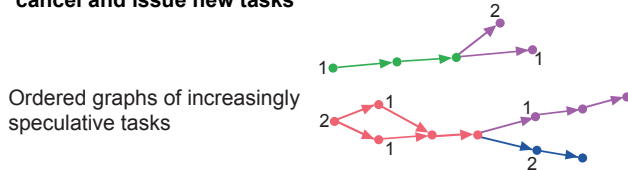


Cluster Scheduling for Explicitly-Speculative Tasks

David Petrou, Greg Ganger, Garth Gibson

Environment

- Computing cluster from 10s to 100s of users and nodes
- Users submit chains of tasks, some speculative
- May not need speculative tasks (analogy: prefetches)
- After analyzing task output for some *think time*, user may cancel and issue new tasks

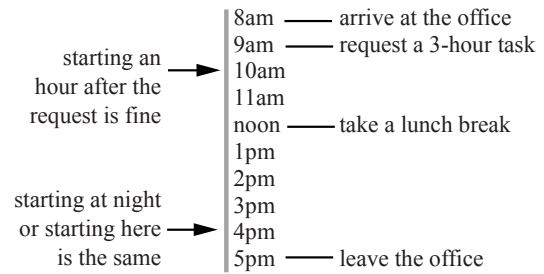


Ordered graphs of increasingly speculative tasks

Applications	Service Time	Service Time Factors
find DNA sequences (Scientists study BLAST results.)	20-40 mins	source X target DNA size
render movie scenes (Artists create computer-animated scenes.)	hours	scene length and complexity
prefetch web docs (Prefetching agents show network applicability.)	<1 min	document size & network conditions

Problems

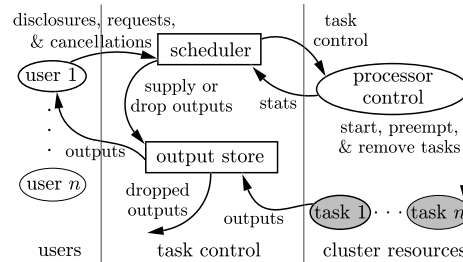
- Submitting all tasks at once (*batch*) wastes resources on unneeded speculative tasks
- Submitting tasks only when needed (*interactive*) precludes pipelining execution and think time
- Response time ignores when users cannot use task output:



Away periods could provide scheduling flexibility

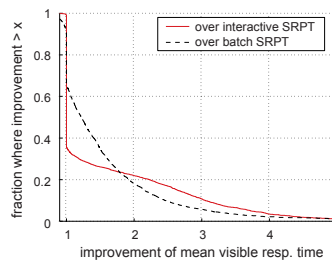
Batchactive Solutions

- Users *disclose* speculative tasks and *request* needed tasks
- Requested tasks run first and disclosed tasks run only when idle
- Scheduler aims to minimize *visible* response time
- Users are charged only for needed tasks

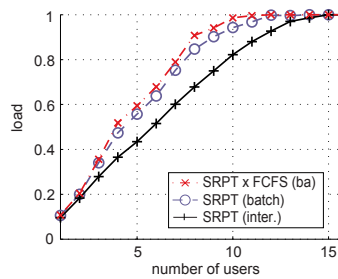
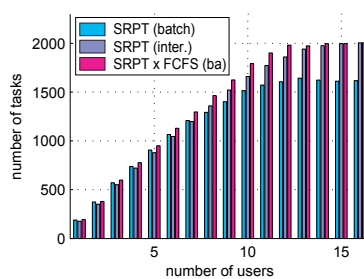
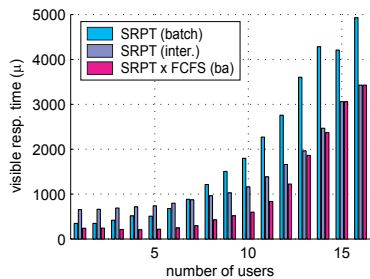


Results

- Synthetic simulation of one processor
- Batchactive achieves better response time than submitting all tasks at once (batch) or only when needed (interactive)



number of users	4-16
task set change prob.	0.0 to 0.0-0.4 (uni.)
tasks in a task set	1 to 1-19 (uni.)
service time (s)	20-3,620 (exp.)
think time (s)	20-18,020 (exp.)



task set change prob.	0.0 to 0.2 (uni.)
tasks in a task set	1 to 15 (uni.)
service time (s)	600 (exp.)
think time (s)	6000 (exp.)