Scheduling and Resource Management

Ashvin Goel

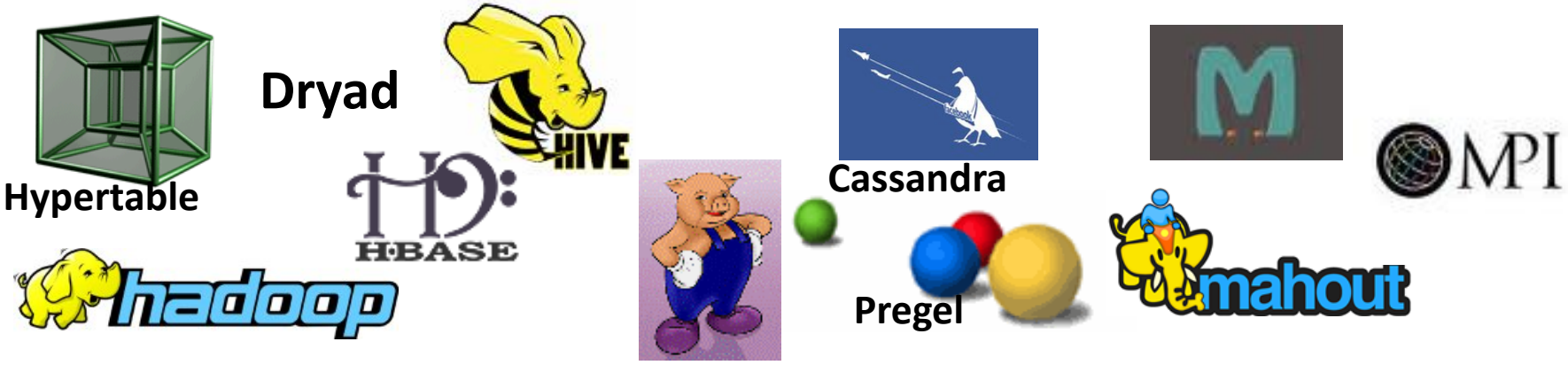
Electrical and Computer Engineering
University of Toronto

ECE1724

Some slides adapted from Ion Stoica

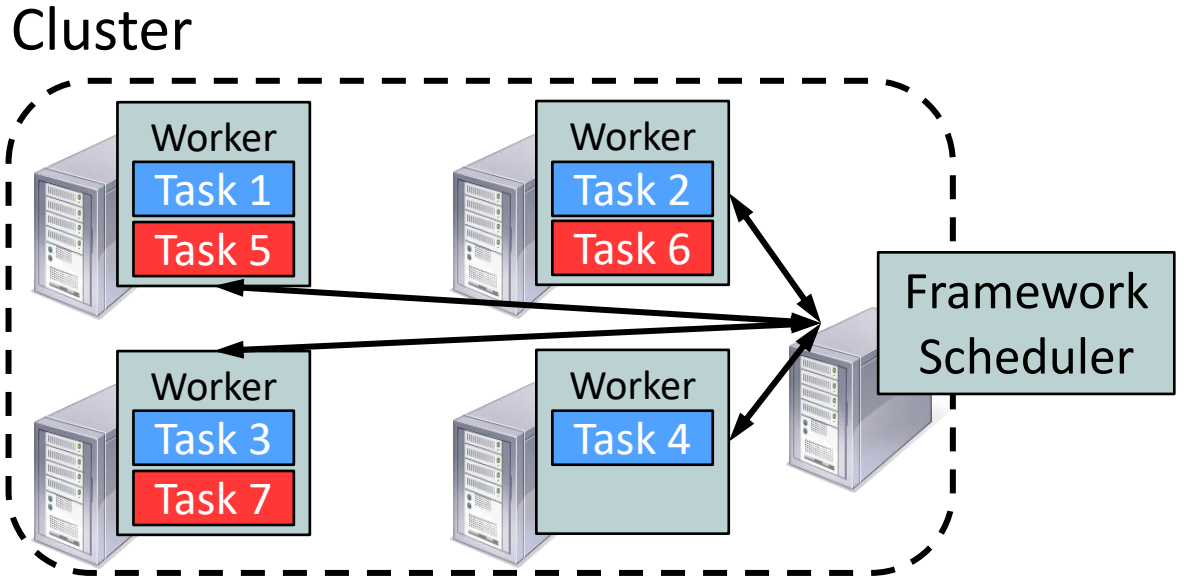
Cluster Computing Frameworks

- Lots of different cluster computing frameworks
- No single framework optimal for all applications



