

# Utilization, Predictability, Workloads, and User Run Time Estimates in Scheduling the SP2 With Backfilling

Presented By: Saverio Scumaci and Tim Lorenz

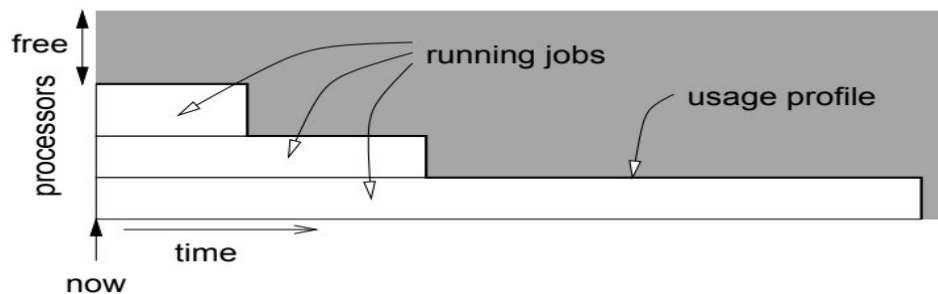


# Contributions

- Comparison of backfilling techniques
- Investigation into the impact of inaccurate user-provided runtime estimates for jobs

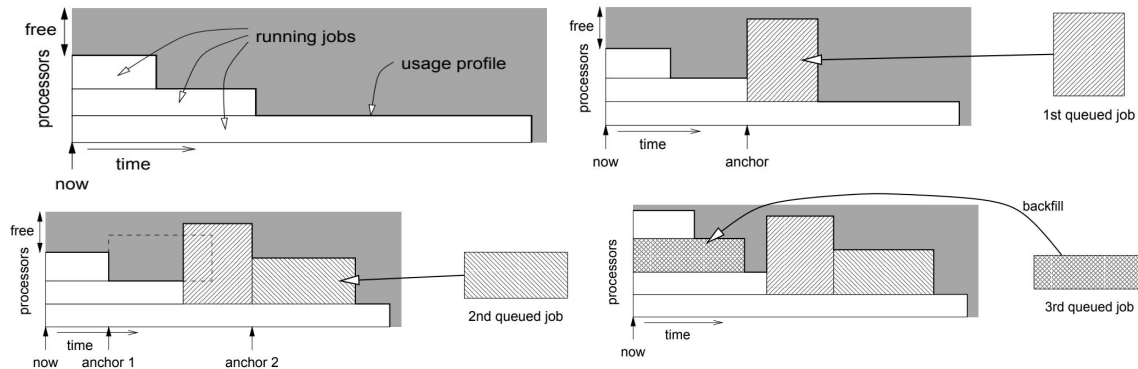
# Backfilling

- Jobs are rectangular
- Users define the number of processors required for each job and a runtime estimate
- “Holes” in the schedule allow for smaller jobs to be placed in is the basis of backfilling



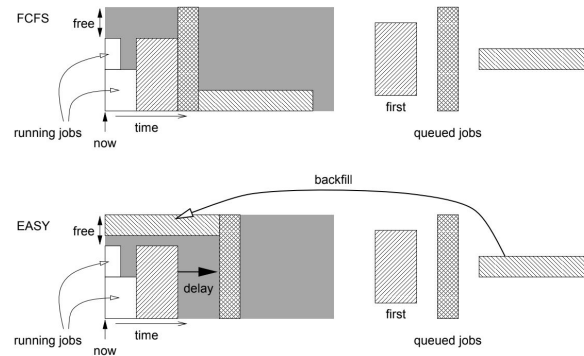
# Conservative Backfilling

- Checks that adding a smaller job does not delay any previous job in the queue
- Allows scheduling decisions to be made upon job submittal
- Can predict when each job will run and give guarantees



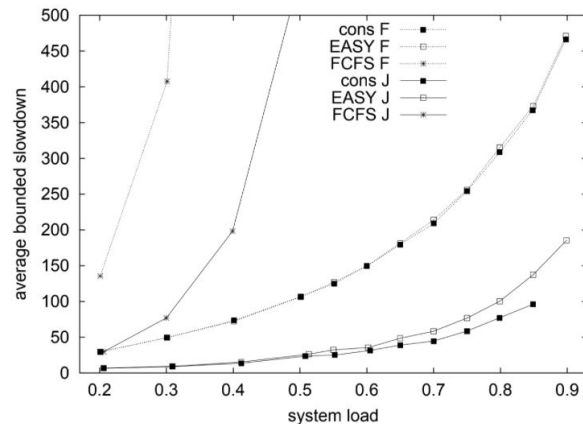
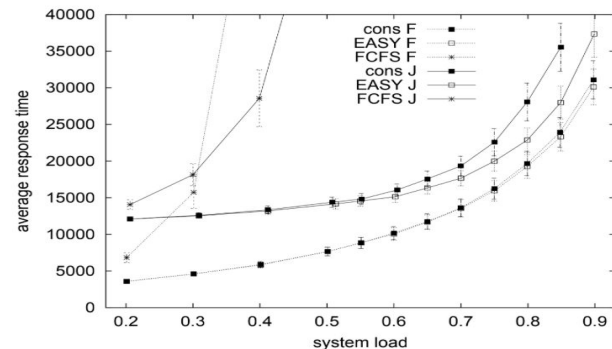
# Easy Backfilling

- Similar to Conservative Backfilling
- Short jobs are added as long as they don't delay the job at the head of the queue



# Results

- Similar results for Feitelson (workload model)
- Jann Model: Easy had better response time but slightly worse bounded slowdown (workload model)
- SP2 and non-SP2 preferred Easy (supercomputer types)
- Par was inconclusive (supercomputer)
- CM-5 preferred conservative (supercomputer)



# User Estimates

- Backfilling cannot work without estimated running times for each job
- If jobs exceed their estimated runtime by a set margin, they are terminated
- Doubling user estimates for job runtime was shown to reduce the number of jobs killed

