Spring Framework 6

Beginner to Guru

Introduction to Caching

Introduction to Caching

- Caching is the process of storing data in a temporary location for improved performance
- Examples:
  - **Web Browsers** - Will cache web site resources such as images and CSS style sheets locally so they can be reused and not downloaded again
  - **DNS** - Once an address is looked up, it is cached locally for efficiency
  - **Databases** - When data is read, it is kept in memory for a period of time for faster access
  - **Web Servers** - When data is read, it is kept in memory for a period of time for faster access

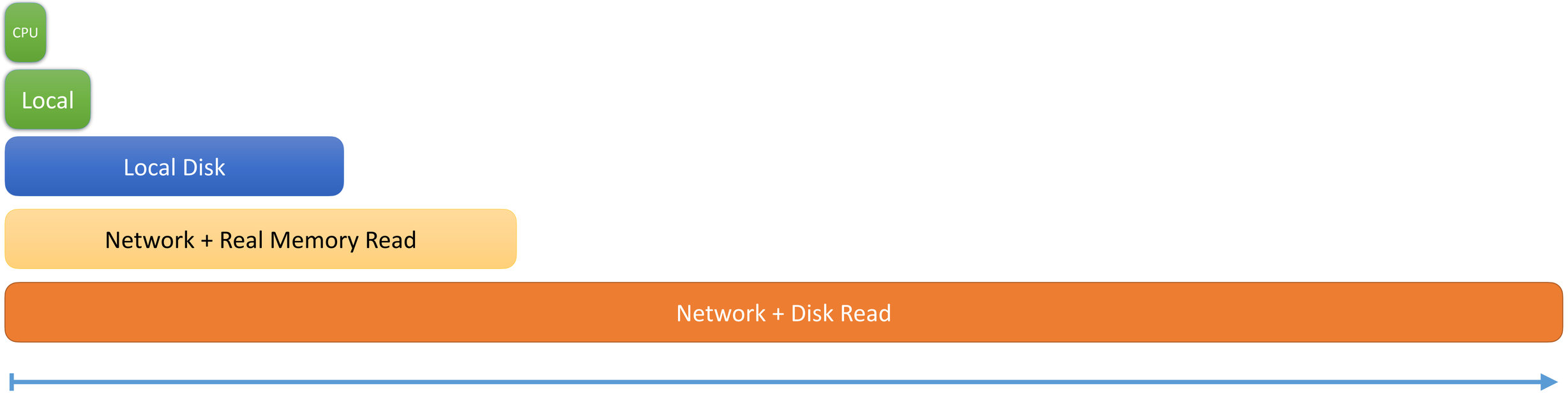
Caching Pros and Cons

- **Pros:**
  - Faster performance
  - Improved Scalability
- **Cons:**
  - Risk of stale data
  - Impact of stale data will vary based on application

When to Use Caching?

- **When to Use Caching?**
  - Caching is best for Read operations
  - Ideal for data that does not change a lot
  - Can make a significant reduction in workload for heavy read operations
- **When Not to Use Caching?**
  - Mostly Transactional Systems
  - When data is frequently updated
  - When data gets updated by external systems

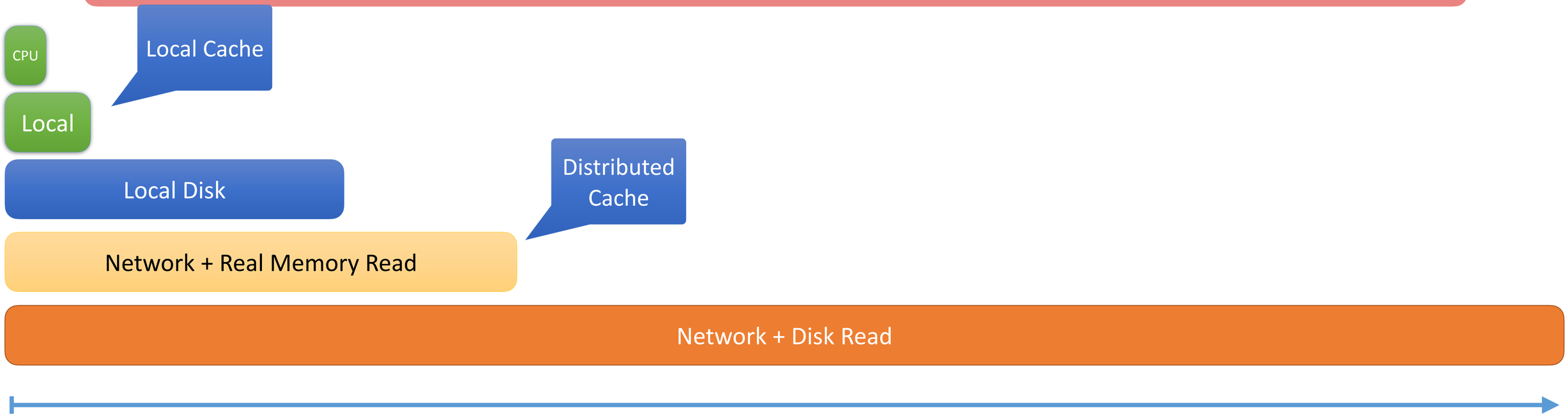
I/O Performance 101



Types of Caches

- **Local**
  - Local to the JVM or machine node, only used for traffic on that machine
  - Fastest
- **Distributed Cache**
  - One or more machines networked together using pooled real memory
  - Still fast and efficient
  - Cache is shared between many cache consumers

Cache Performance



Caching with Spring

- Spring Framework has robust support for Caching
  - **Simple** - Java ConcurrentHashMap
  - **JRS-107 Java Caching API** - Ehcache 2, Hazelcast, Ininispan
  - **In Memory Databases** - Couchbase, Redis
  - **None** - No-op cache

SPRING FRAMEWORK

GURU