

INTRODUCTION TO BIG DATA

ECAP456

Dr. Rajni Bhalla
Associate Professor

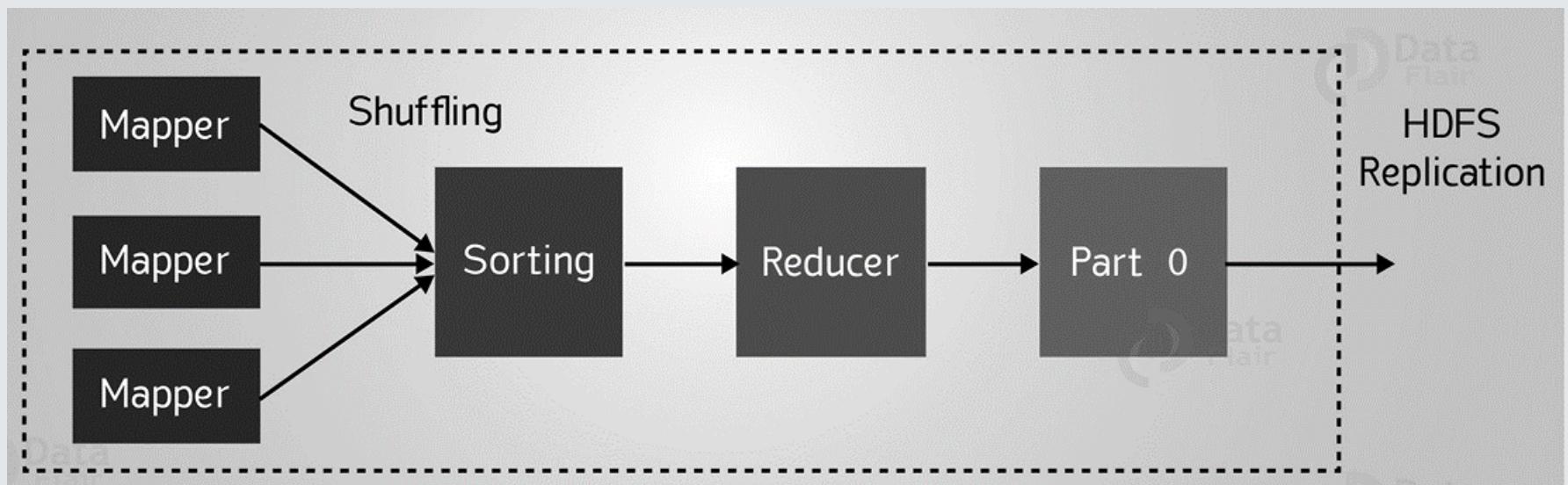
Learning Outcomes



After this lecture, you will be able to

- Learn basics of Hadoop scheduler
- Learn different types of Hadoop Scheduler
- Learn failure scenario that can occur in Hadoop
- Learn how to handle failure in mapreduce

Introduction to Hadoop Scheduler



Introduction to Hadoop Scheduler

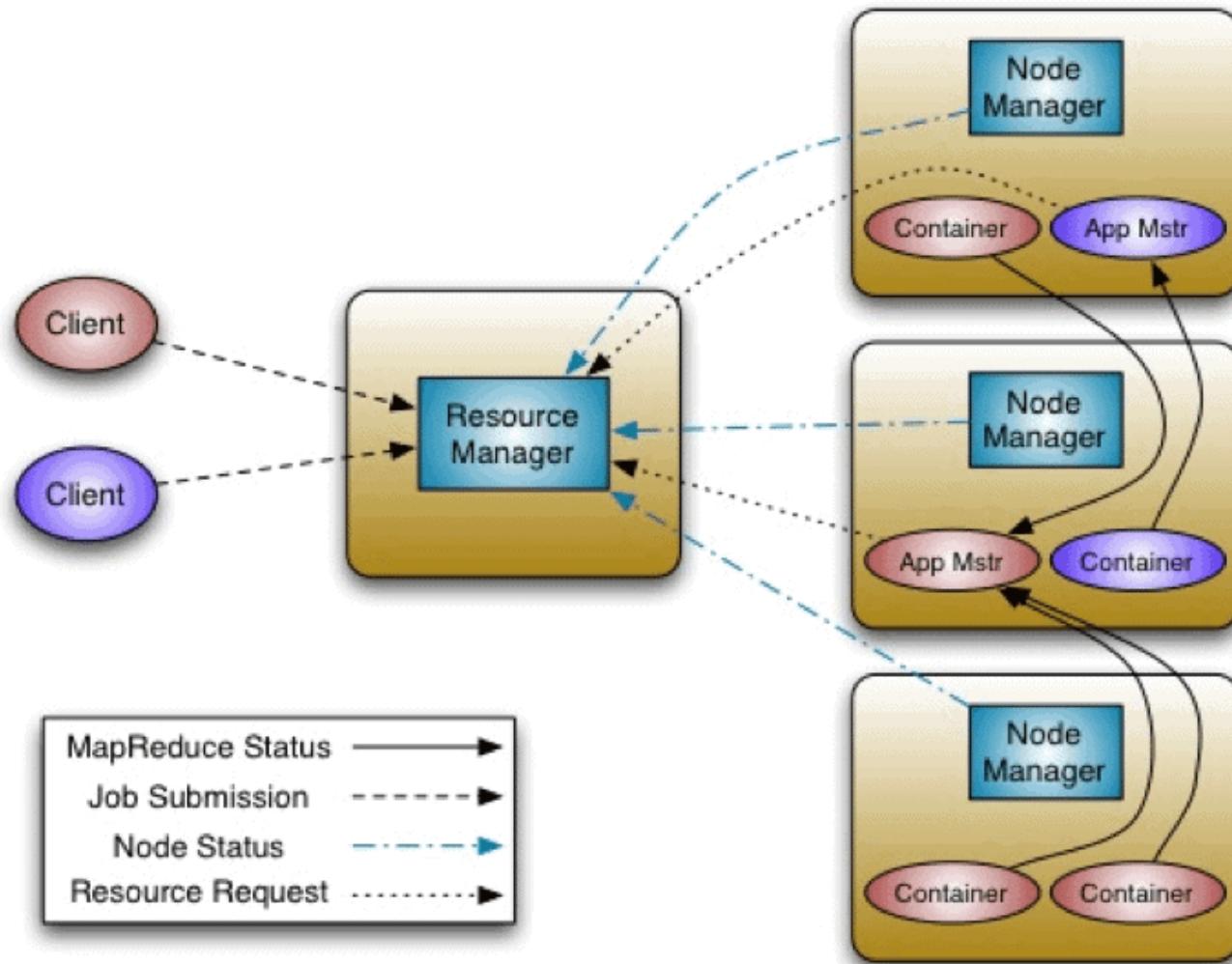
Hadoop MapReduce is responsible for :

scheduling
tasks

monitoring
them

re-executes
the failed task.

Introduction to Hadoop Scheduler



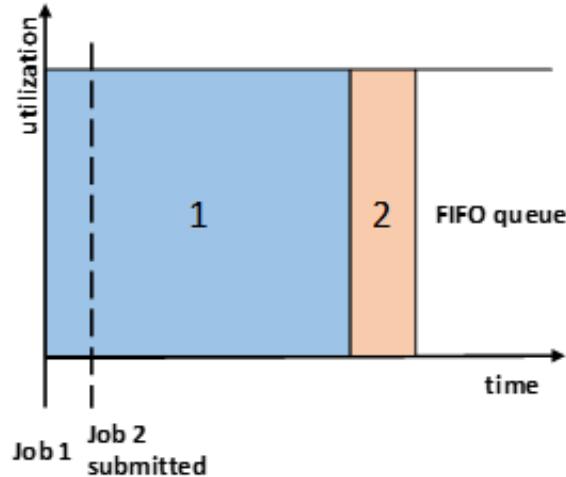
Introduction to Hadoop Scheduler

The
ResourceManager
has two main
components that are

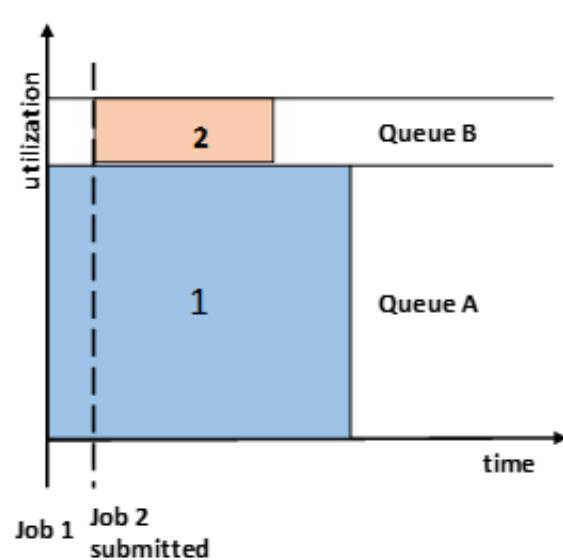
Schedulers

ApplicationsManager

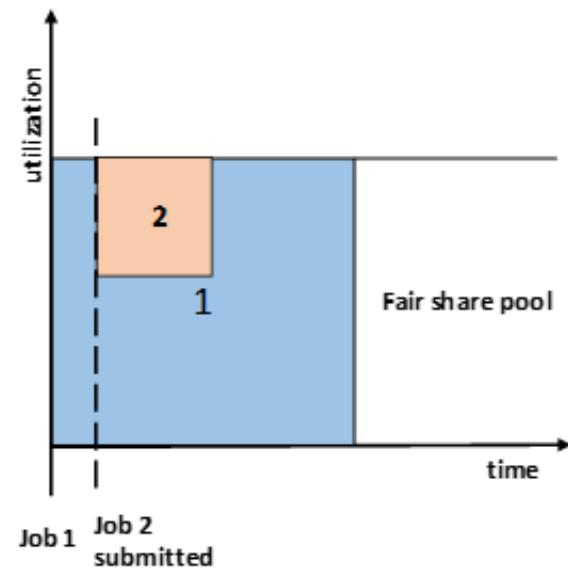
Types of HADOOP Scheduler



(a) FIFO scheduler



(b) Capacity Scheduler



(c) Fair Scheduler

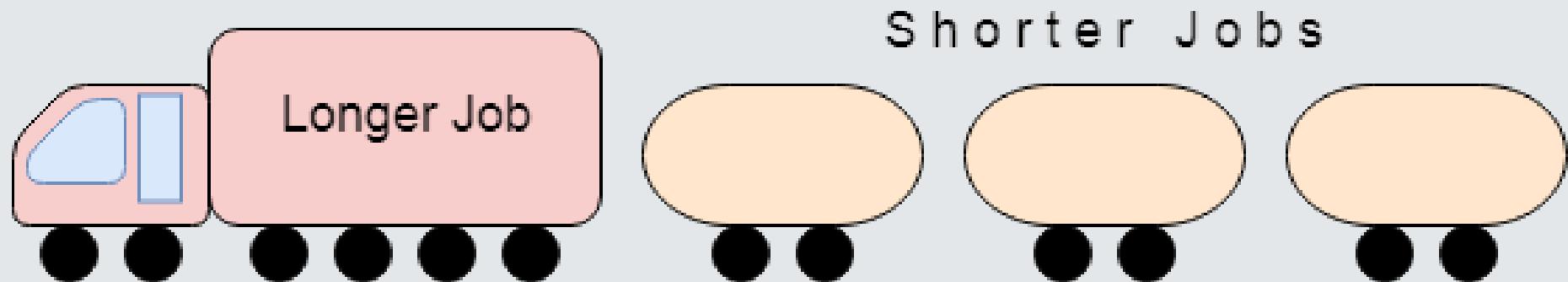
Types of HADOOP Scheduler

Fifo Scheduler

Advantages of FIFO Scheduling

- It is simple to understand and doesn't need any configuration.
- Jobs are executed in the order of their submission.

Disadvantages of FIFO Scheduling



Disadvantages of FIFO Scheduling

- It is not suitable for shared clusters. If the large application comes before the shorter one, then the large application will use all the resources in the cluster, and the shorter application has to wait for its turn. This leads to starvation.
- It does not take into account the balance of resource allocation between the long applications and short applications.

Types of HADOOP Scheduler

Capacity Scheduler

Capacity Scheduler

- It allows multiple-tenants to securely share a large Hadoop cluster.
- It is designed to run Hadoop applications in a shared, multi-tenant cluster.
- It supports hierarchical queues to reflect the structure of organizations.

Capacity Scheduler

A queue hierarchy contains three types of queues that are

root

parent

and leaf.

Capacity Scheduler

To ensure fairness and stability, it also provides limits on initialized and pending apps from a single user and queue.



Advantages of Capacity Scheduler

- It maximizes the utilization of resources and throughput in the Hadoop cluster.
- Provides elasticity for groups or organizations in a cost-effective manner.
- It also gives capacity guarantees and safeguards to the organization utilizing cluster.

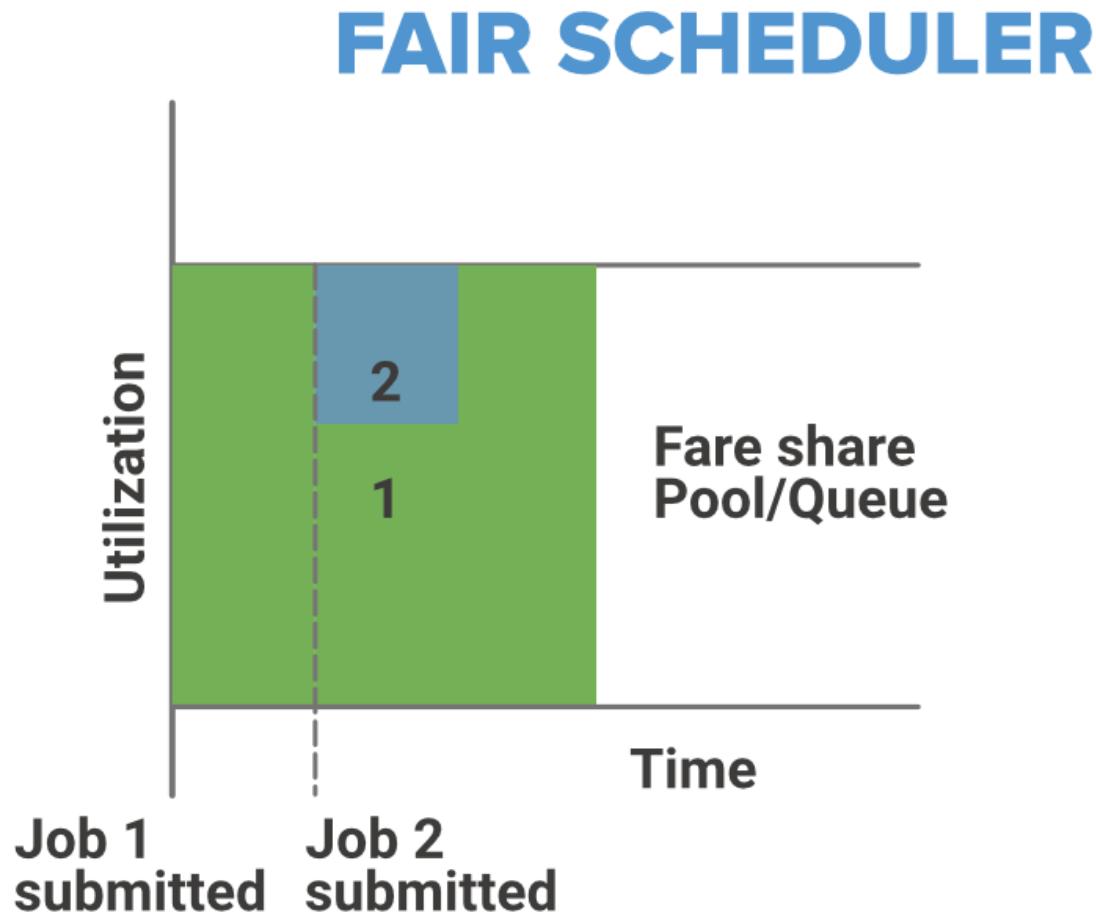
Disadvantages of Capacity Scheduler

- It is complex amongst the other scheduler.

Types of HADOOP Scheduler

Fair Scheduler

Types of HADOOP Scheduler



Advantages of Fair scheduler

- It provides a reasonable way to share the Hadoop Cluster between the number of users.
- Also, the Fair Scheduler can work with app priorities where the priorities are used as weights in determining the fraction of the total resources that each application should get.

Disadvantages of Fair scheduler

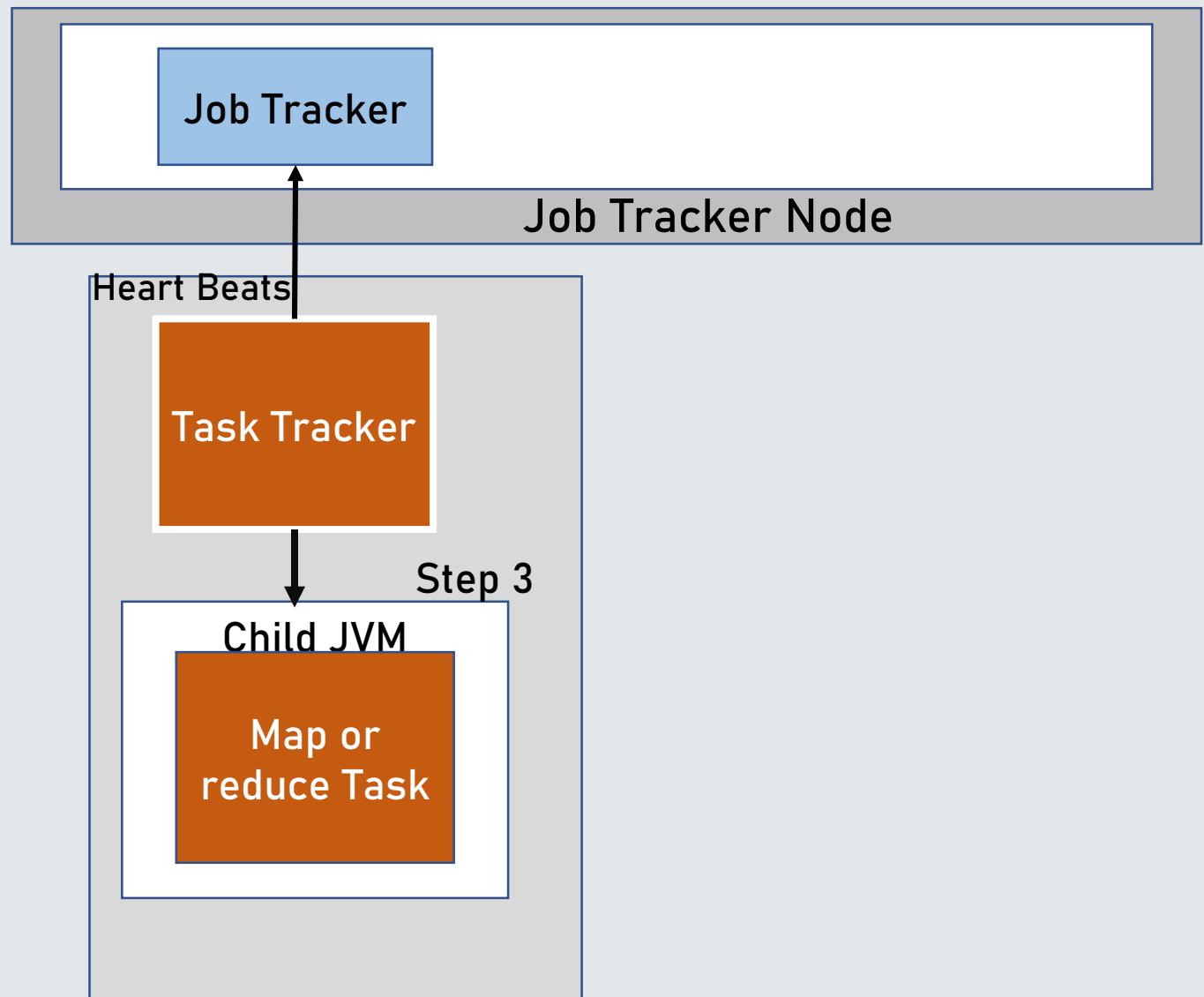
- It requires configuration.

Failures in MapReduce

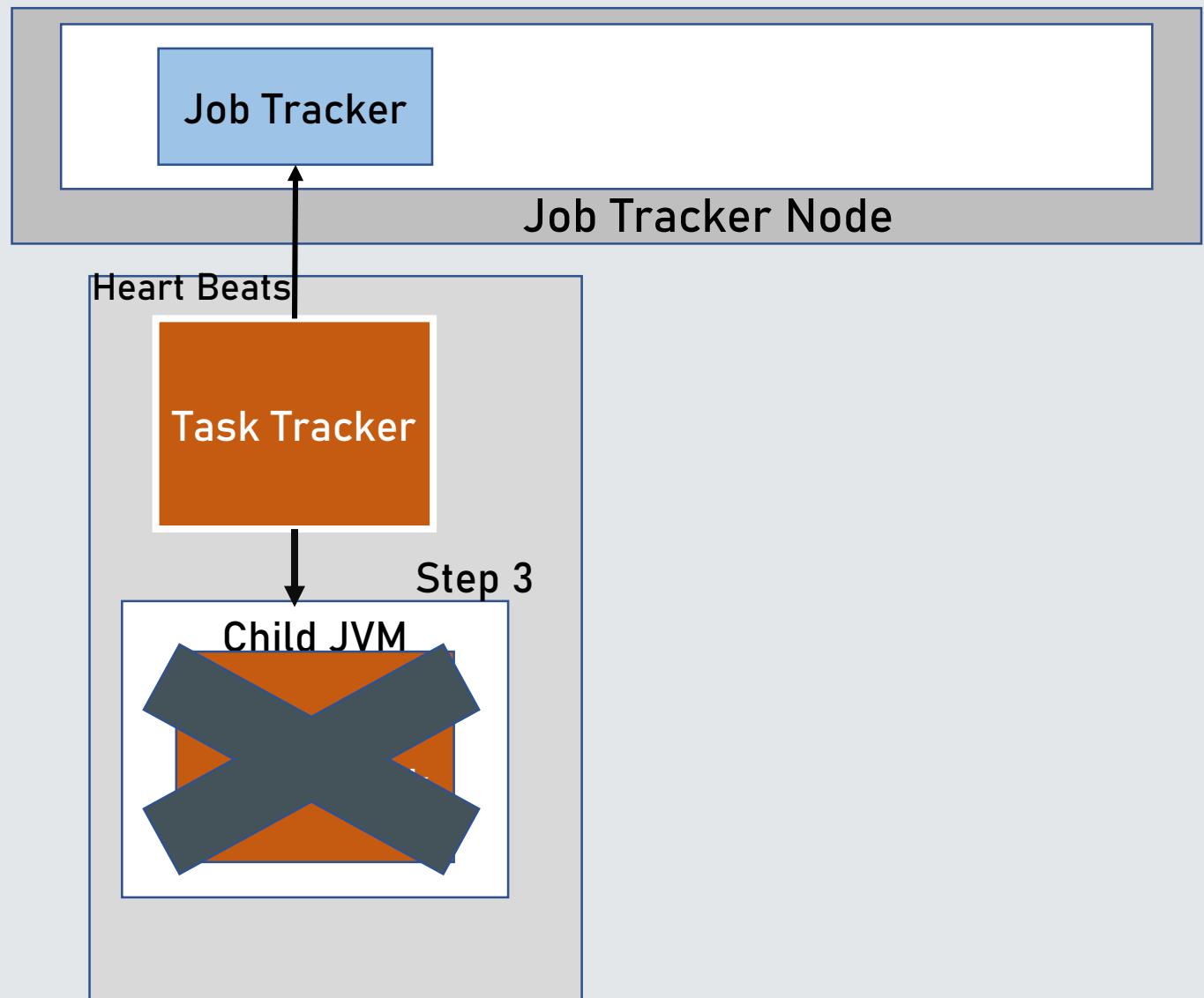
In classic MapReduce, there can be three kinds of failures

