

aselsan

www.aselsan.com.tr

ASELSAN Türk Silahlı Kuvvetlerini Güçlendirme Vakfı'nın bir kuruluşudur.



Liderlik Tablosu Uygulamasında In-Memory Repository Karşılaştırması

Temmuz 2025

Hazırlayan: Yusuf Emre Bayrakcı

Sorumlu Mühendis: Ömer Esas

Staj Süresi: 16.06.2025 – 11.07.2025 (20 iş günü)

Staj Birimi: SST Hava ve Füze Savunma Komuta Kontrol

Yazılım Tasarım Müdürlüğü

- ❖ Gazi Üniversitesi Mühendislik Fakültesi
- ❖ Bilgisayar Mühendisliği
- ❖ 3. Sınıf
- ❖ GPA: 3.57

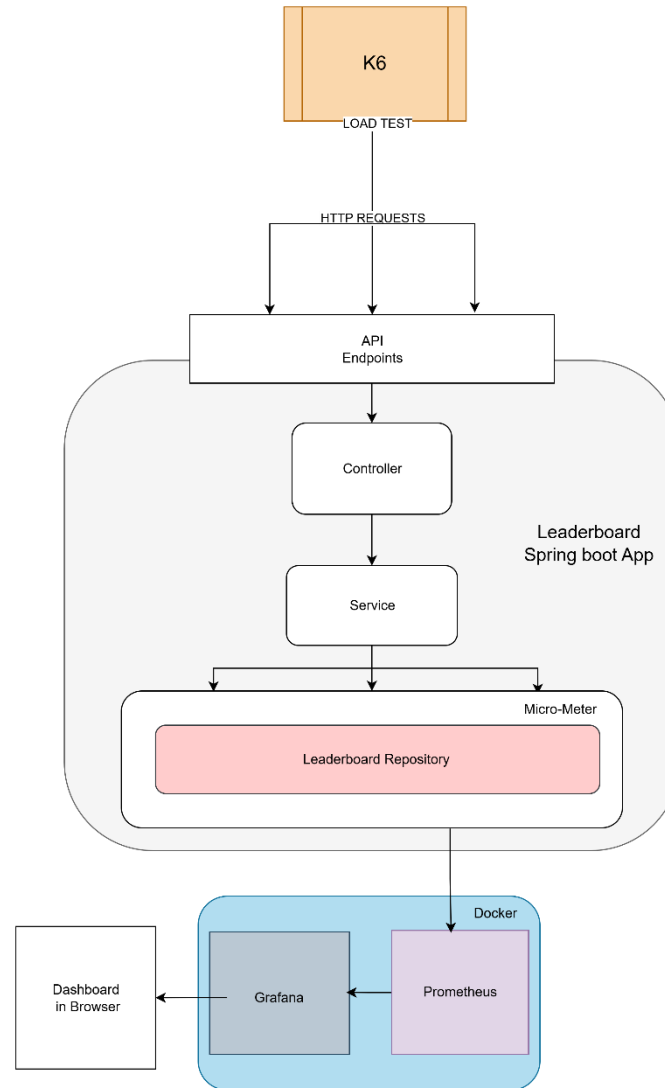


İÇİNDEKİLER

1. Staj Dönemi Özeti
2. Projede Kullanılan Teknolojiler
3. Staj Projesi
4. Kazanımlar
5. Soru-Cevap

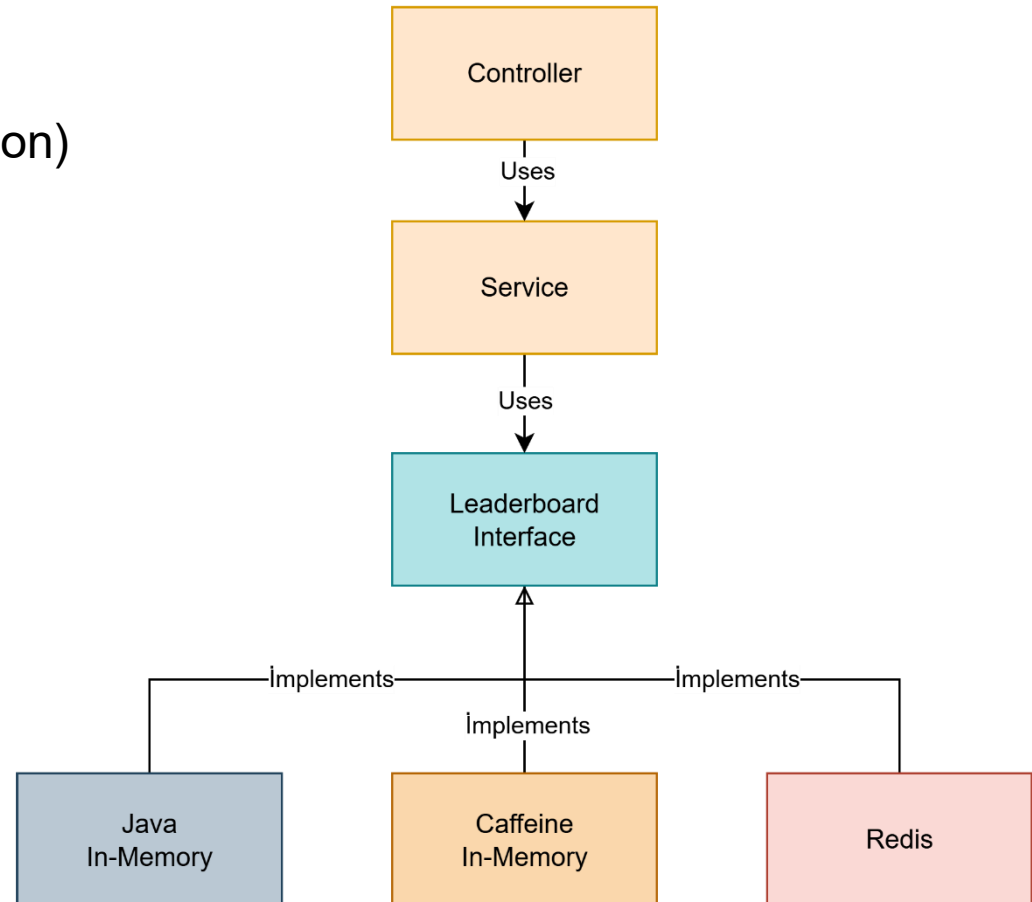
- Hafta-1: Kayıt, İSG ve Bilgisayar kurulumu, Docker ve Spring Boot Başlangıç
- Hafta-2: Leaderboard Redis-Caffeine-Java REST Backend Uygulaması
- Hafta-3: Leaderboard uygulaması benchmark implementasyonu
- Hafta-4: Testlerin yapılması, Sunum hazırlığı, Staj raporu yazımı

