

Performance & Scalability Plan

۴/۸/۱ مقدمه

باید در صورت رشد کاربران و افزایش در خواست‌ها، عملکرد `nobatnou.ir` سیستم نوبتدھی قابل قبول و در دسترس بودن بالا را حفظ کند. این سند اهداف عملکردی، نقاط ضعف شناخته شده، استراتژی‌های مقیاس‌پذیری و برنامه‌ تست بار را تشریح می‌کند.

۴/۸/۲) اهداف سطح سرویس (Service Level Objectives – SLOs)

هدف روش اندازه‌گیری SLO SLI (SLO) شاخص سطح سرویس
از طریق uptime (ماهانه) مانیتورینگ ۹۹/۵٪ در دسترس بودن** HTTP checks
مانند APM ثانیه برای ۹۵٪ درخواست‌ها ابزارهای ۲ < زمان پاسخگویی** New Relic یا لاجهای Nginx
برای عملیات نوبت‌گیری شمارش ۹۹٪ (Success Rate)** نرخ موفقیت** response codes 2xx/3xx vs 4xx/5xx
پشتیبانی از ۱۰۰ کاربر همزمان بدون افت ظرفیت همزمان** Concurrent Users) sessions عملکرد تست بار و مانیتورینگ فعال

های فعلی بر اساس زیرساخت فعلی (هاست اشتراکی/서버 공유) تعریف شده و با SLO **:توجه مقیاسگذاری بهبود خواهد یافت.

۴/۸/۳ (Bottlenecks)**:معماری فعلی و نقاط تنگنا

**:معماری فعلی

- **Frontend:** WordPress با پوسته و پلاگین‌های سفارشی.

- **Backend:** PHP با پردازش سنگین در Ajax handlerها.

- MySQL با `nobat_system` و `nobat_users` دیتابیس.

- ذخیره‌سازی: فایل‌های وردپرس و آپلودها روی همان سرور.

- ، کش پلاگین وردپرس (محدود client-side) کش: کش مرورگر.

#: نقاط تنگنا شناسایی شده

| نقطه تنگنا | تأثیر بر عملکرد | راهکار کوتاه‌مدت | راهکار بلندمدت |

| ----- | ----- | ----- | ----- | ----- |

| در بار زیاد | بهینه‌سازی کوئری‌ها، افزودن latency پرس‌جوهای دیتابیس سنگین | افزایش index تفکیک دیتابیس | Read/Write، Replication |

| فعال‌سازی کش شیء | MySQL و PHP عدم کش در سطح سرور | بار اضافه روی | استفاده از کش کامل صفحه Redis | (Object Cache) Full Page Cache |

| PHP-MSDOD شدن درخواست‌های همزمان | محدودیت همزمانی در Ajax | پردازش همزمان FPM | (Queue) انتقال پردازش به صف کار |

| ** هاست اشتراکی/منابع محدود** | کاهش عملکرد با رشد کاربران | ارتقاء به سرور مجازی (VPS) |
| مهاجرت به زیرساخت ابری (Cloud) |

۴/۸/۴ برنامه مقیاس‌پذیری (Scalability Roadmap)**

فاز ۱: بهینه‌سازی (۰-۳ ماه) **## #

- کد:**

کاهش تعداد کوئری‌های دیتابیس در هر درخواست -

برای داده‌های ثابت (Redis/Memcached) افزودن کش در حافظه -

- دیتابیس:**

- ایجاد index روی ستون‌های پرکاربرد (`phone_number`, `position`, `status`).

- تنظیم پارامترهای `innodb_buffer_pool_size` و `query_cache`.

- سرور:**

- مناسب `pm.max_children` با PHP-FPM تنظیم

- فعال‌سازی opcode cache (OPcache).

۶-۳ ماه فاز ۲: مقیاس عمودی (Vertical Scaling)

- ارتقاء سرور به ماشین با -

- هسته ۴ CPU:

- RAM: ۸ گیگابایت

- گیگابایت ۱۰۰: SSD:

- مهاجرت دیتابیس به سرور جداگانه (Database Server).