1. Failure of map reduce task
2. Failure of task tacker
3. Failure of Job Tracker

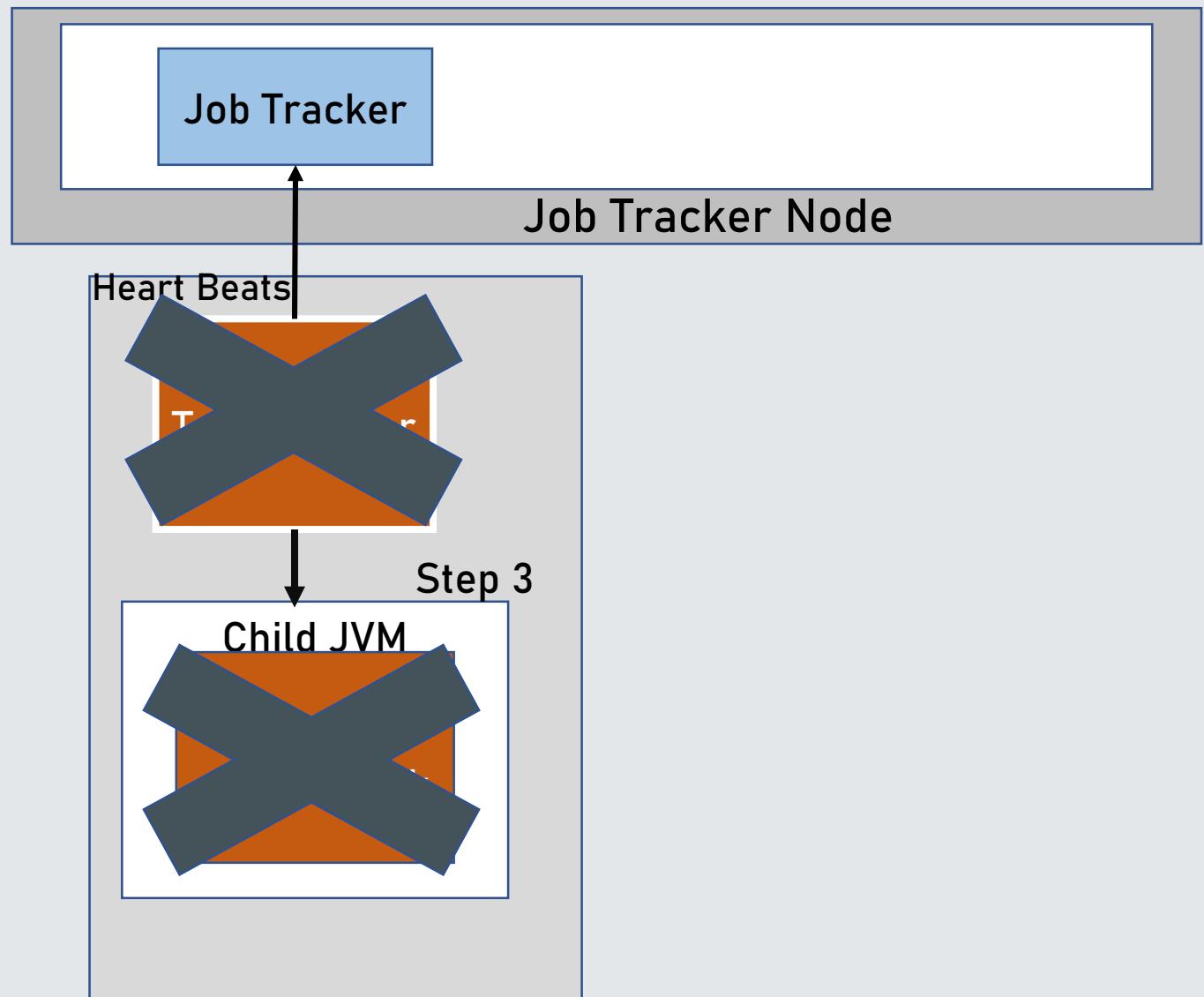
Failures in Classic Map-reduce



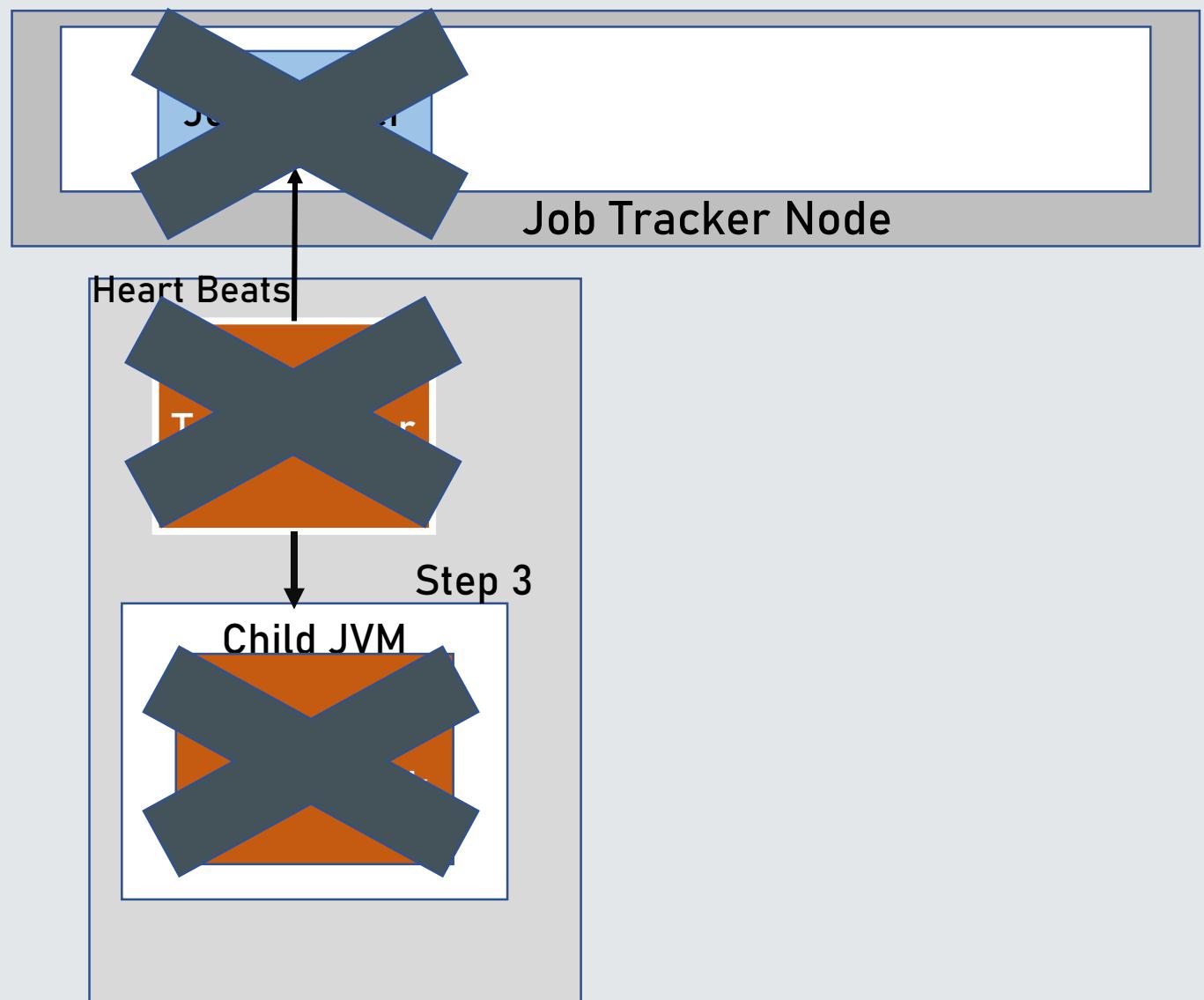
Failures in Classic Map-reduce