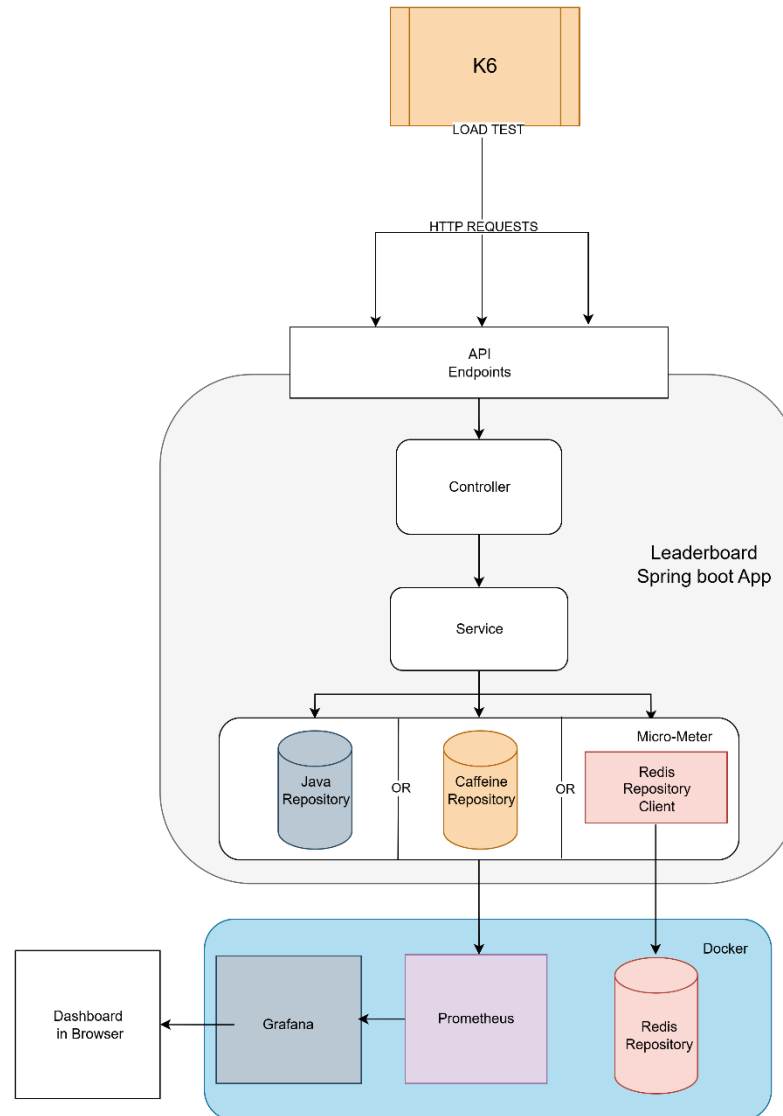
- ❖ **Java** ve **Spring Boot** ile RESTful servis geliştirdik.
- ❖ **Docker** aracılığıyla konteynerize edilen teknolojiler ayağa kaldırıldık.
- ❖ **Micro-meter** ile uygulama içi metrikler toplandı.
- ❖ **Prometheus** bu metrikleri zaman serisi olarak depoladık.
- ❖ **Grafana** üzerinden metrikler görselleştirdik.
- ❖ **k6** ile API'lara yönelik yük testleri (load test) çalıştırdık.



Lider Tablosu Repository olarak Kullanılan 3 Veri Yapısı

- ❖ 1 Arayüz (Interface)
- ❖ 3 Gerçekleştirim (Implementation)
 1. Java
 2. Caffeine
 3. Redis





Problem Tanımı

- Çok oyunculu oyunlarda **lider tablosu**, rekabetin merkezi bir bileşenidir.
- Saniyede **binlerce skor güncellemesi** ve **sorgu** oluşur.
- Gecikmesiz erişim, anlık sıralama ve yüksek performans gereklidir.
- Bu proje, bu soruna **in-memory çözümlerle yaklaşmayı** hedeflemektedir.