##:# # مقیاس افقی (Horizontal Scaling) فاز ۳: (ماه ۶-۱۲)

- PHP برای توزیع بار بین چند سرور HAProxy یا Nginx یا لود بالانسر استفاده از **:

- **: دیتابیس

- (برای خواندن slave برای نوشتن، یک یا چند master یک) replication راه اندازی.

- (`nobat_system` پارتیشن بندی جدول `created_at`) بر اساس تاریخ.

- و داده های پر تکرار session Redis cluster برای کش **:

##:# # معماری مبتنی بر ابر (Cloud-Native) فاز ۴: (ماه ۱۲+)

- انتقال به زیر ساخت ابری AWS, DigitalOcean).

- استفاده از سرویس های مدیریت شده

- RDS برای دیتابیس.

- ElastiCache برای کش.

- CloudFront/CDN برای استاتیک.

- برای مأذول های مستقل (ثبت نام، نوبت دهی، پیام ک) طراحی microservices.

##:# # برنامه تست بار (Load Testing Plan) ** ۵/۸/۴

**: اهداف تست بار # ## # #

- قابل پشتیبانی (concurrent users) شناسایی حداکثر کاربر همزمان.
- و رفتار سیستم تحت بار بیش از حد (breakpoints) تعیین نقاط شکست.
- تأیید بهبودها پس از اعمال بهینه‌سازی.

**: ابزارها # ## # #

- برای شبیه‌سازی کاربران Apache JMeter یا k6.
- برای مانیتورینگ منابع New Relic یا Datadog.

**: سناریوهای تست بار # ## # #

**: سناریو ۱: بار عادی (Normal Load)

- کاربر همزمان در مدت ۱۰ دقیقه ۵۰.
- عملیات: مشاهده صفحه اصلی، ورود، مشاهده نوبت.
- زیر ۲ ثانیه، نرخ خطای latency < ۱٪: انتظار.

**: سناریو ۲: بار سنگین (Peak Load)

- کاربر همزمان در مدت ۵ دقیقه ۱۵۰.
- عملیات: رزرو نوبت همزمان (۲۰ درخواست در ثانیه).
- زیر ۳ ثانیه، نرخ خطای latency < ۵٪: انتظار.

**: سناریو ۳: استرس (Stress Test)

- کاربر همزمان تا زمان شکست سیستم ۳۰۰ -

.هدف: یافتن حد ظرفیت سیستم -

**: معیارهای اندازه‌گیری #####

- **: زمان پاسخگویی p95، p99.

- **: نرخ درخواست در ثانیه (RPS).

- **: استفاده از CPU، RAM، I/O.

- **: تعداد اتصالات همزمان به دیتابیس.

**: فرکانس اجرا #####

- پس از هر تغییر عمده (ماهانه).

- قبل از انتشار نسخه جدید.

- (در مرحله بعدی) CI/CD به صورت خودکار با -

**: Capacity Planning) ظرفیت‌سنجی و پیش‌بینی رشد ۴/۸/۶

**: پارامترهای کلیدی #####

- نفر (پیش‌بینی سال اول) ۵۰۰ (**: DAU) کاربران فعال روزانه -

- **: HTTP درخواست‌ها در روز: ~ ۱۰,۰۰۰ درخواست -

- **: ۲۰٪ (پیش‌بینی) رشد ماهانه:

** محاسبه منابع مورد نیاز (بر اساس پیش‌بینی ۱۲ ماهه) # #####

ماه کاربران فعال روزانه درخواست روزانه سرور پیشنهادی هزینه ماهانه تخمینی					
۰ ۱۰۰	۲,۰۰۰	VPS 2GB	دلار ۱۵		
۶ ۵۰۰	۱۰,۰۰۰	VPS 4GB + DB	دلار ۴۰		
	۱۲ ۱,۵۰۰	دو سرور ۸۰ دلار + ۴GB			
	۱۸ ۵,۰۰۰	۳۰,۰۰۰ دلار	۲۰۰ دلار	زیرساخت ابری (۳ سرور)	

توجه: هزینه‌ها بر اساس ارائه‌دهنگان داخلی/خارجی تخمین زده شده است**

۴/۸/۷ هشدار و مانیتورینگ (Monitoring & Alerting)

** معیارهای مانیتورینگ #####

- ** منابع سرور **: RAM > ۹۰٪، CPU > ۸۰٪، Disk I/O > ۱۰۰ms.

