکسی
 - رفع باگ سکنه پراکسی در زمان جمع آوری لاگ thread ها
 - تقویت auditor جهت شناسایی داده های تاریک
 - بهبود کارایی object-updater جهت پردازش موازی و سازگاری سریعتر
 - امکان محدود سازی نرخ object-expirer در هر ثانیه
 - قابلیت socket_timeout جهت خاتمه دادن به سوکت های بیکار
 - بهبود مدیریت کش برای پاسخ های حساب و کیسه
 - بهبود کارایی انتقال های object-replicator

- پاییز ۱۴۰۲

- اصلاح منطق پوشه بندی در سمت بهره بردار در قالب ایجاد کیسه های روزانه و سبک شدن حجم کیسه ها
- استفاده از Symlink جهت کاهش افزونگی داده ها در عملیات کپی
- پیاده سازی مکانیزم توازن بار (HA) جهت بهبود کارایی در سطح پراکسی و افزایش تحمل پذیری خطا
- پیاده سازی قابلیت Ratelimit جهت ایجاد محدودیت برخی از عملیات سامانه

- زمستان ۱۴۰۲

- ارایه طرح هاردنینگ هیولا
- استقرار MonsterAI_v1.0