Framework Computation Model

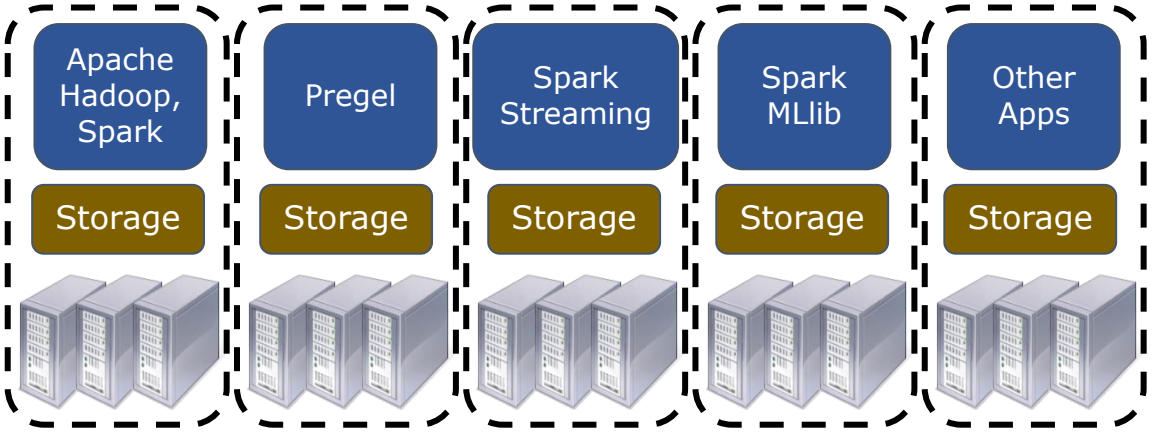
- A framework (e.g., Hadoop, Spark) manages jobs in a computer cluster using a scheduler and a set of workers
  - A job consists of one or more operators
  - An operator (e.g., map, reduce) runs one or more tasks in parallel
  - A task is implemented using one or more processes on a worker