- ** خطاهای **: اپلیکیشن PHP کوئری‌ها

- ** slow queries **: کوئری‌های کند، تعداد اتصالات فعال، دیتابیس (slow queries).

- **SSL شبکه:** ترافیک ورودی/خروجی، وضعیت**

:ابزارها**

- **Server:** Netdata, Prometheus + Grafana.

- **Application:** Sentry (برای خطاها), New Relic (برای performance).

- **External Uptime:** UptimeRobot, Pingdom.

:هشدارها**

- **SLO (Critical):** اعلام از طریق (Zir ۹۵٪، سرور down) شدید** Telegram/Email)

- **latency (Warning):** اعلام از طریق (بالای ۷۰٪ CPU بالای ۳ ثانیه، استفاده هشدار Email)

۴/۸/۸ (Performance Degradation Recovery)**

:اقدامات فوری**

بررسی لگ‌ها: شناسایی درخواست‌های سنگین یا کند** . ۱.

(در صورت قفل) MySQL، PHP-FPM: ری‌استارت سرویس‌ها** . ۲.

پاکسازی کش: کش دیتابیس یا کش صفحات** . ۳.

کاهش overload: موقت برای جلوگیری از rate limit محدودسازی ترافیک:** . ۴.

**: اقدامات بلندمدت پس از حادثه #####

- تحلیل ریشه‌ای (Root Cause Analysis – RCA).

- بروزرسانی مستندات و runbooks.

- اعمال تنظیمات پیشگیرانه.

** (آینده Auto-scaling Plan –) برنامه مقیاس خودکار # # # # # ۴/۸/۹

* پیش‌نیاز: مهاجرت به زیرساخت ابری **

** (Instances) ماشین‌های مجازی **

- گروه (Group): ** سرور‌های PHP (Frontend).

- بعیار مقیاس ** CPU > ۷۰٪: مدت ۵ دقیقه -

- ۵ حداکثر: instance.

** دیتابیس:

- استفاده از سرویس مدیریت شده AWS RDS با قابلیت auto-scaling storage/IOPS.

۴/۸/۱۰) هزینه‌های مقیاس‌پذیری (Scalability Costs)

مرحله اقدامات هزینه ماهانه تخمینی تأثیر بر عملکرد
----- ----- ----- -----
تنظیمات ~ ۱۰۰ هزار تومان (نیروی انسانی) بهبود index بهینه‌سازی** کش، **
----- ----- ----- -----
قوی‌تر ۶۰–۴۰ دلار بهبود ۲۰۰٪–۱۰۰٪ VPS مقیاس عمودی** ارتقاء سرور به**
----- ----- ----- -----
مقیاس افقی** افزودن سرور و لود بالانسر ۲۰۰ دلار بهبود ۳۰۰٪–۵۰۰٪**
----- ----- ----- -----
ابر** مهاجرت به زیرساخت ابری ۵۰۰–۲۰۰ دلار بهبود ۱۰۰٪ + قابلیت اطمینان بالا**

توجه: ** هزینه نیروی انسانی برای پیکربندی و نگهداری جدگانه محاسبه شده است**.

۴/۸/۱۱) نتیجه‌گیری

با معماری فعلی می‌تواند تا **۱۰۰ کاربر همزمان** را با عملکرد قابل 'nobatnou.ir' سیستم قبول پشتیبانی کند. با اجرای برنامه بهینه‌سازی و مقیاس‌پذیری مرحله‌ای، سیستم قادر خواهد بود تا های تعریف شده مدیریت کند. کلید موفقیت، **مانیتورینگ SLO ۵۰۰۰** کاربر همزمان** را با حفظ مستمر، تست بار دوره‌ای و انعطاف‌پذیری در انتخاب زیرساخت** است. این برنامه بهگونه‌ای طراحی شده که هزینه‌ها با رشد درآمد کسب‌وکار افزایش یابد.