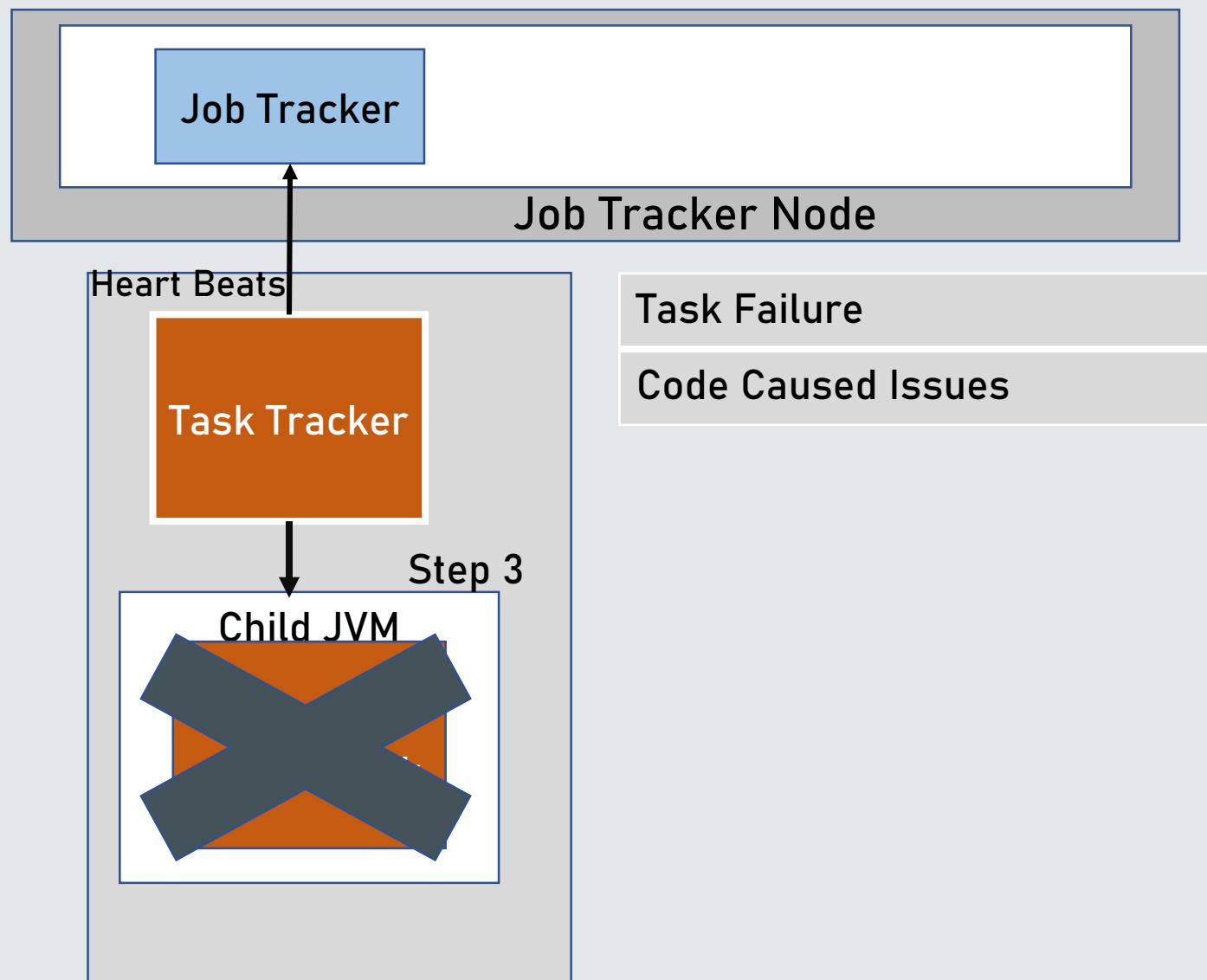
Failures in Classic Map-reduce



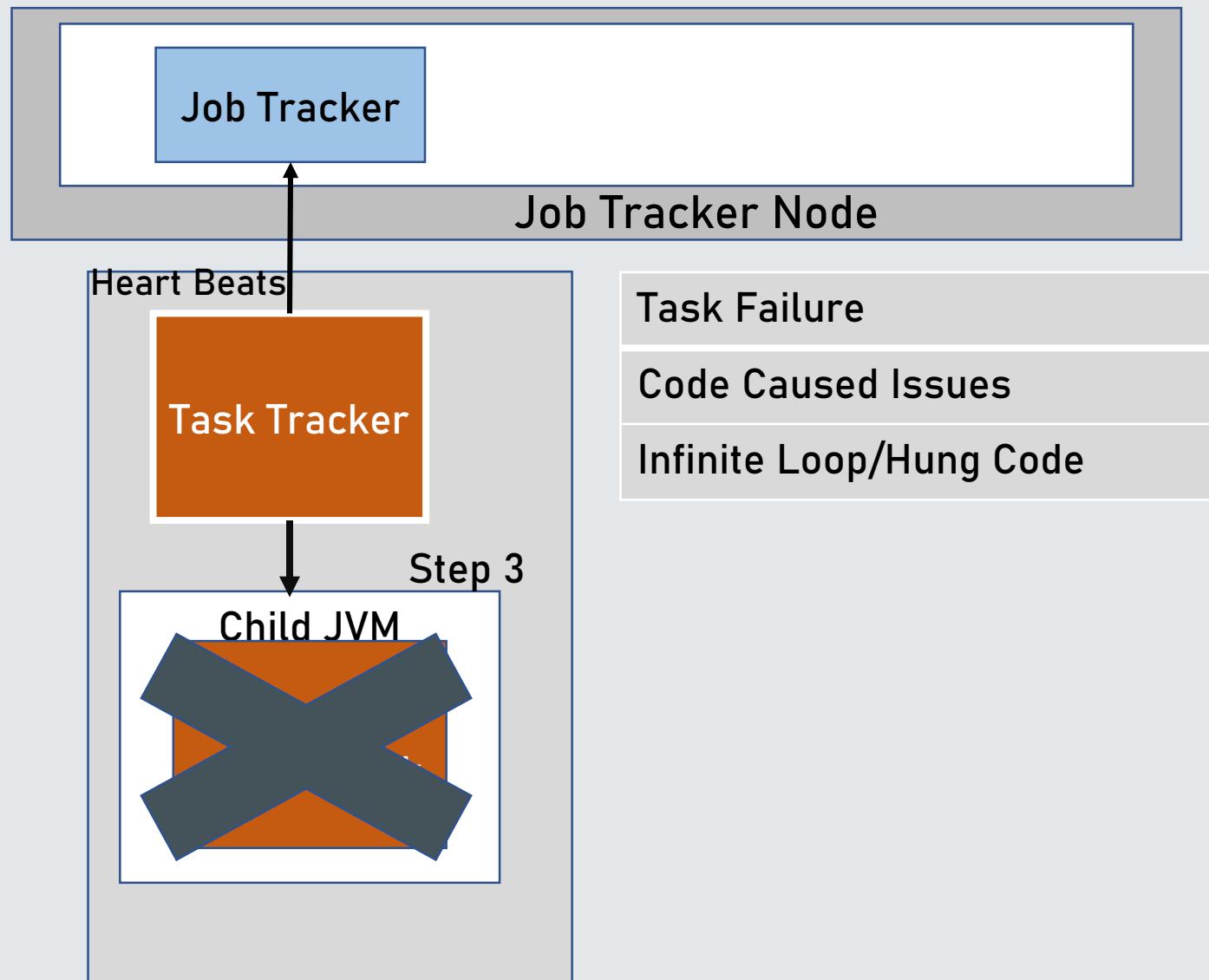
Failures in Classic Map-reduce



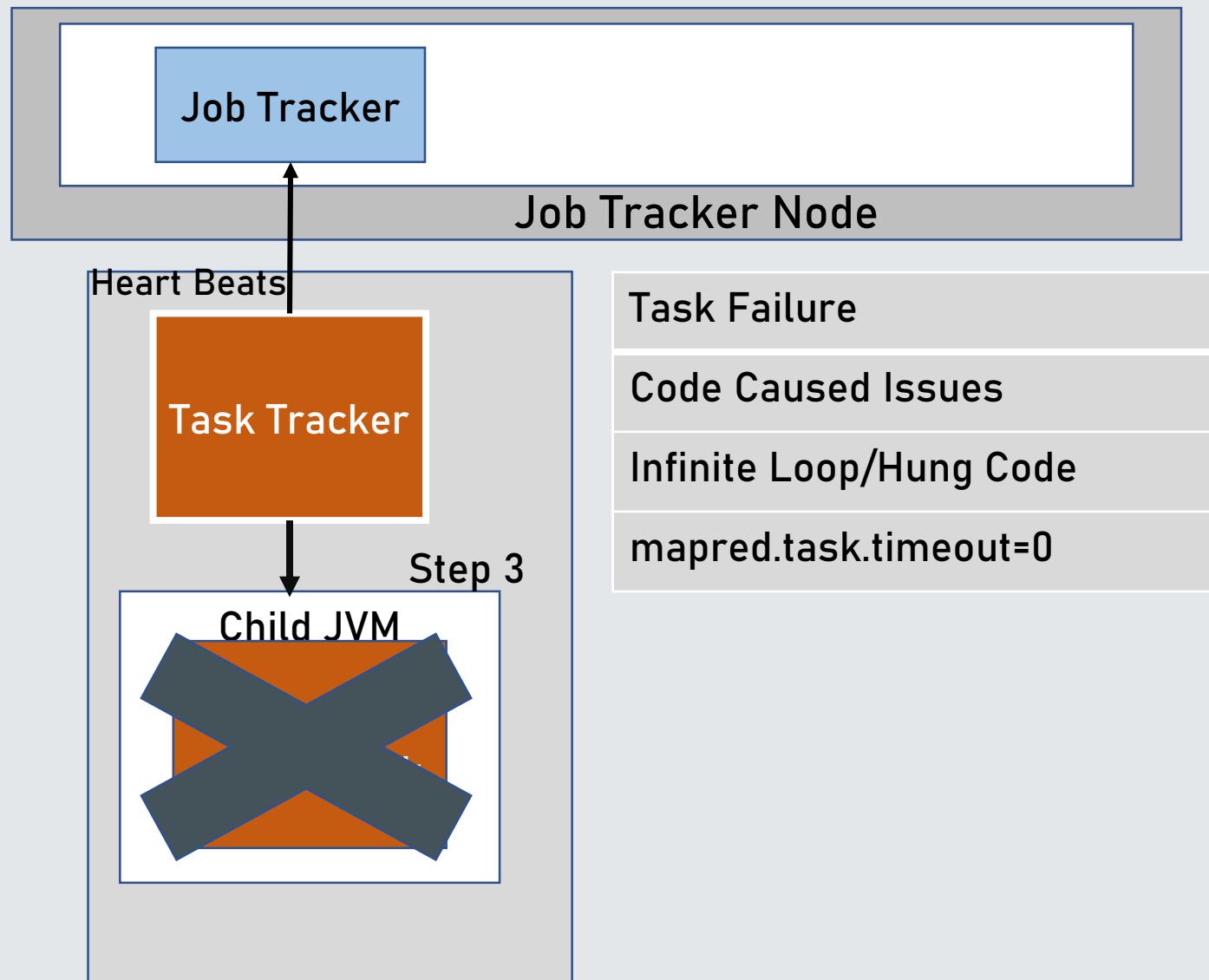
Failures in Classic Map-reduce



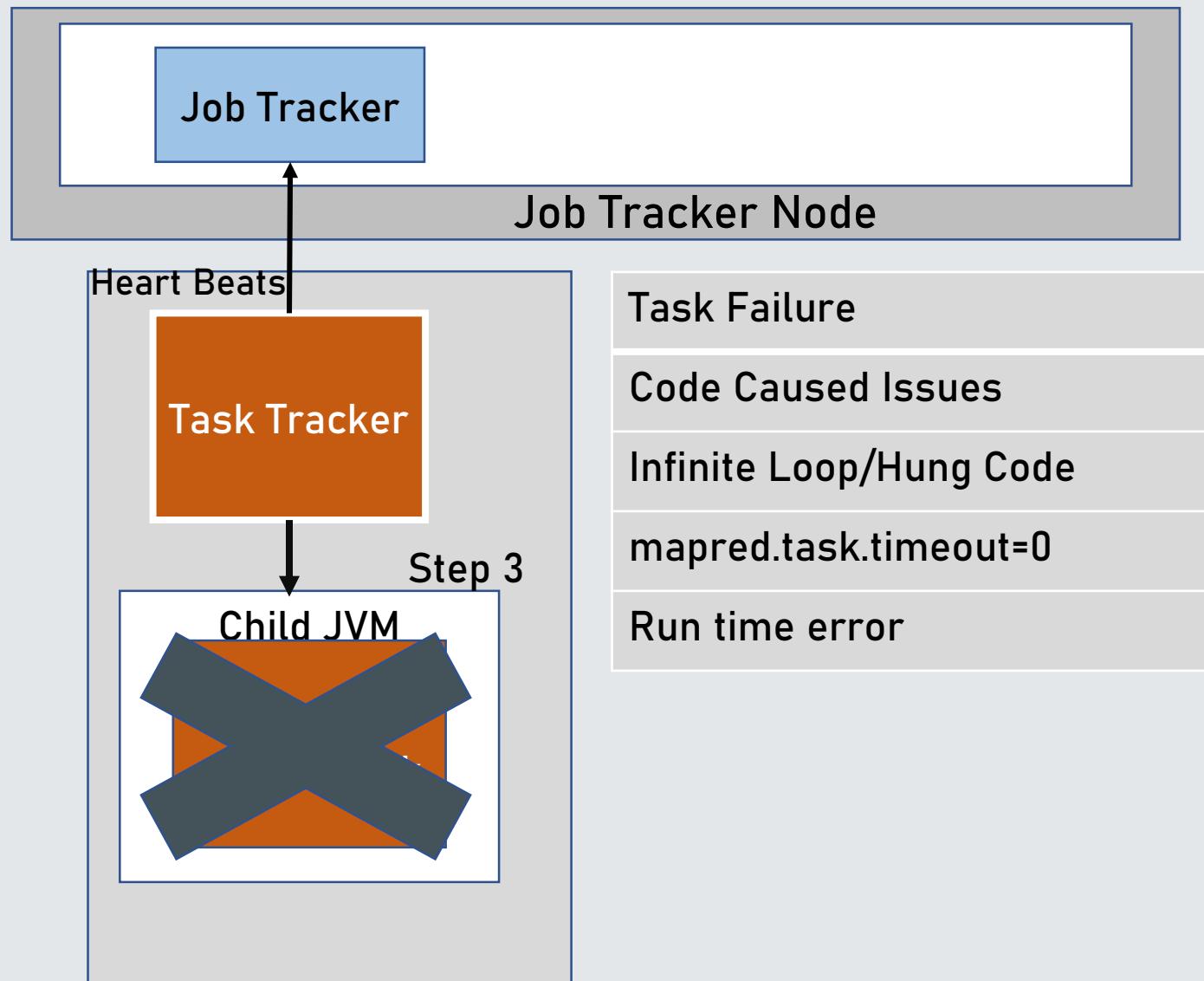