Amaç

- ❖ Lider tablosu (**Leaderboard use case**) problemini **In-Memory veri yapılarıyla çözümünü** performans, esneklik ve ölçeklenebilirlik açısından farklarını gözlemlemek
- ❖ Aynı işlevi gören 3 farklı implementasyonu (**Plain Java, Caffeine, Redis**) lider tablosu kullanım senaryosunda karşılaştırmak

Senaryo

- **K6**

→ REST API üzerinden skor ekleme, sıralama sorgulama isteklerini gönderir

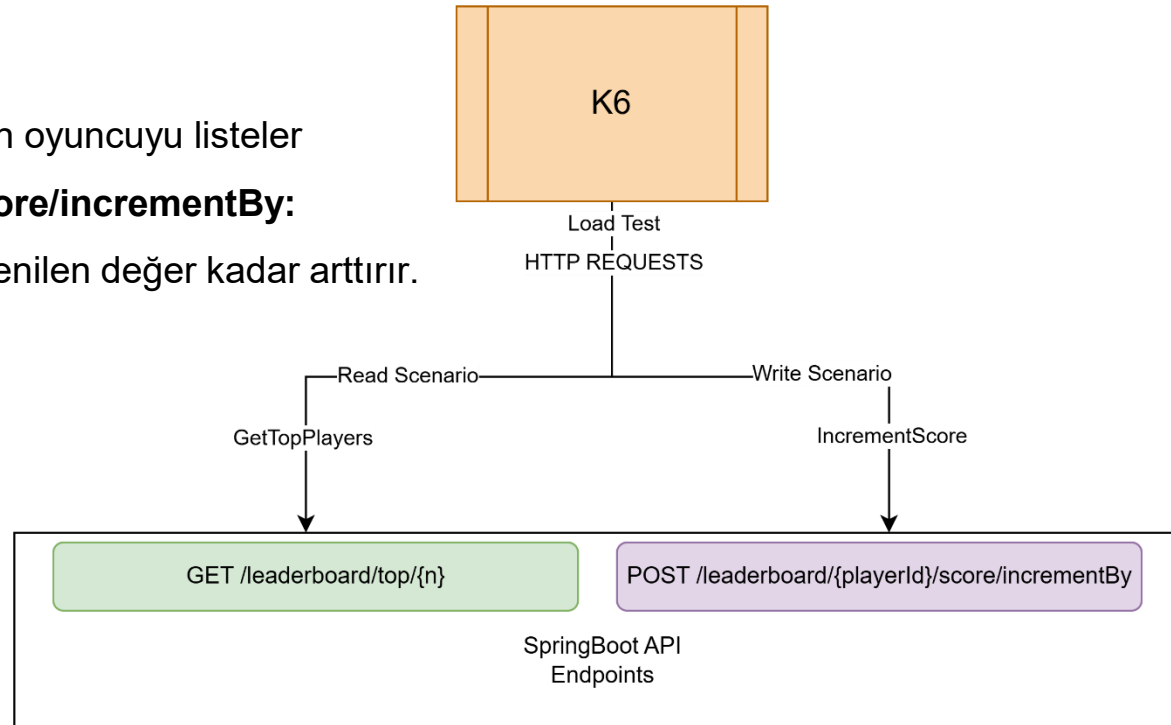
API Endpointler

- **GET /leaderboard/top/{n} :**

En yüksek puana sahip ilk n oyuncuyu listeler

- **POST /leaderboard/{playerId}/score/incrementBy:**

İlgili oyuncunun puanını istenilen değer kadar artırır.



Repository Gerçekleştirimlerinde Lider tablosu veri yapıları

❖ Java

HashMap

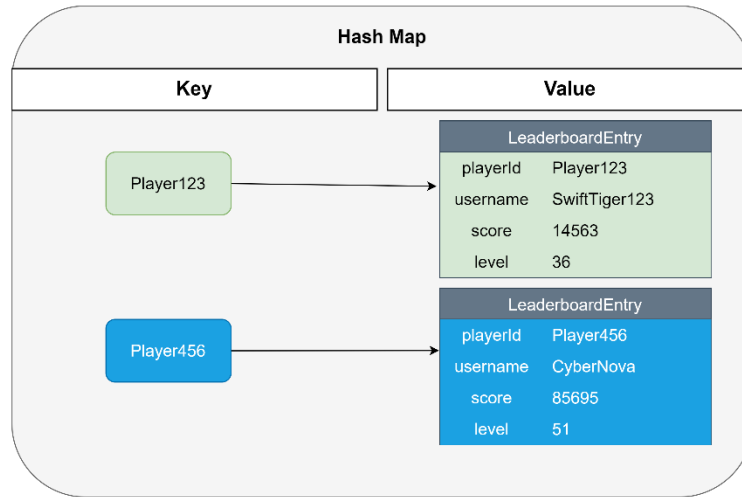
❖ Caffeine

Cache(Concurrent Hashmap)

