High Performance Computing Systems (CMSC714)

Lecture 16: Job Scheduling

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Announcements

- Assignment 1 grades are posted
- Schedule for project presentations online
- Most readings have been posted

Summary of last lecture

- OS daemons can lead to noise or jitter, which leads to performance variability
- Variability leads to practical issues and impacts software optimization cycle
- Can be mitigated by pinning processes and threads and leaving some cores free
- Can significantly impact performance of bulk synchronous programs
- Communication variability comes from other jobs sharing the same network

Job (batch) scheduling

Job (batch) scheduling

- HPC systems use job or batch scheduling
- Each user submits their parallel programs for execution to a “job” scheduler

Job Queue

		#Nodes Requested	Time Requested
1	🟦	128	30 mins
2	🟧	64	24 hours
3	🟩	56	6 hours
4	🟨	192	12 hours
5	🟦	…	…
6	🟨	…	…

Job (batch) scheduling

- HPC systems use job or batch scheduling
- Each user submits their parallel programs for execution to a “job” scheduler
- The scheduler decides:
  - what job to schedule next (based on an algorithm: FCFS, priority-based, ….)
  - what resources (compute nodes) to allocate to the ready job

Job Queue

		#Nodes Requested	Time Requested
1	🟦	128	30 mins
2	🟧	64	24 hours
3	🟩	56	6 hours
4	🟨	192	12 hours
5	🟦	…	…
6	🟨	…	…

Job (batch) scheduling

- HPC systems use job or batch scheduling
- Each user submits their parallel programs for execution to a “job” scheduler
- The scheduler decides:
  - what job to schedule next (based on an algorithm: FCFS, priority-based, ….)
  - what resources (compute nodes) to allocate to the ready job

- Compute nodes: dedicated to each job
- Network, filesystem: shared by all jobs

Job Queue

		#Nodes Requested	Time Requested
1	🟦	128	30 mins
2	🟧	64	24 hours
3	🟩	56	6 hours
4	🟨	192	12 hours
5	🟦	…	…
6	🟨	…	…

Two components of a scheduler

- Decide what job(s) to schedule next: scheduler
- Decide what nodes (and other resources) to allocate to them: resource manager

Scheduling policies

- First come first serve (FCFS)
- Priority-based
  - Depending on project name and remaining allocation
- Backfilling
  - Use idle nodes that are being reserved for the next large jobs
  - Aggressive (EAZY) backfill: run jobs as long as they don't delay the first job (could lead to unbounded delays)
  - Conservative backfill: runs jobs as long as they don't delay **any** future job

Resource management

- Most primitive: manage nodes
- Advanced management:
  - Node type aware (low vs. high memory, GPU nodes)
  - Network topology aware
  - Power aware

Quality of service metrics

- Job Wait Time: time between a job's submission and start

$$T_{\text{wait}} = T_{\text{start}} - T_{\text{submit}}$$

- Slowdown: incorporates running time of a job

$$\text{Slowdown} = \frac{T_{\text{wait}} + T_{\text{running}}}{T_{\text{running}}}$$

Quality of service metrics

- System Utilization: fraction of nodes allocated to running jobs at a given time

$$utilization_t = \frac{N_t}{N}$$

- Schedule Makespan: time between the first job's submission and last job's completion for a job trace (workload)

Questions

Job Scheduling Under the Portable Batch System

- This is a fairly old job scheduling system. What are some of the key features newer systems like slurm provide?
- Are BASL and Tcl still the main scheduling script languages? Or are there more modern alternatives?
- How does the PBS system allow for more variable implementation than previous scheduling systems?
- What are the key shortcomings of PBS?

Questions

A Comparative Study of Job Scheduling Strategies in Large-scale Parallel Comp. Systems

- What is the difference between batch mode and online mode scheduling? When might one method be preferred?
- Can one dynamically choose a scheduling policy based on certain conditions? I.e. type of jobs being submitted, taking into consideration user priority during certain time windows, etc? How much overhead would a complicated system like this have?

Questions?

UNIVERSITY OF MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu