Recitation 8 — The Tail at Scale

Relation to Lecture

- Previous lecture looked at performance-improving techniques related to a single machine. This recitation looks at performance-improving techniques in a large distributed system with a "fan-out" architecture, and specifically focuses on latency.

Key concepts

- **Tail latency:** The slowest responses in a distribution of request latencies. Often measured at high percentiles (e.g., 99th or 99.9th percentile).
- **Fan-out architecture:** Many internet services process user requests by querying dozens or hundreds of servers in parallel (e.g., a single Google search query may contact hundreds of servers).
- **Impact on user experience:** The user sees the slowest server response because the final result can only be assembled after all required sub-requests have completed.

Sources of long tail latencies

- **Queueing delays and overload:** Requests pile up during high load, increasing response times.
- **Resource interference and contention:** Shared resources among services (e.g., CPU, cache, network bandwidth).
- **Straggler tasks:** The presence of outlier tasks that run significantly slower due to hardware hiccups, garbage collection pauses, or data skew.
- **Network variability:** Packet loss, congestion, or retry overhead in a distributed system.

Mitigation Techniques

Lots of options, including

- **Hedged requests:** Send requests to multiple servers or replicas in parallel, use the result from whichever completes first, and cancel the others.
- **Selective/adaptive replication:** Instead of replicating every request, replicate only requests that are "lagging"
- Break big tasks into smaller subtasks
- If parts of the result are non-critical, return partial results faster while slower tasks complete.

Trade-offs

- Cost vs. performance
- Complexity vs. performance
- Discuss: When is the extra cost of replication or hedged requests justified?