Job 1: Tasks 1, 2, 3, 4
Job 2: Tasks 5, 6, 7

Challenge

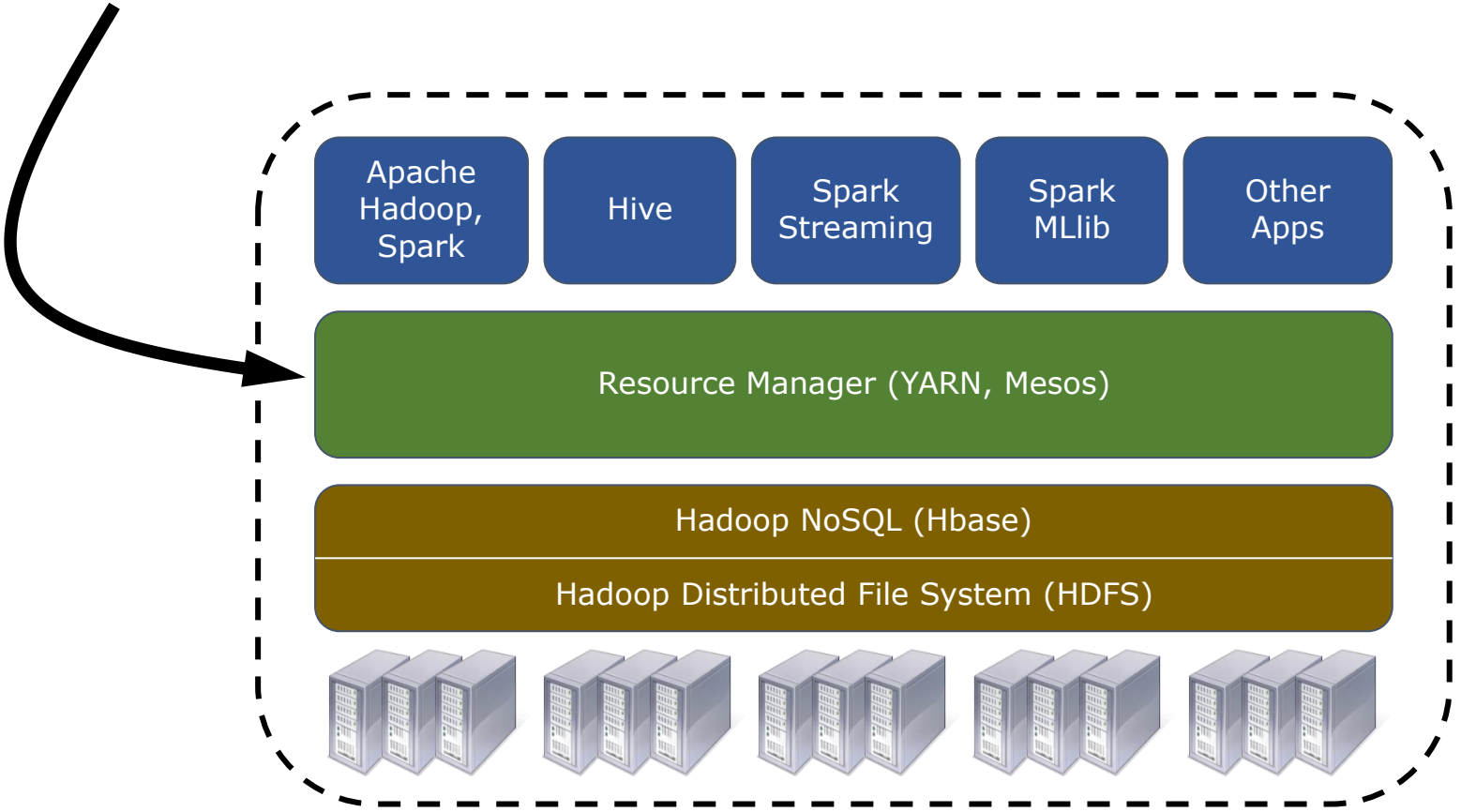
- Each framework runs on a dedicated cluster



- Problems
  - Inefficient resource usage, since no statistical multiplexing
    - E.g., Hadoop cannot use resources available from Pregel's cluster
  - Sharing data requires copying data or access to remote storage

Solution: Distributed Resource Management

- Use a resource manager that enables different frameworks to share cluster



Key Idea

- Hierarchical resource management
  - Each framework manages and runs tasks using containers
    - A container is an abstraction for virtualizing nodes, provides a certain number of CPUs, amount of memory, disk, network bandwidth, etc.
  - Distributed resource manager assigns one or more containers to physical nodes
- Similar to operating systems
  - Each process manages and runs requests using threads
    - A thread is an abstraction for virtualizing CPU cores
  - OS assigns one or more threads to CPU cores

Key Idea

- Hierarchical resource management
  - Each framework manages and runs tasks using containers
    - A container is an abstraction for virtualizing nodes, provides a certain number of CPUs, amount of memory, disk, network bandwidth, etc.
    - Implements scheduling mechanism, e.g., create task, run task
  - Distributed resource manager assigns one or more containers to physical nodes
    - Implements scheduling policy, e.g., allocate container, place container

Challenges

- Use resources efficiently
  - Ensure that allocation requests are satisfied when free resources are available
  - Ensure locality, e.g., run tasks on nodes with input data
- Support diverse frameworks
  - Each framework uses its own scheduler
  - Some frameworks are adaptive, can use any resource available
  - Need to ensure resource isolation across frameworks
- Support various resource allocation policies
  - E.g., fairness across frameworks
  - E.g., manage short and long jobs

More Challenges

- Support heterogeneous hardware
  - E.g., machine have different # of cores, DRAM, GPUs
- Scale to large number of nodes
  - Scale across data centers
- Handle node failures, resource manager failures
- We will see even more later!