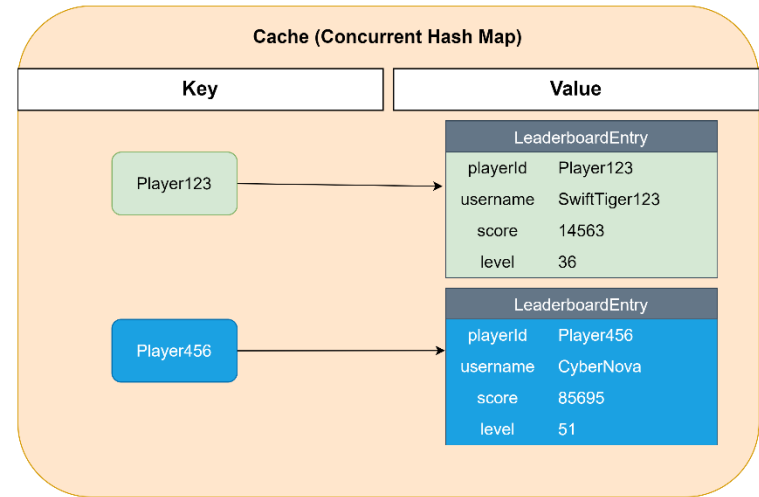
❖ Redis

Sorted Set + Hash

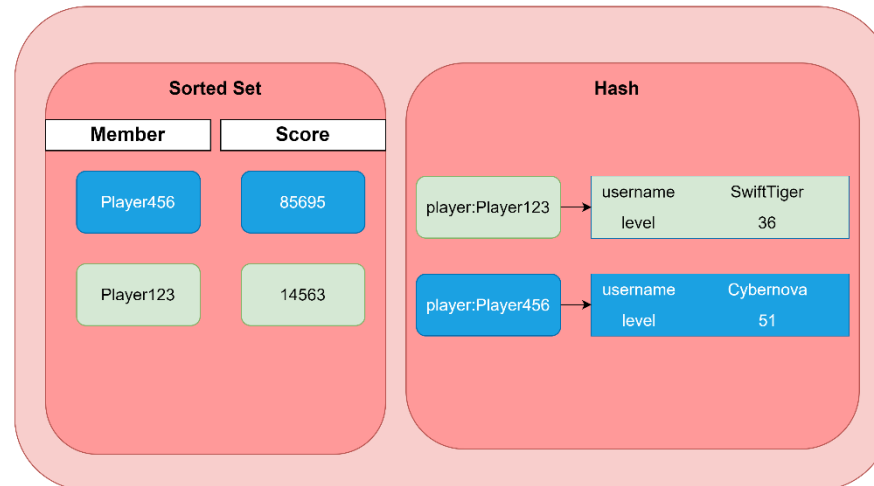
Java Plain



Caffeine

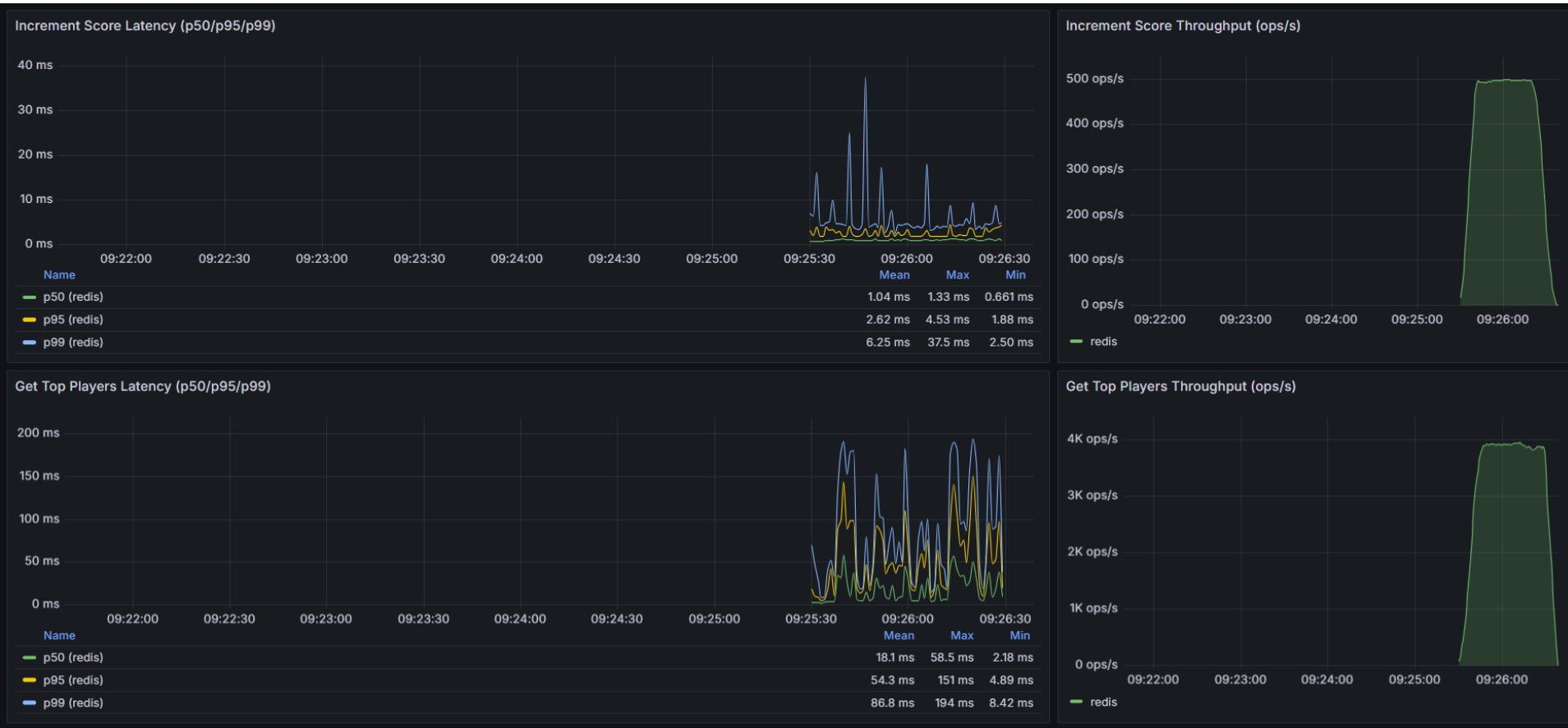


