Sparrow

Fan Jun Bo

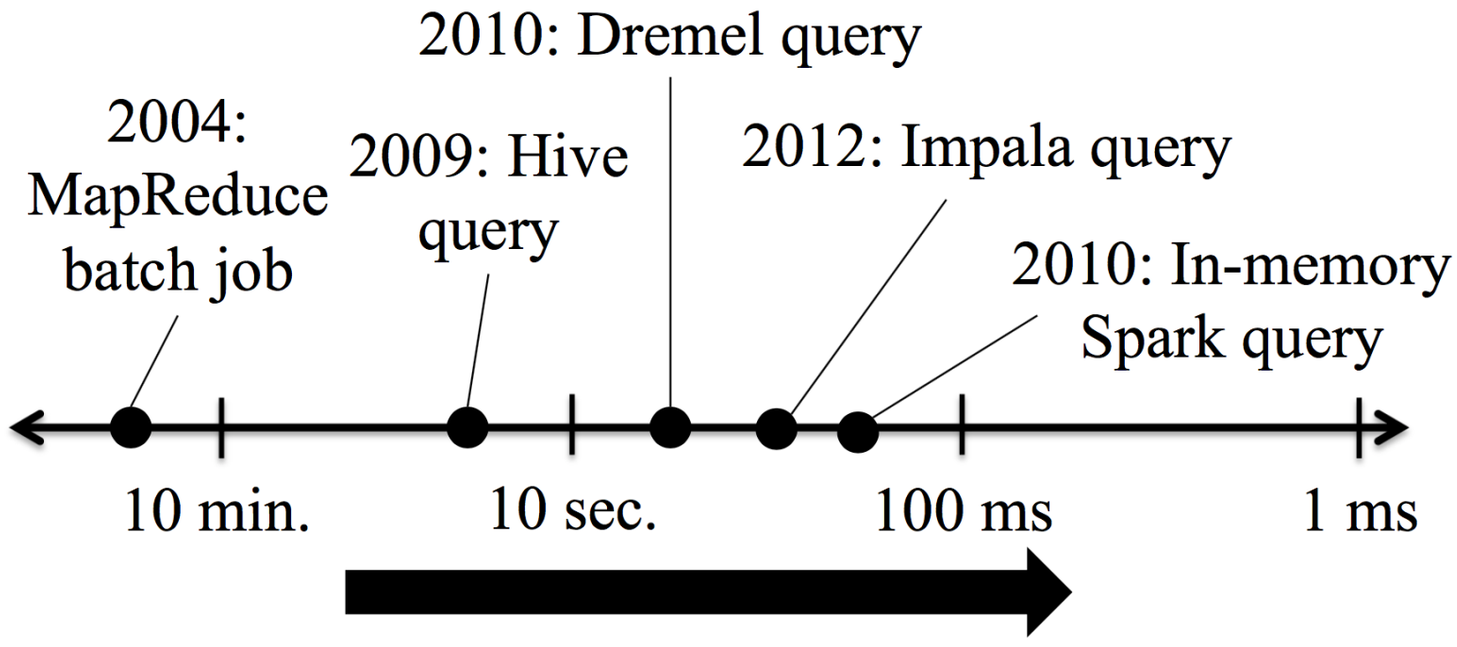


Figure 1: Data analytics frameworks can analyze large volumes of data with ever lower latency.

breaking long-running batch jobs into a large number of short tasks

Low Latency

Time

wait time: job submitted -> begin to execute

service time: begin to execute -> job done

response time: job submitted -> last task done

delay: scheduler time + queue time

Sparrow

decentralized, randomized sampling approach provides near-optimal per- formance while avoiding the throughput and availability limitations of a centralized design.

Random Sampling

Per-task Sampling

Batch Sampling

Late-Binding

Constraints(per job vs per task)

Fault tolerance

Schedular failure

worker failure and cluster fail

n	Number of servers in the cluster
ρ	Load (fraction non-idle workers)
m	Tasks per job
d	Probes per task
t	Mean task service time
$\rho n/(mt)$	Mean request arrival rate

Table 1: Summary of notation.

Random Placement	$(1-\rho)^m$
Per-Task Sampling	$(1-\rho^d)^m$
Batch Sampling	$\sum_{i=m}^{d \cdot m}(1-\rho)^i \rho^{d \cdot m - i} \binom{d \cdot m}{i}$

Table 2: Probability that a job will experience zero wait time under three different scheduling techniques.