Failures in Classic Map-reduce



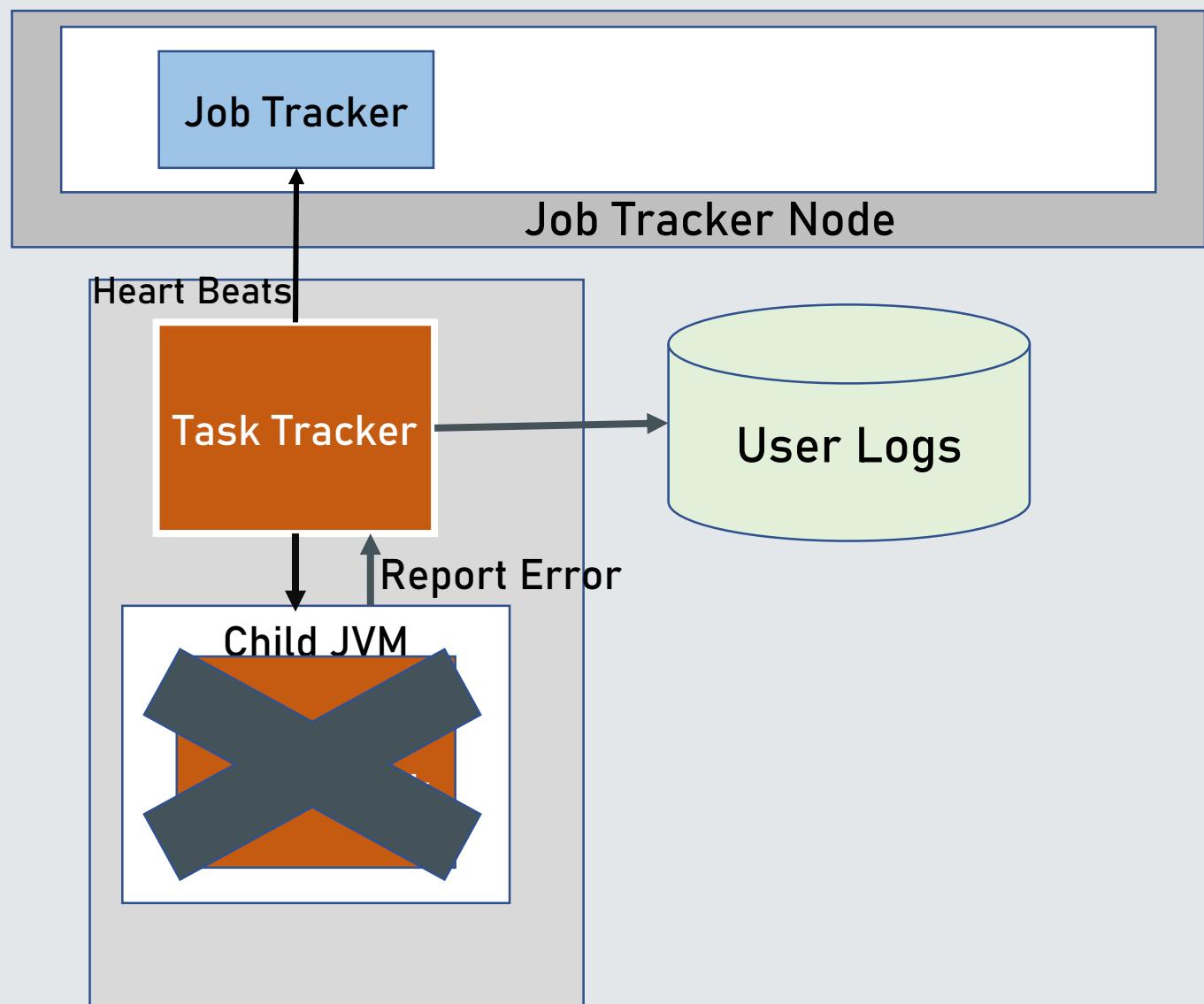
Failures in Classic Map-reduce



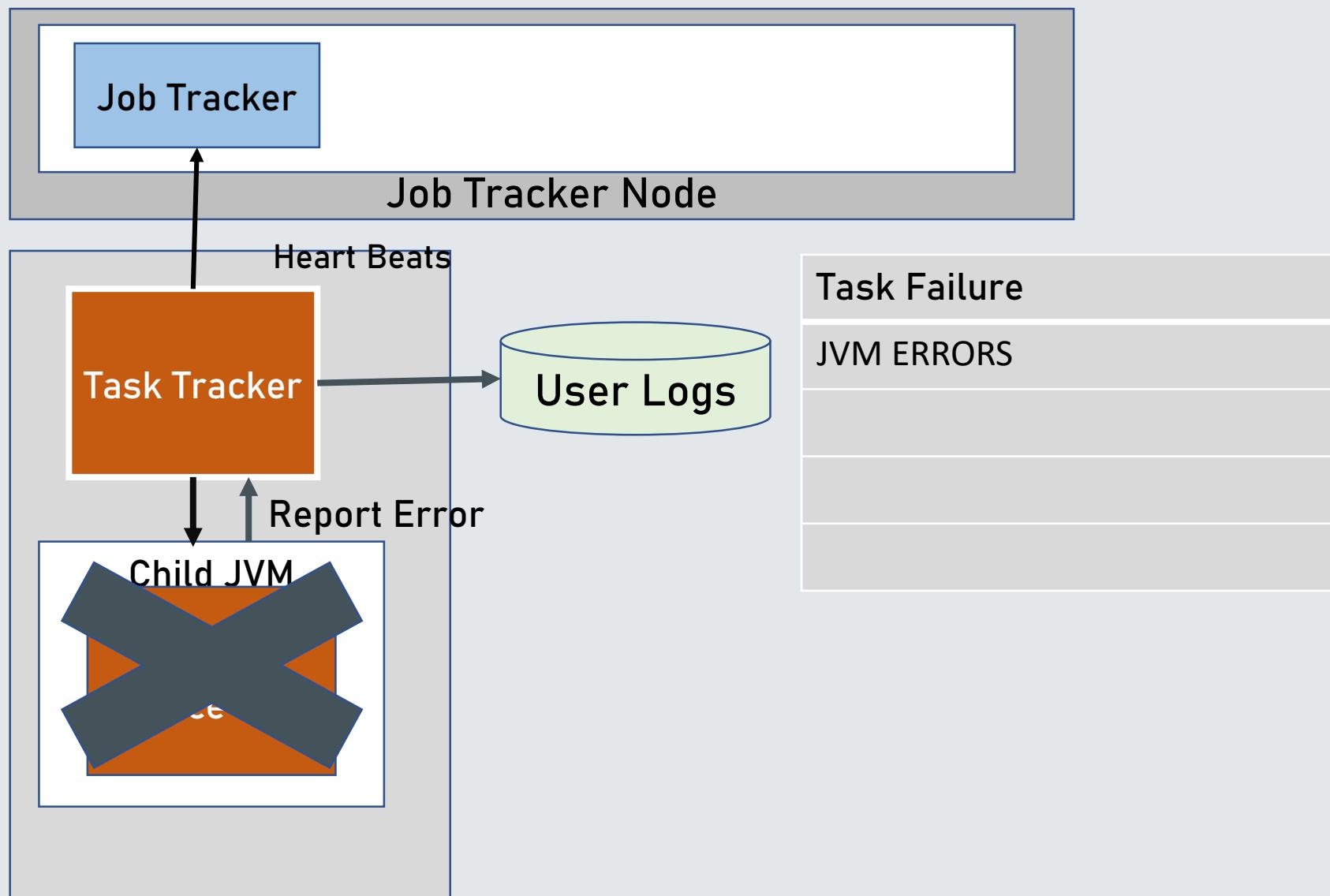
Failures in Classic Map-reduce



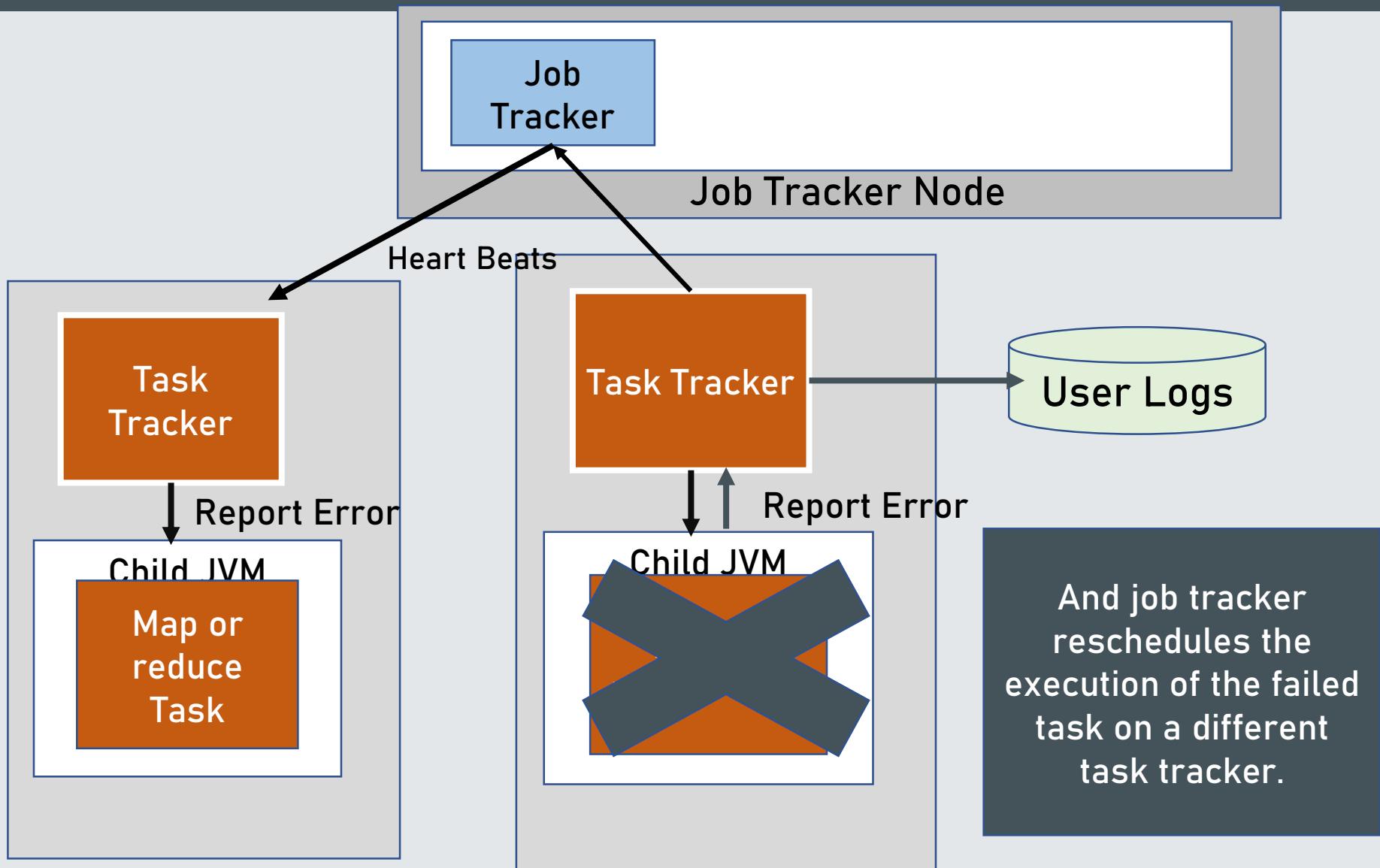
Failures in Classic Map-reduce