Redis

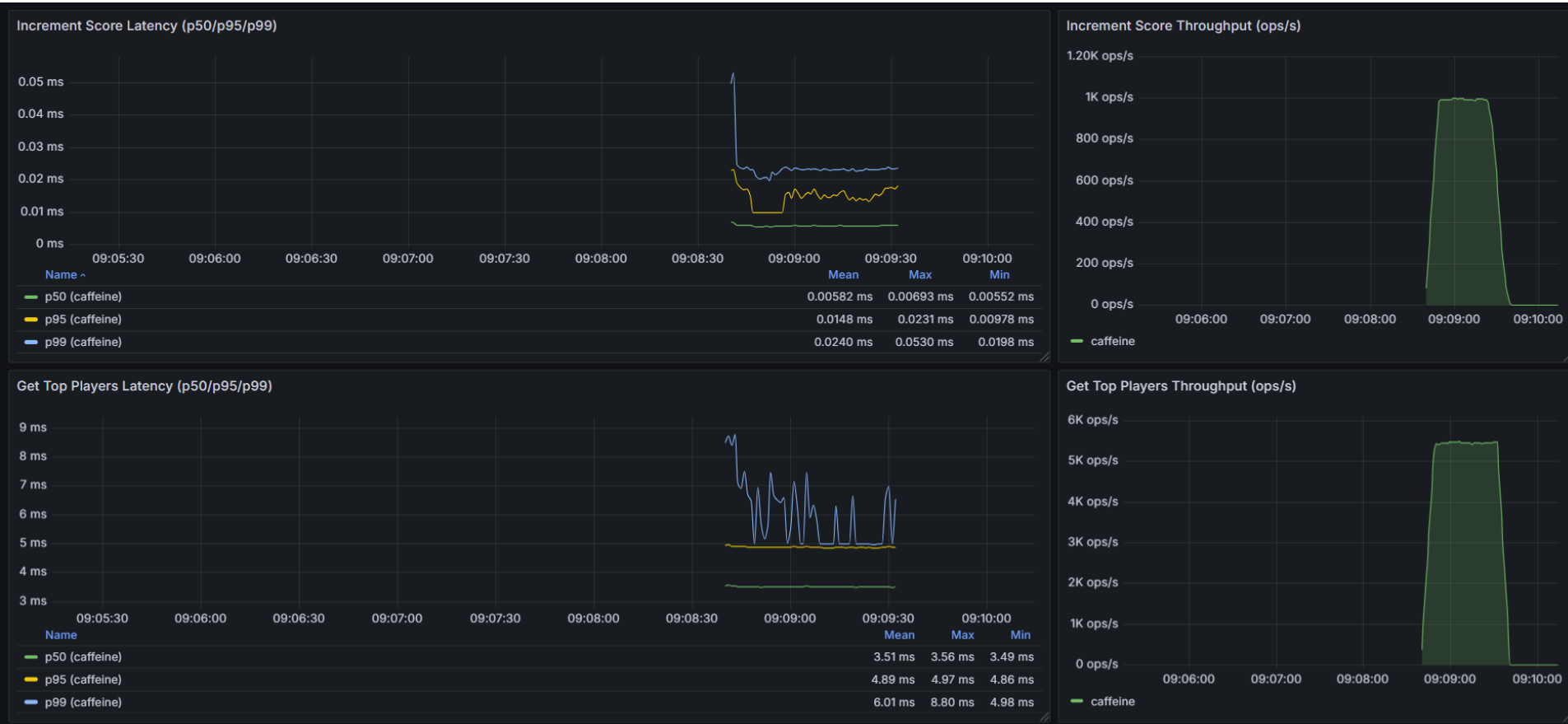


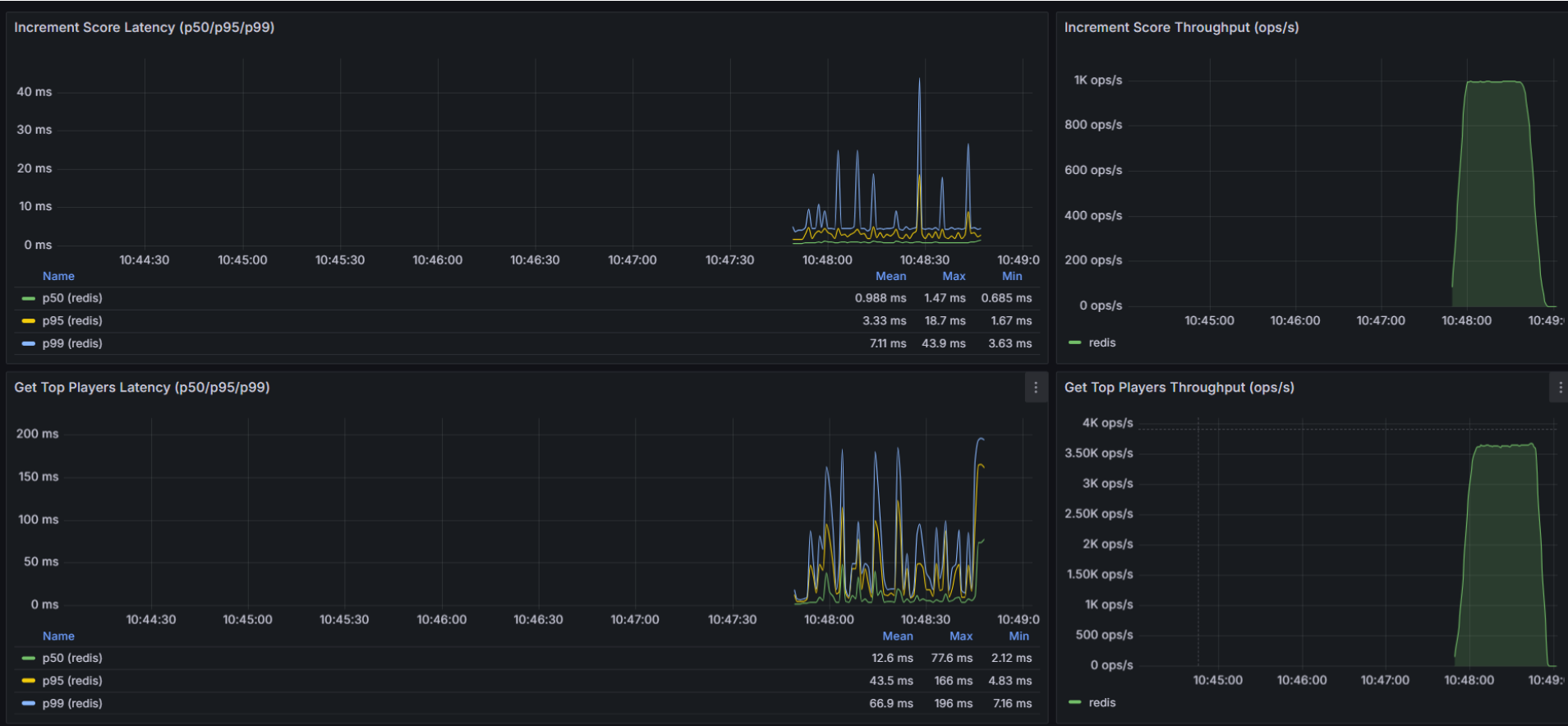
0.5k ops/s Yazma Kayıpsız işlenirken Maksimum Okuma : **JAVA**