Figure 4: Probability that a job will experience zero wait time in a single-core environment using random placement, sampling 2 servers/task, and sampling $2m$ machines to place an m-task job.

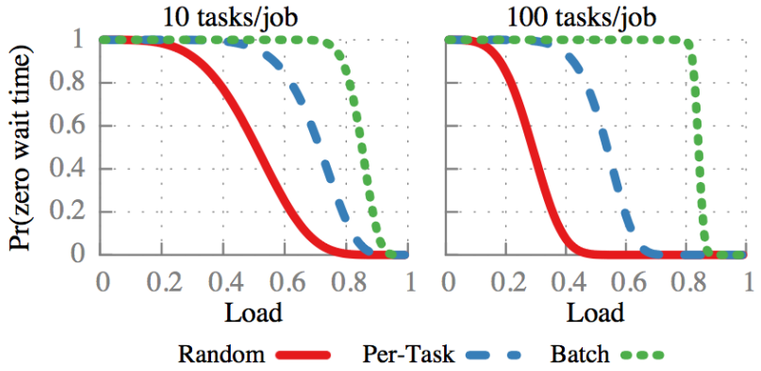
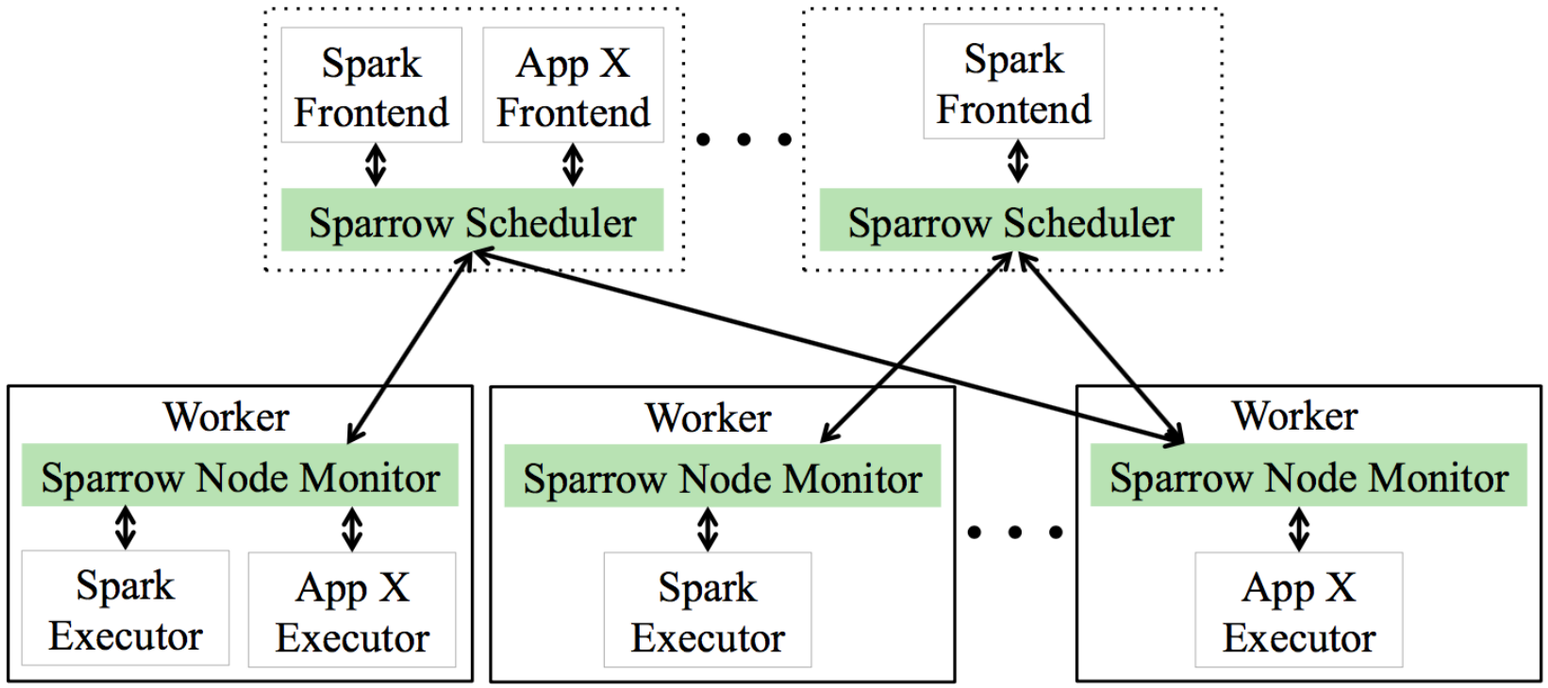


Figure 5: Probability that a job will experience zero wait time in a system of 4-core servers.