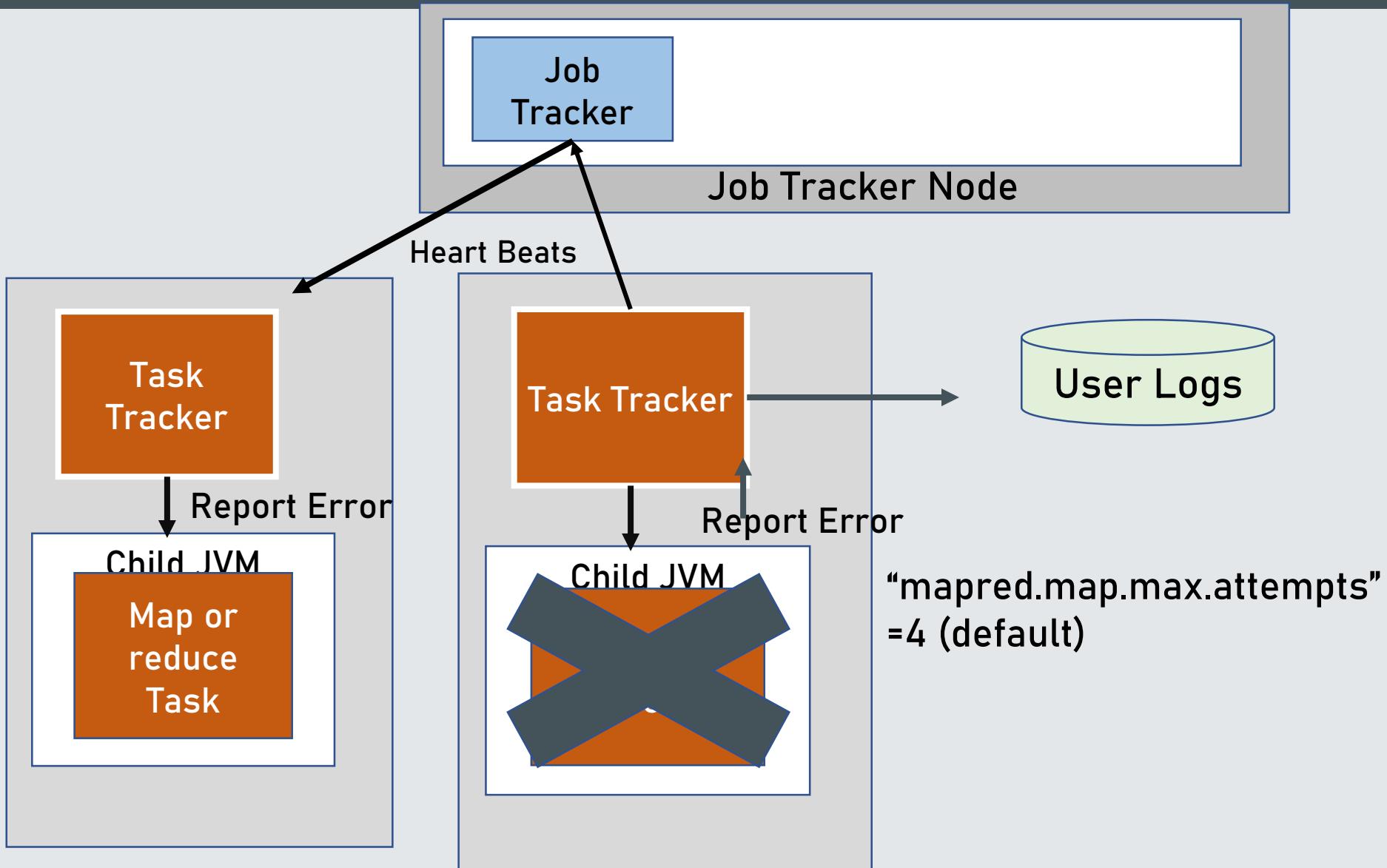
Failures in Classic Map-reduce



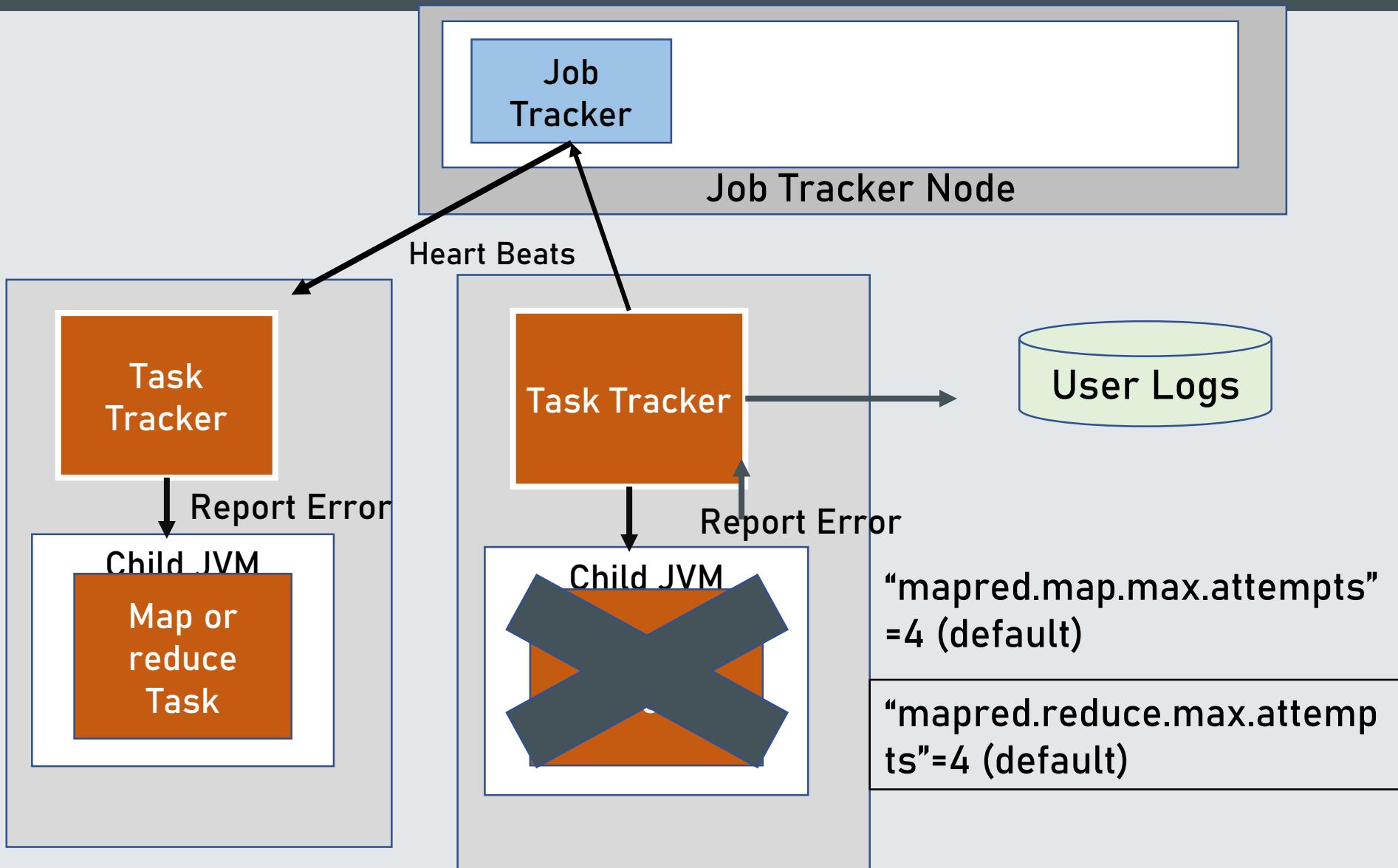
Failures in Classic Map-reduce



Failures in Classic Map-reduce

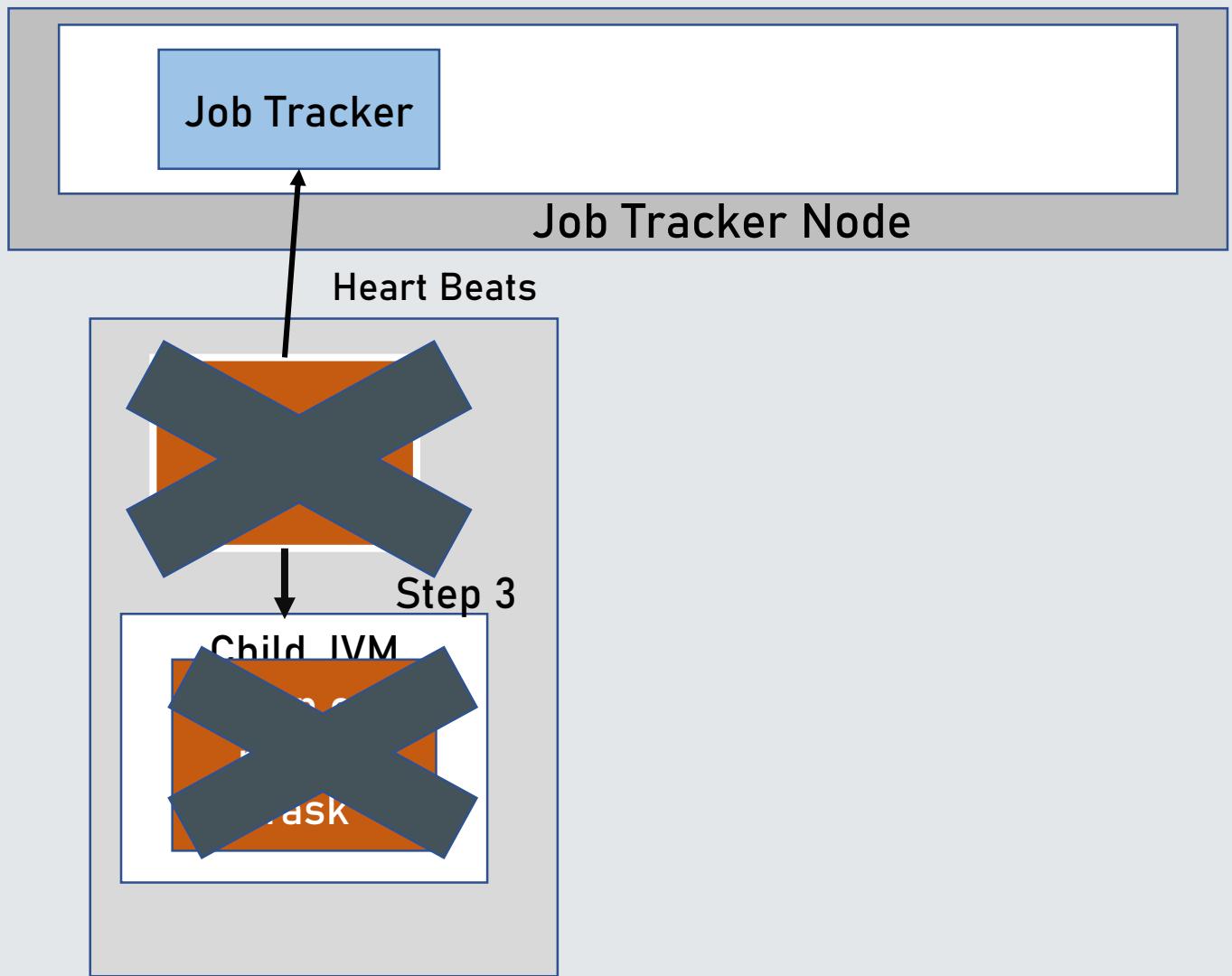