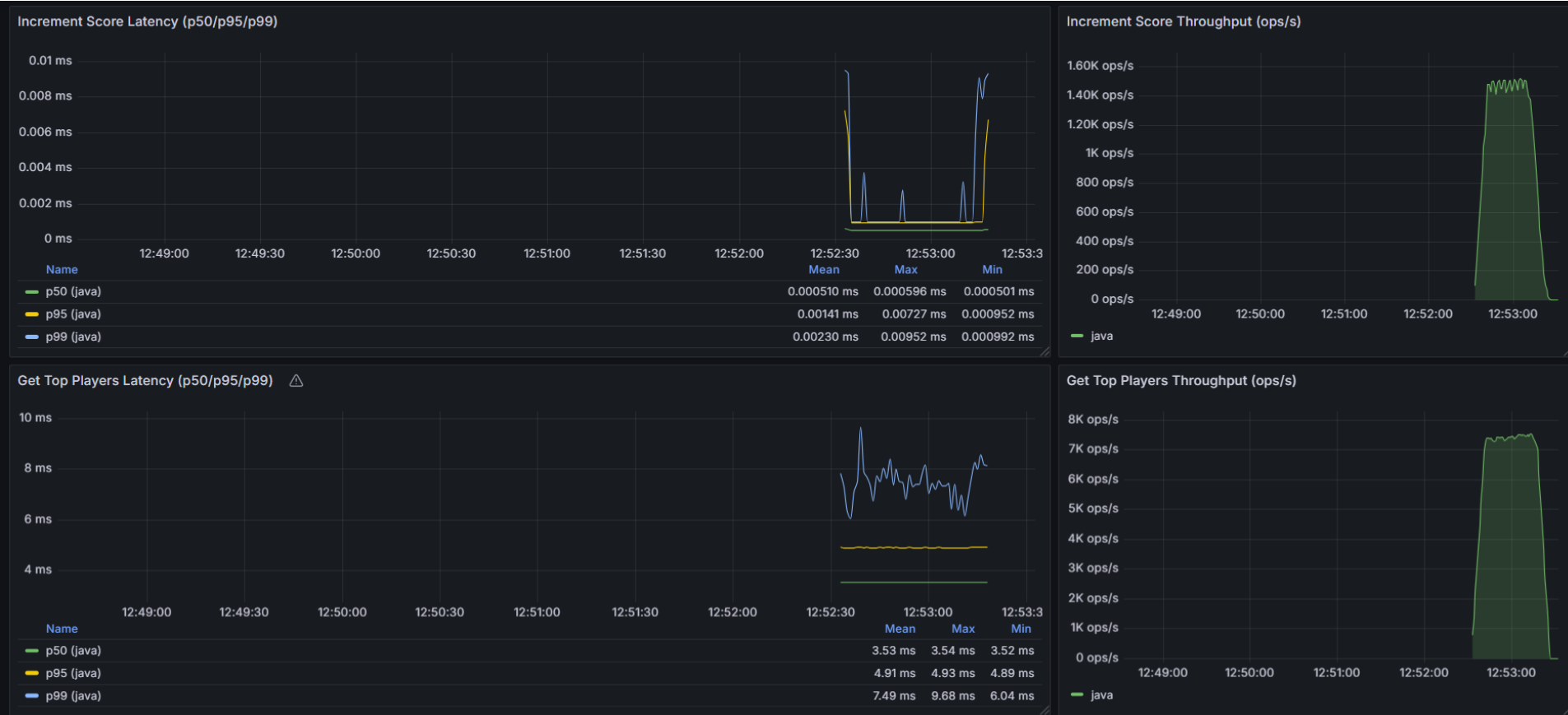
0.5k ops/s Yazma Kayıpsız işlenirken Maksimum Okuma : **CAFFEINE**