Pr(zero wait time) in theoretical condition for single and multicore.



Sparrow structure in real

The Experiments

100 worker machines<8cores, 68.4GB RAM> with 10 schedulers
probe ratio = 2

Performance on TCP-H Workload

10 users launch random permutation of TCP-H queries to make the overload 80% for a period of 15 minutes. During the middle 200 seconds, Sparrow scheduler handles 20K jobs that make up 6.2K TCP-H queries.

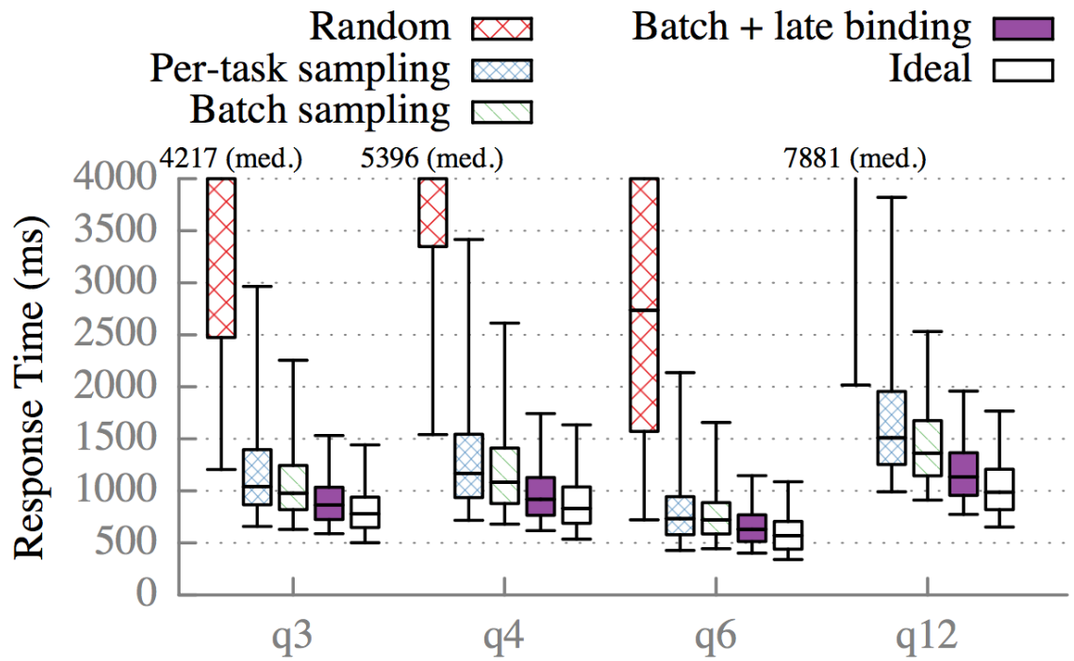


Figure 8: Response times for TPC-H queries using different placement stategies. Whiskers depict 5th and 95th percentiles; boxes depict median, 25th, and 75th percentiles.