Failures in Classic Map-reduce

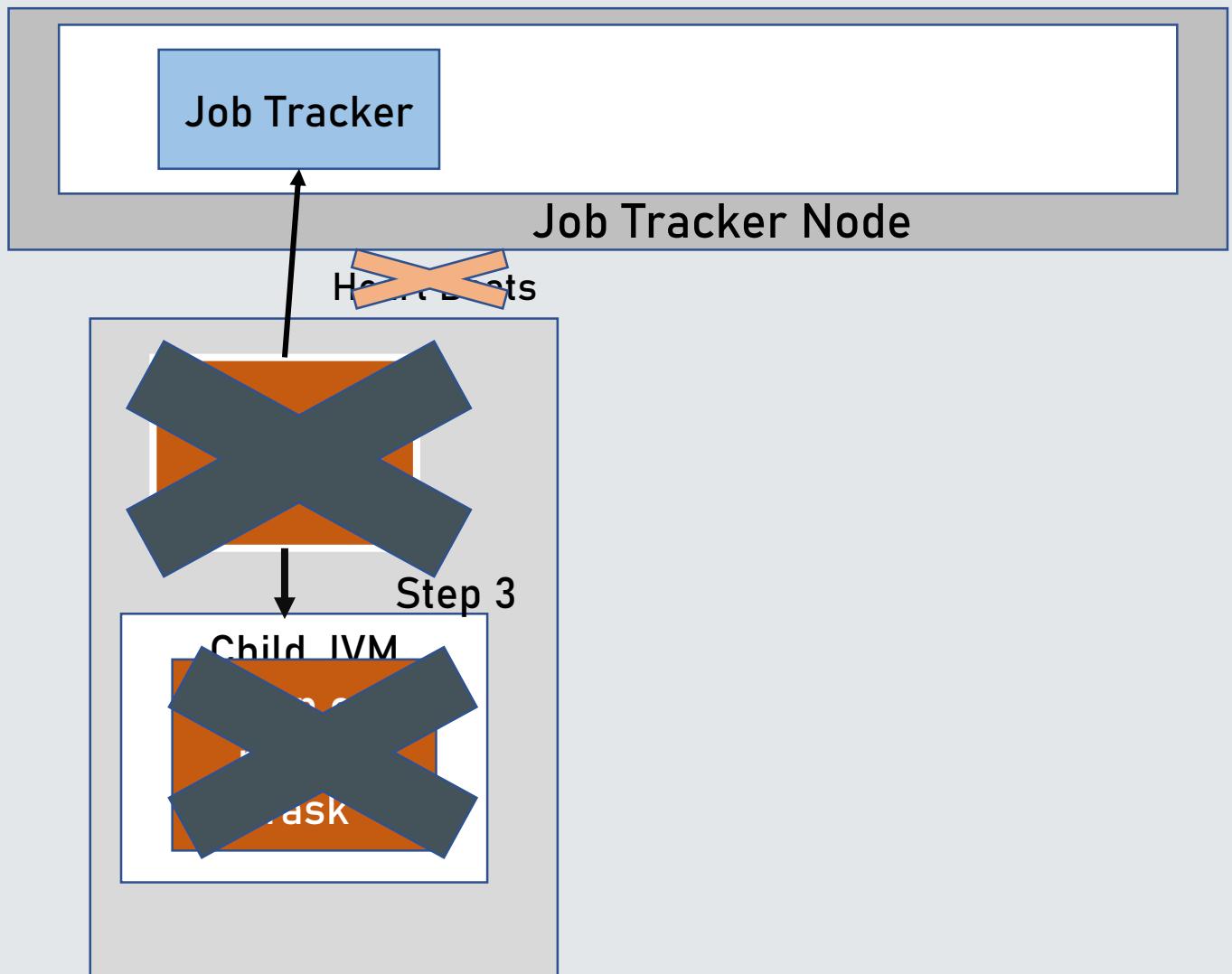


Failure of TaskTracker

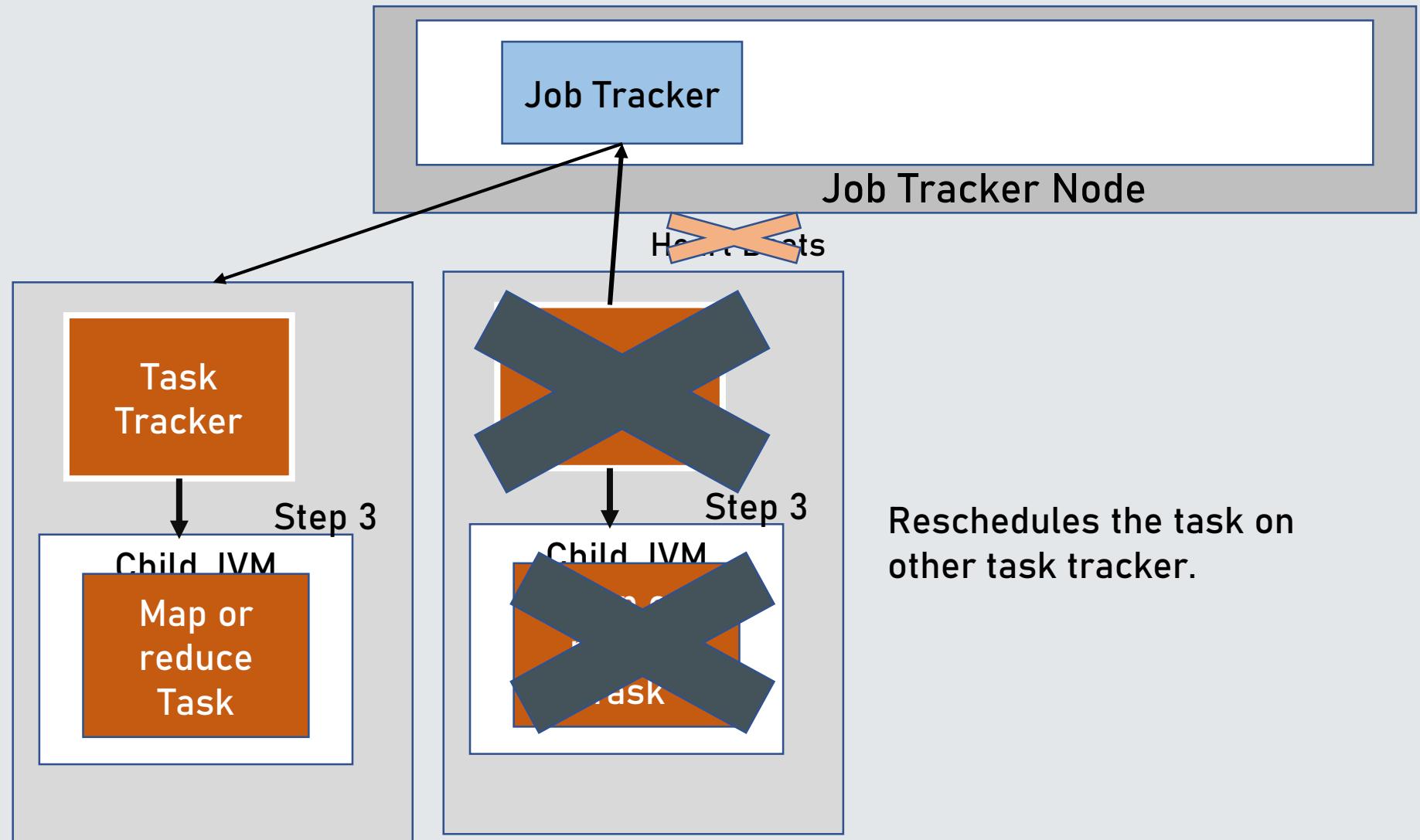
Failure of TaskTracker



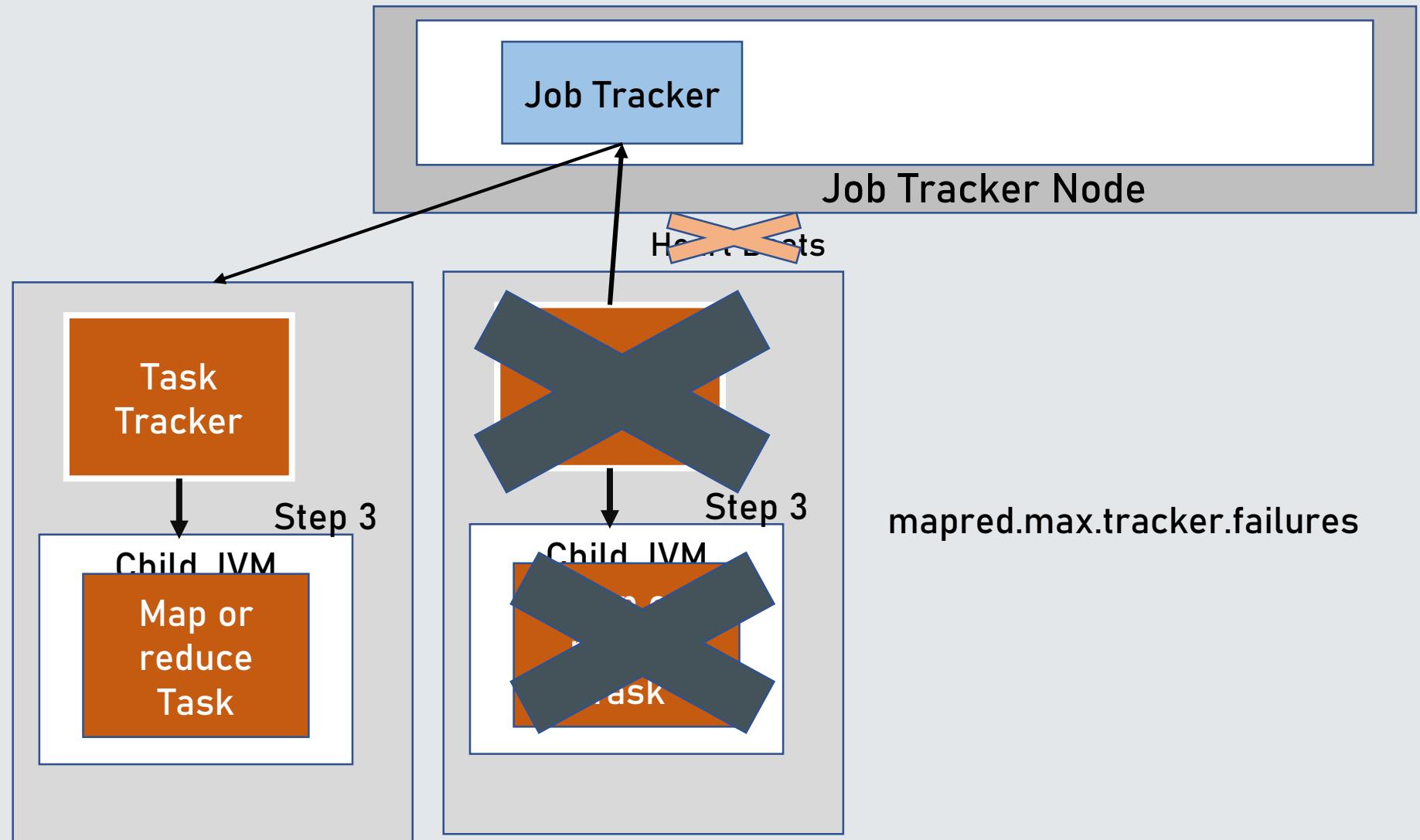
Failure of TaskTracker



Failure of TaskTracker

