

Java uygulamalarının performansını optimize etmek için aşağıdaki stratejileri uygulardım:

JVM Ayarları ve Bellek Optimizasyonu

- Heap Boyutunu Ayarlarım: -Xms ve -Xmx parametrelerini ayarlayarak garbage collection verimliliğini artırırım.
- Garbage Collector Seçerim: Uygulamaya uygun GC algoritmasını (G1GC, ZGC, Shenandoah, vb.) seçerek daha düşük gecikme veya daha az bellek tüketimi sağlarım.

Veritabanı Optimizasyonu

- Doğru İndeksleri Kullanırım: Sorguların hızlanması için gerekli sütunlara indeks eklerim.
- Lazy vs. Eager Loading Seçimi Yaparım: Gereksiz veri çekmeyi önlemek için Lazy Loading kullanırım.
- Connection Pooling Kullanırım: HikariCP gibi kütüphanelerle veritabanı bağlantılarını optimize ederim.

Kod Optimizasyonu ve İnceleme

- Profiling Araçları Kullanırım: JVisualVM, YourKit, Flight Recorder gibi araçlarla darboğazları tespit ederim.
- Efektif Veri Yapıları Seçerim: ArrayList yerine LinkedList, HashMap yerine TreeMap gibi uygun veri yapıları kullanırım.
- Multithreading Kullanırım: ForkJoinPool, CompletableFuture gibi yapılarla paralel işlem yaparak performansı artırırım.

I/O ve Network Optimizasyonu

- Buffering Kullanırım: BufferedReader, BufferedWriter gibi araçlarla I/O işlemlerini hızlandırırım.
- Asenkron İşlemler Kullanırım: CompletableFuture, Reactive Streams ile IO gecikmelerini azaltırım.