Flink on a Cluster

Flink Setup

JobManager

- controls the execution of one application
- a single "master" process

ResourceManager

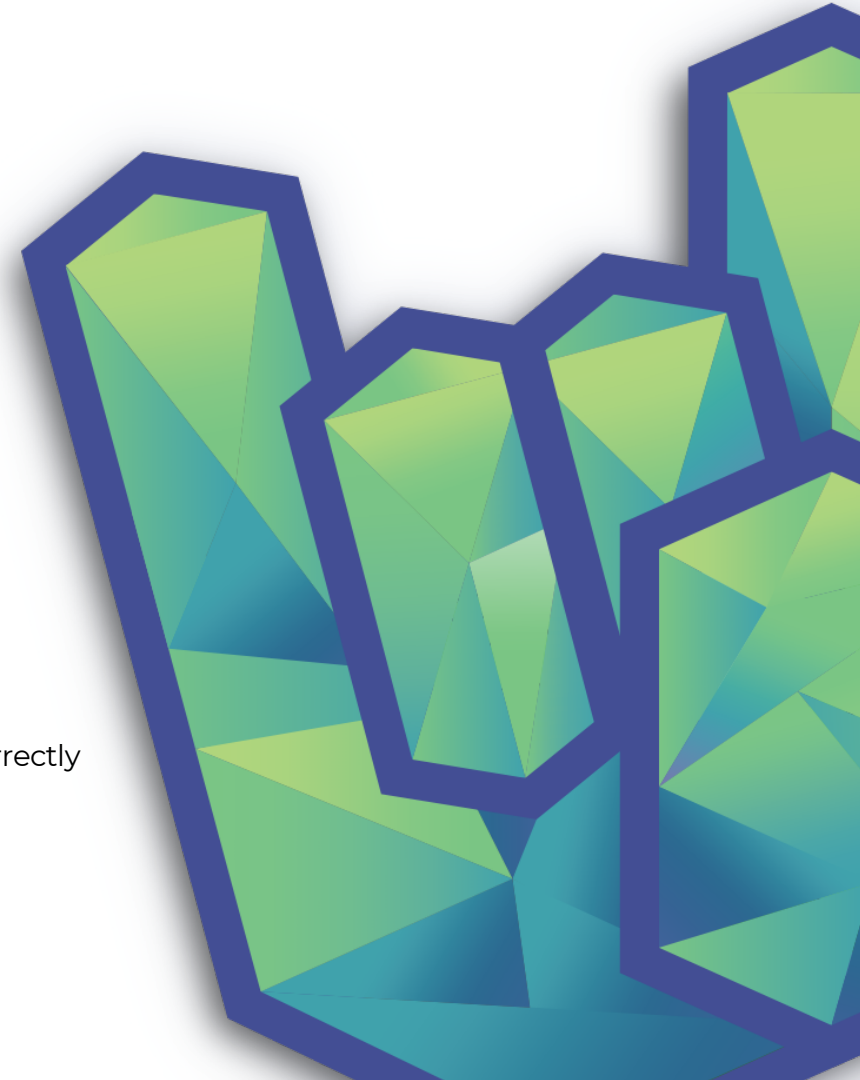
- requests and allocates computing resources
- can be run by YARN, Mesos, Kubernetes, standalone

TaskManagers

- worker processes, run tasks
- can offer *task slots* as resources
- can communicate with other TaskManagers to run the job correctly

Dispatcher

- single process *across* multiple jobs
- spawns a JobManager when an application is submitted
- offers a REST API



Submitting an Application

Submit to Dispatcher

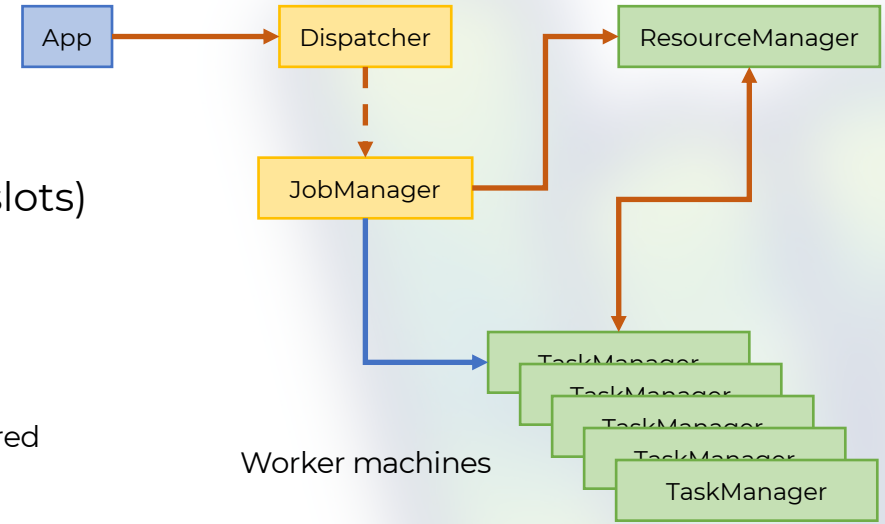
Dispatcher starts JobManager

JobManager requests resources (slots)

TaskManagers

- start
- register slots to the ResourceManager
- notifies ResourceManager of slots to be offered
- offer slots to JobManager

JobManager sends tasks to workers



Deployment Modes

Application mode

- other libraries (JARs) are included with the Flink distribution
- JobManager and cluster created just for this application, then torn down
- best for: bundled jobs, faster deployment

Per-job mode

- uses the resource provider to create a cluster for this *job*, then tear it down
- creates a JobManager per *job*
- best for: isolated, granular resource management

Session mode

- a cluster is already present, can submit multiple jobs to the same cluster
- one JobManager for the entire session
- jobs compete for the same resources
- advantage: no resource overhead spinning up cluster for every job/application

Flink rocks