Response time for different types of schedulers

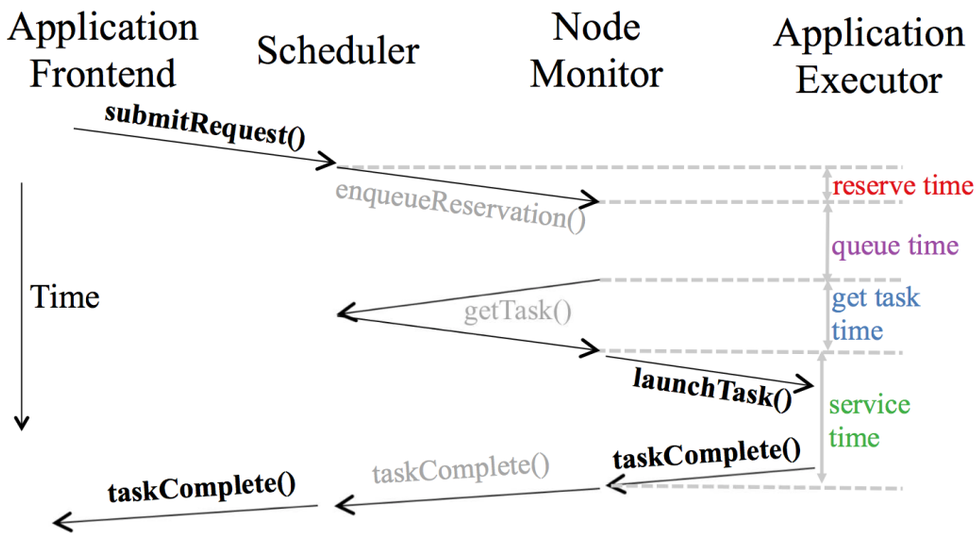


Figure 7: RPCs (parameters not shown) and timings associated with launching a job. Sparrow's external interface is shown in bold text and internal RPCs are shown in grey text.

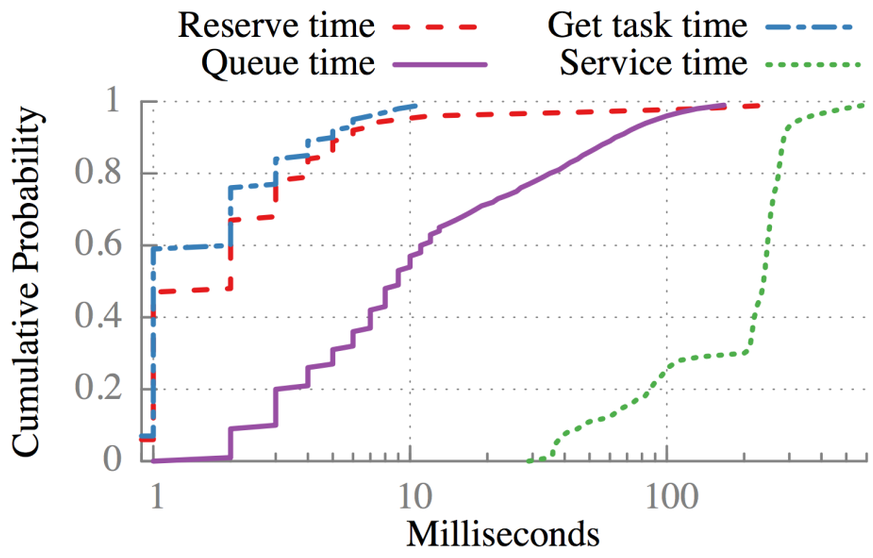


Figure 9: Latency distribution for each phase in the Sparrow scheduling algorithm.

