**BACKGROUND**

A Cloud Scenario
- Cloud computing allows sharing of resource at the cost of reduced isolation.
- Storage systems are highly sensitive to performance interference.
- Lack of perform isolation will lead to unpredictable latencies.
- In worst case, a particular HTTP request may require 35 database lookups. [2]
- Amazon reported 100ms of latency cost them 1% in sales. [3]
- Google found an extra .5 seconds delay caused 20% drop in search traffic. [3]

Performance Isolation
- One can give clients dedicated resources, however this eliminates cost reduction.
- MicroFuge aims to meet clients’ response time requirement in the shared environment.
- Represent response time requirements with request deadlines.
- Provide performance isolation by meeting request deadlines.

**REFERENCES**


**MicroFuge**

A distributed caching and scheduling middleware that provides performance isolation.

**Deadline Cache (DLC)**
- Builds a performance model.
- Uses multiple LRU queues for deadline-aware evictions.
- Performs intelligent replica selection.
- Implements feedback-driven deadline-aware scheduling.
- Optionally performs admission control.

**Deadline Scheduler (DLS)**
- Performs intelligent replica selection.
- Implements feedback-driven deadline-aware scheduling.
- Optionally performs admission control.

**Deadline Scheduling**

With MicroFuge’s distributed design, each DLS is responsible for scheduling client access to a subset of distributed data servers. Each scheduler performs three tasks to provide performance isolation in the scheduling layer.

- DLS will select the replica which is most likely to meet request’s deadline.
- In order to make the selection, DLS relies on the latency modeling component which uses previous request latencies to predict incoming request’s latency.
- There are cases where server load just exceeds its capacity. DLS additionally provides an optional admission control mechanism which performs early rejection of requests which are likely to miss their deadlines.

**Performance Evaluation**

**Experimental Setup**
- Twenty-node test cluster on AWS. Each cluster node is an m1.medium EC2 instance.
- Benchmarking System: Yahoo! Cloud Serving Benchmark (YCSB). Modified to assign different ranges of deadlines to each key.

**Conclusion**

- Predictable performance is necessary in multi-tenant environments.
- MicroFuge tackles the performance isolation problem with its deadline-aware caching and scheduling middleware.
- MicroFuge reduces deadline miss rate from 17.5% to 7.7% and it can be as low as 4.7% if we turn on the admission control.