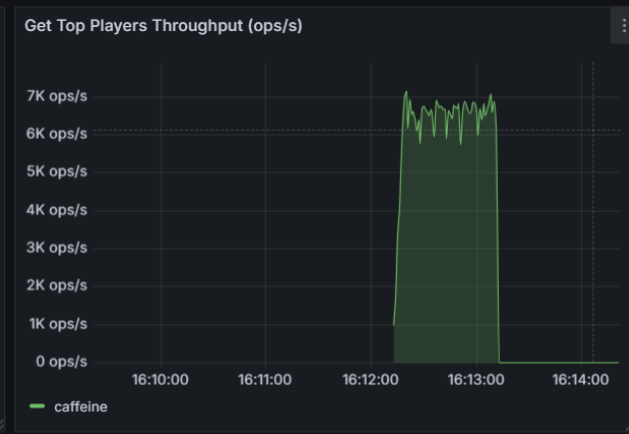
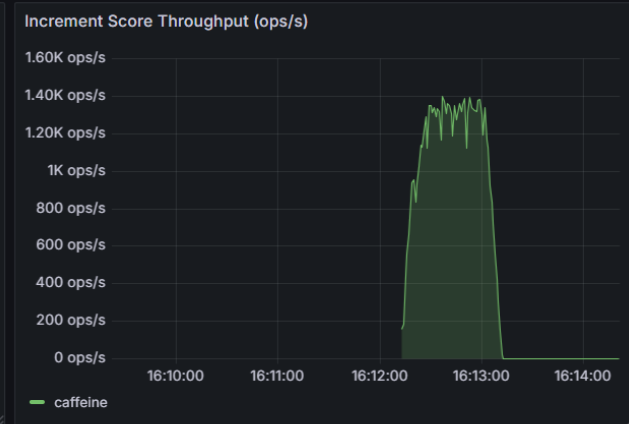
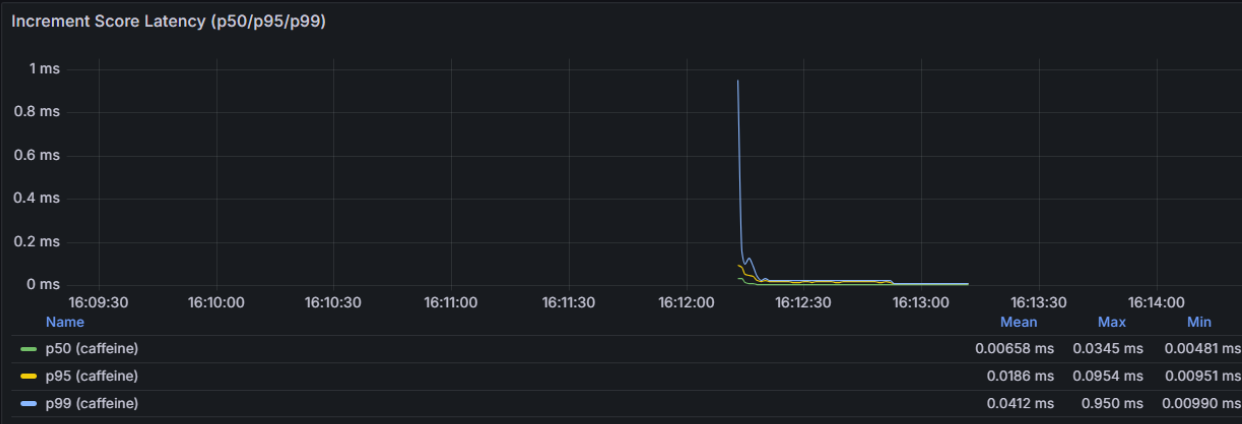
0.5k ops/s Yazma Kayıpsız işlenirken Maksimum Okuma : **REDIS**