Latency distribution among different stages

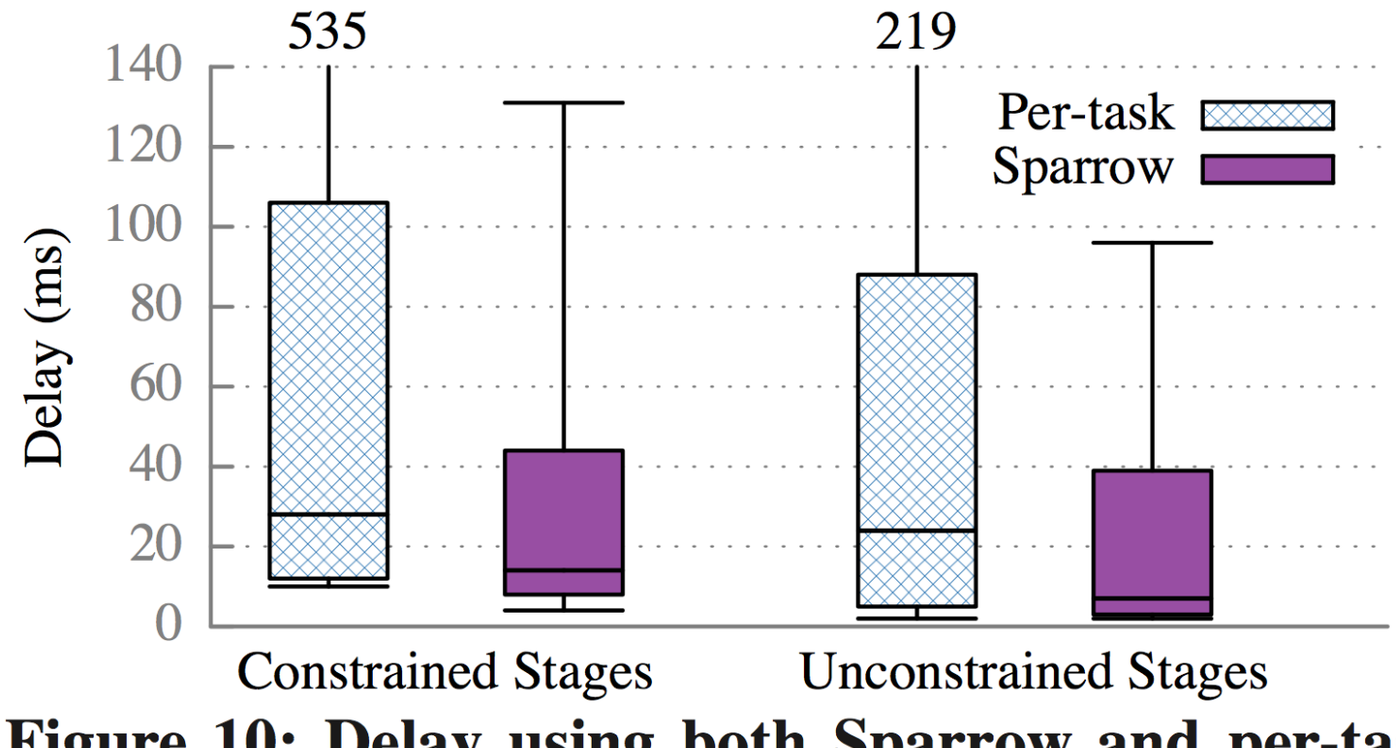
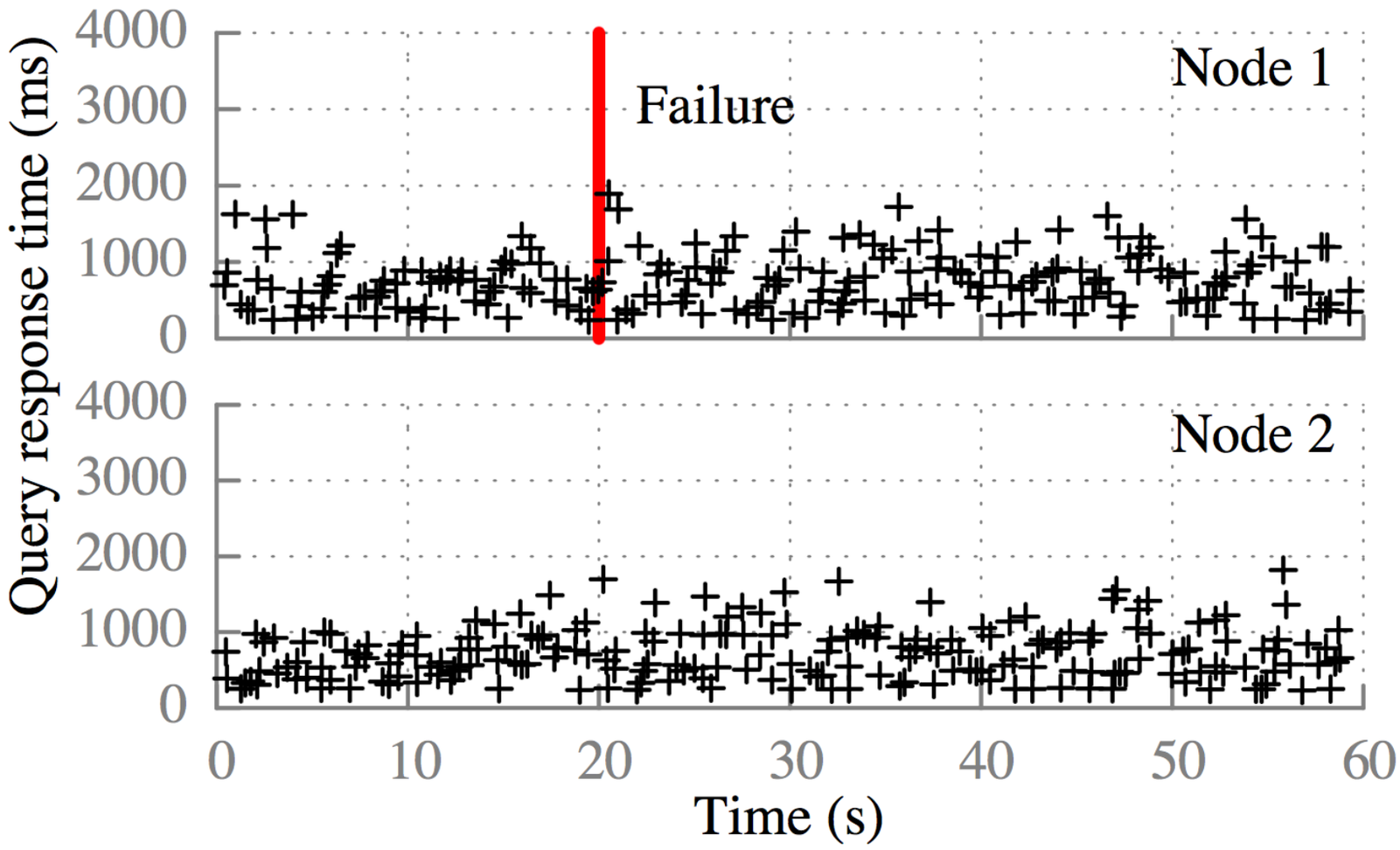


