# Distributed Systems & Scheduling

Tai Henrichs

#### **Overview of Distributed & Cluster Systems**



**Example Distributed Resource Manager: Apache Hadoop's YARN** (Yet Another Resource Manager)





- Job: Splits input data into chunks to be processed by map tasks in parallel, joined back together by reduce task(s)
- **Application:** Collection of job(s), potentially with dependencies indicated by a DAG
  - **Container**:
    - Resource-abstraction, contains a subset of the resources of one node, one computer (or VM, etc.)





Two-Part **Resource Manager**:

- Scheduler:
  - Can plug-in different schedulers
  - Only allocates resources
- Application Manager:
  - Accepts + validates jobs
  - Monitors and restarts jobs as needed
- Node Manager: Monitors resource-usage for scheduler, launches containers
- Application Master:
  - Handles the execution of applications' jobs
  - Comm. with nodes to assign + distribute jobs to containers





- Why this design? Performance!
  - Before 2011, Hadoop had a single-node **Job Manager** that handled all resource management
  - Remember: node=machine, so all task-management was done on a single machine
  - This distributes scheduling itself across different machines!

# Distributed Scheduling: Unique Considerations

## Heterogeneity

- Jobs + containers have, respectively, varied bottlenecks and resources
- Example:
  - X is a CPU-constrained job
  - Y is a I/O-constrained job
- Impacts Fairness:
  - Programs X and Y receive differential benefit from 100 CPU-ticks OR IO calls
- Impacts Resource-Utilization:
  - What if we put X on a container with bad CPU and good I/O, vice-versa for Y?
    - It's worse for everyone!
- Solutions:
  - Consider all resources during scheduling
    - Assign mixed-workloads per-node, optimized using heuristics using heuristics (NP-Hard)
    - Hadoop computes resources only uses
      RAM + CPU cores to limit overhead



Figure 2-1. Per-container physical memory usage versus CPU usage during a representative period of time on a production cluster. Note that some jobs consume large amounts of memory while using relatively little CPU; others use significant CPU but relatively little memory.

#### **Dependencies**

- Jobs have dependencies:
  - Input data
    - Intermediate processing results
    - Initial input
  - Auxiliary-helper programs
- Presents multiple challenges in a distributed context!



#### **Dependencies Challenge 1: Locality**

- Data-Locality:
  - Hadoop considers locality of input-data when assigning containers to nodes



#### **Dependencies Challenge 2: Network Usage vs. Delays**



#### **Dependencies Challenge 2: Network Usage vs. Delays**

Problem: R may idle for input data if tR <<< t4!

- E.g. Dependency-2's node fails, requires rescheduling on a busy system
- R may be delayed for a while!



#### Handling High Demand

- Each user may have a certain resource-share
- Example:
  - A needs more than share, B is under-utilizing
  - Allocate B's share to A!
- What about fairness? If B eventually needs resources, preempt (kill) A's excess containers
  - Frequent preemption wastes resources



## Handling High Demand

- What if there are too many large jobs?
- Encourage efficient ones!
  - Linked-In:
    - Fast-queue for < 1 hr. jobs with < 15-minute containers/tasks
    - Kill jobs that exceed limits
  - MapR (Hadoop Vendor):
    - Guess job-size using input data-size
    - Schedule small jobs on busy-cluster





#### **Resources**

- Effective Multi-Tenant Distributed Systems, Ch. 2; Chad Carson & Sean Suchter
- Distributed Systems, 2nd Edition, Ch. 1; Sukumar Ghosh
- IBM Blue Gene Wikipedia
- What Is Apache Hadoop? Databricks
- MapReduce Tutorial | Mapreduce Example in Apache Hadoop | Edureka
- Apache Hadoop 3.3.2 Apache Hadoop YARN
- Architecture of Next Generation Apache Hadoop MapReduce Framework PDF Free Download (docplayer.net)
- <u>Hybrid model for tasks scheduling in distributed real time system | SpringerLink</u>
- <u>Distributed computing functions, advantages, types, and applications IONOS</u>
- How Amazon Quietly Powers The Internet (forbes.com)
- Usage Statistics and Market Share of Amazon as Web Hosting Provider, May 2022 (w3techs.com)