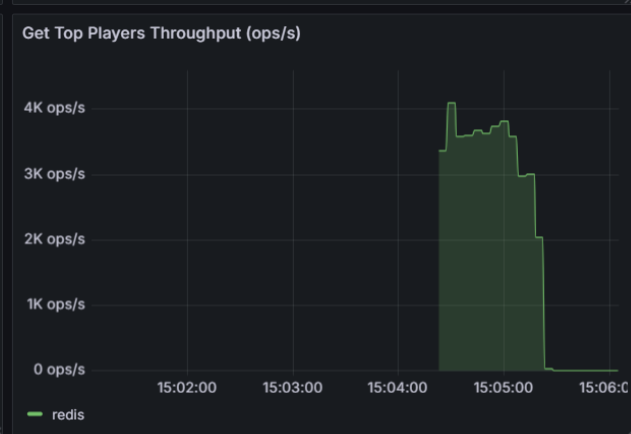
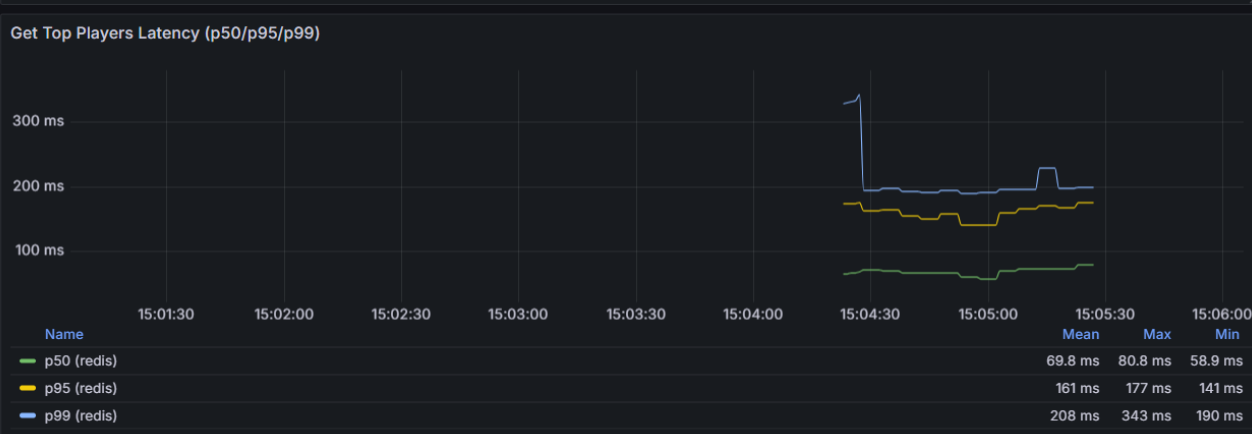
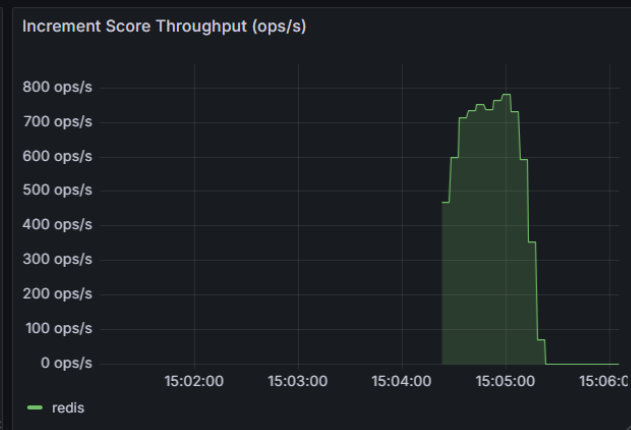
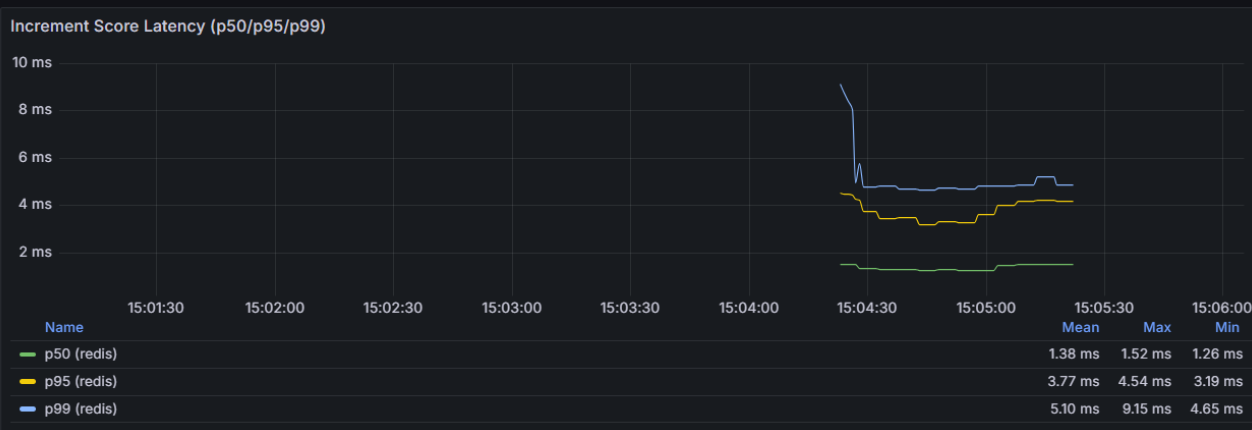
1k ops/s Yazma Kayıpsız işlenirken Maksimum Okuma : **JAVA**

1k ops/s Yazma Kayıpsız işlenirken Maksimum Okuma : **CAFFEINE**

1k ops/s Yazma Kayıpsız işlenirken Maksimum Okuma : **REDIS**

2k ops/s Yazma ve 10k ops/s Okuma isteğinin verimi : **JAVA**

2k ops/s Yazma ve 10k ops/s Okuma isteğinin verimi : **CAFFEINE**

2k ops/s Yazma ve 10k ops/s Okuma isteğinin verimi: **REDIS**

Yorum

- **Java**, yazma ve okumada **en yüksek** verim (throughput) ve en düşük gecikmeleri sunuyor.
- **Caffeine**, Java'ya kıyasla **~%10–20 daha düşük** kapasite ama hâlâ mikro-saniye düzeyinde gecikme.
- **Redis**, dağıtık **persistan (persistence) mimarinin** bedeli olarak milisaniyeler düzeyinde gecikme ve **daha düşük verim'e** (throughput) sahip

•Plain Java

- ✓ En yüksek performans
- ✗ Bellek içi, restart ile tüm veri kaybolur
- ✗ Dağıtık kullanım desteği yok

•Caffeine

- ✓ Otomatik TTL ve maksimum boyut kontrolü
- ✓ Yine bellek içi, ancak cache miss durumunda yavaşlayabilir
- ✗ Dağıtık kullanım desteği yok

•Redis

- ✓ Kalıcı, dağıtık mimari desteği
- ✓ Sıralama desteği (Sorted Set)
- ✗ En düşük verim (throughput) ve en yüksek gecikme

- **RESTful API Tasarımı**
- **Bellek İçi Çözümler**
- **Performans İzleme**
- **Yük Testi Tecrübesi**
- **Modern Tech-Stack Tecrübesi**
- **Sunum ve Raporlama**

Soru - Cevap