Figure 10: Delay using both Sparrow and per-task sampling, for both constrained and unconstrained Spark stages. Whiskers depict 5th and 95 percentiles; boxes depict median, 25th, and 75th percentiles.

Delay with and without constraints



Failure for scheduler in node 1 at 20s
100ms failure detection
5ms to reconnect scheduler in node 2
15 ms to relaunch jobs

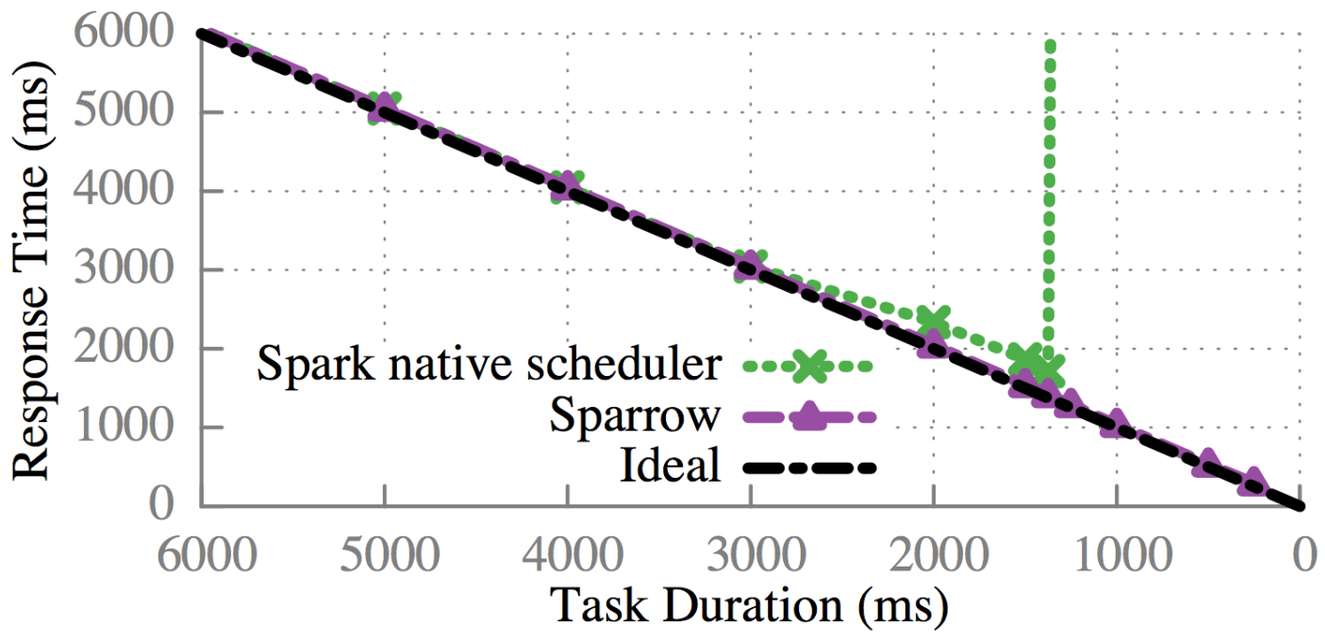


Figure 12: Response time when scheduling 10-task jobs in a 100 node cluster using both Sparrow and Spark's native scheduler. Utilization is fixed at 80%, while task duration decreases.

Sparrow vs Spark's native scheduler. For task duration less than 1380ms, Spark's native scheduler suffers performance issue

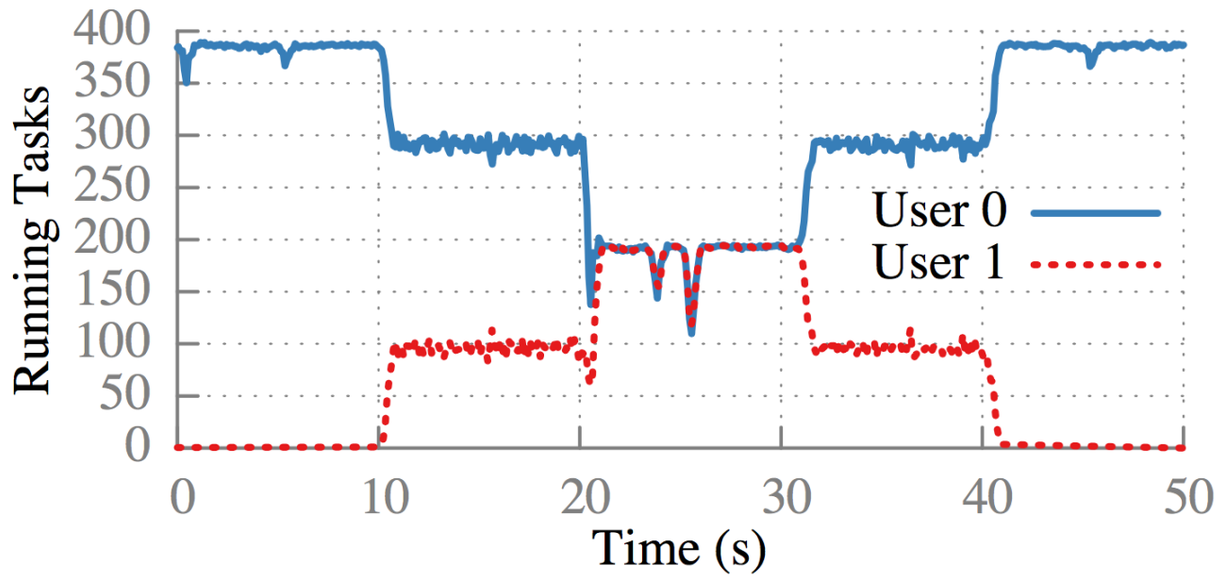


Figure 13: Cluster share used by two users that are each assigned equal shares of the cluster. User 0 submits at a rate to utilize the entire cluster for the entire experiment while user 1 adjusts its submission rate each 10 seconds. Sparrow assigns both users their max-min fair share.

Fairness sharing between two users.

HP load	LP load	HP response time in ms	LP response time in ms
0.25	0	106 (111)	N/A
0.25	0.25	108 (114)	108 (115)
0.25	0.5	110 (148)	110 (449)
0.25	0.75	136 (170)	40.2k (46.2k)
0.25	1.75	141 (226)	255k (270k)

Table 3: Median and 95th percentile (shown in parentheses) response times for a high priority (HP) and low priority (LP) user running jobs composed of 10 100ms tasks in a 100-node cluster. Sparrow successfully shields the high priority user from a low priority user. When aggregate load is 1 or more, response time will grow to be unbounded for at least one user.

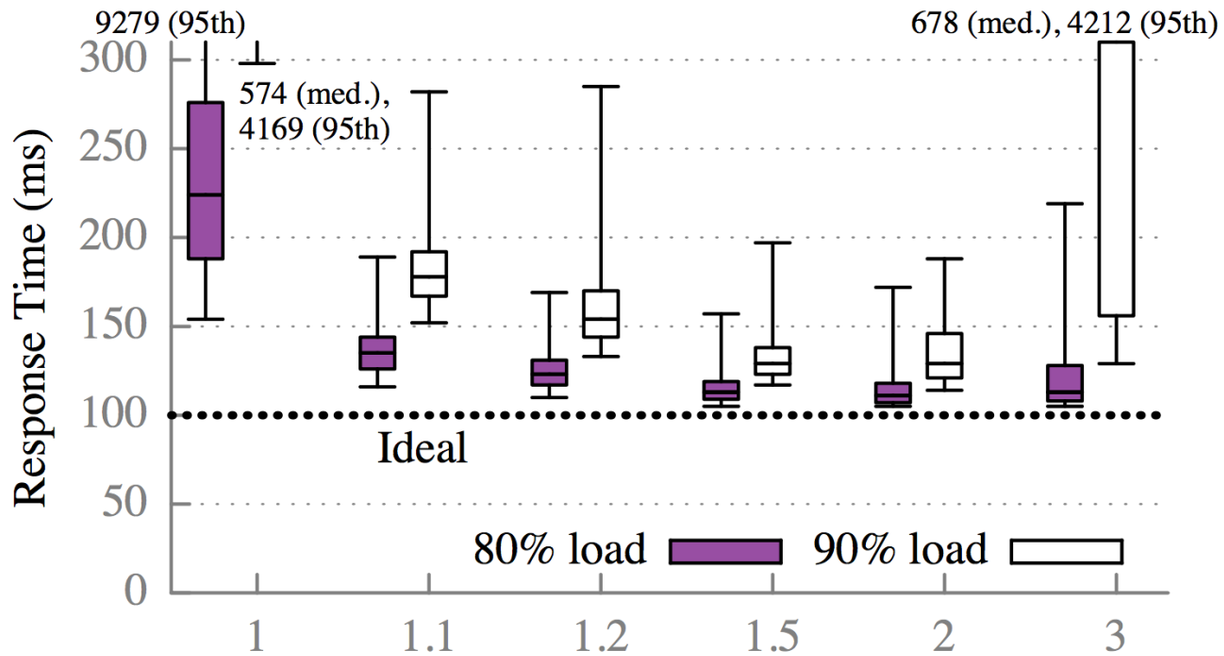


Figure 14: Effect of probe ratio on job response time at two different cluster loads. Whiskers depict 5th and 95th percentiles; boxes depict median, 25th, and 75th percentiles.

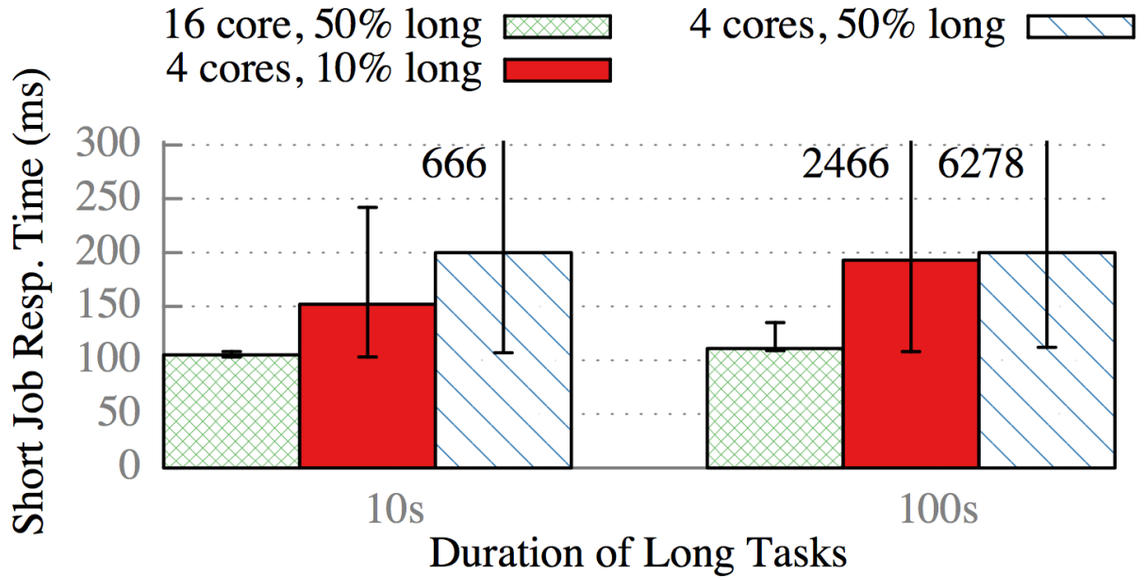


Figure 15: Sparrow provides low median response time for jobs composed of 10 100ms tasks, even when those tasks are run alongside much longer jobs. Error bars depict 5th and 95th percentiles.