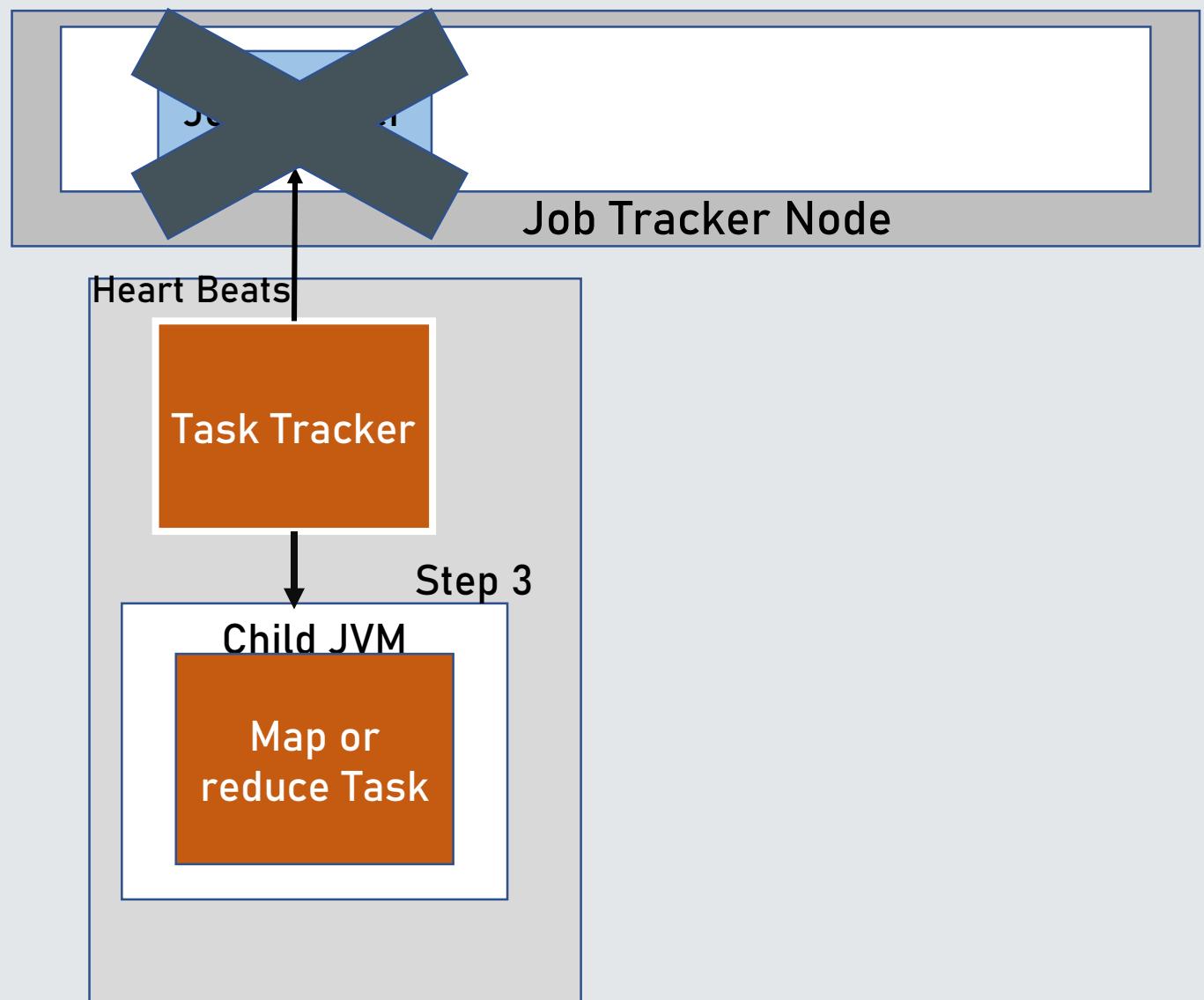


Failure of TaskTracker

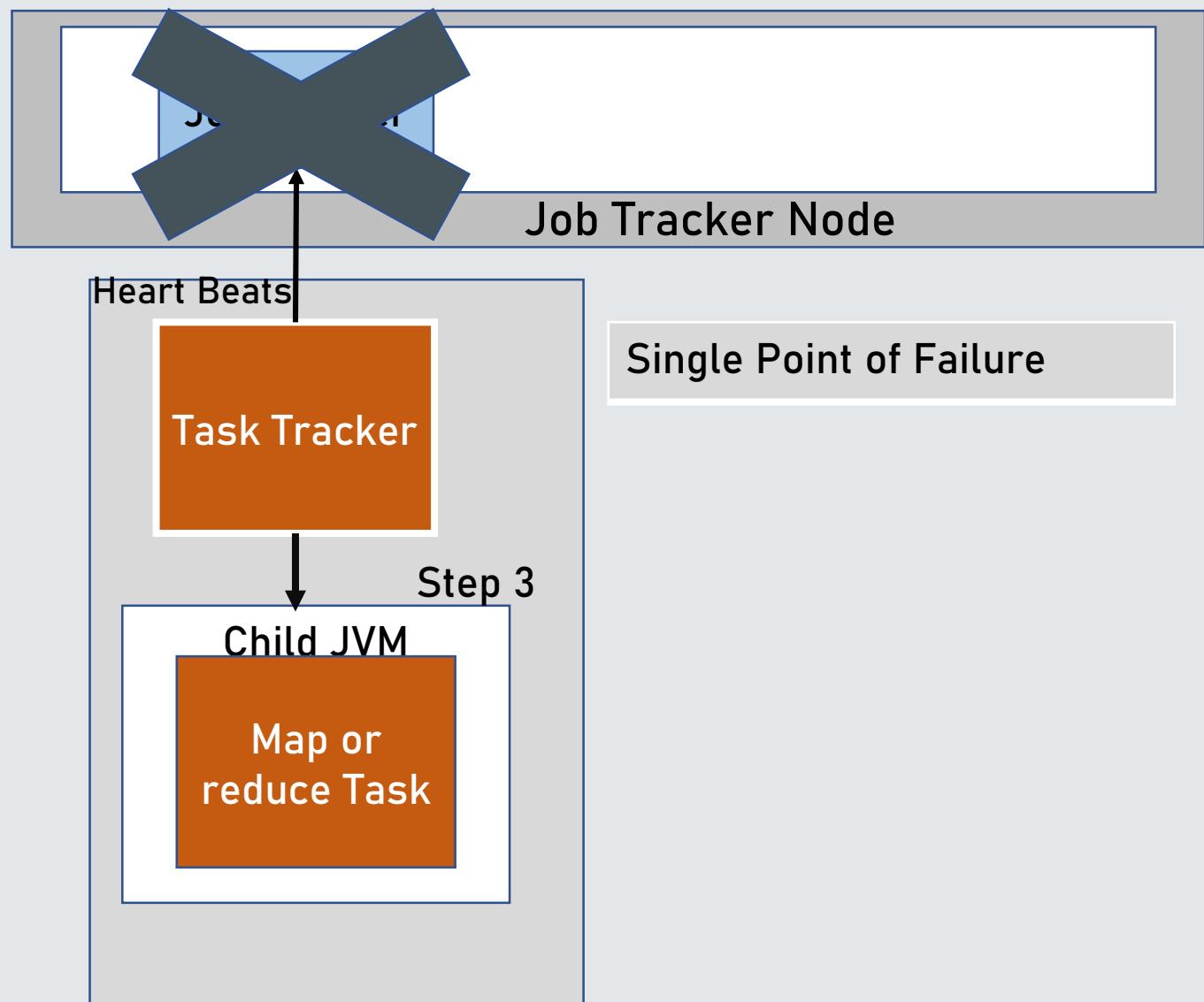


Failure of Job Tracker

Failures of JobTracker



Failures of JobTracker

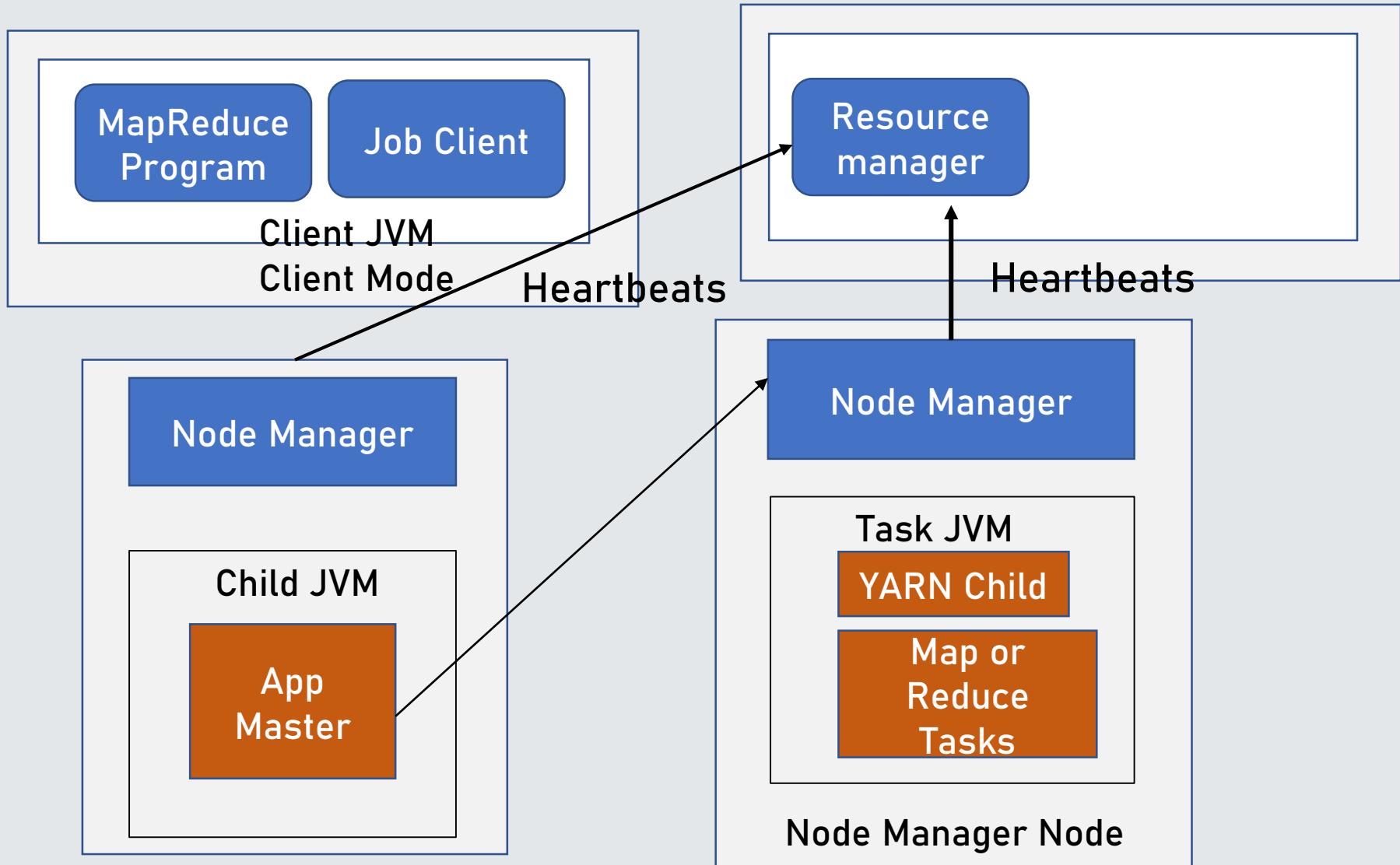


Failures in Yarn

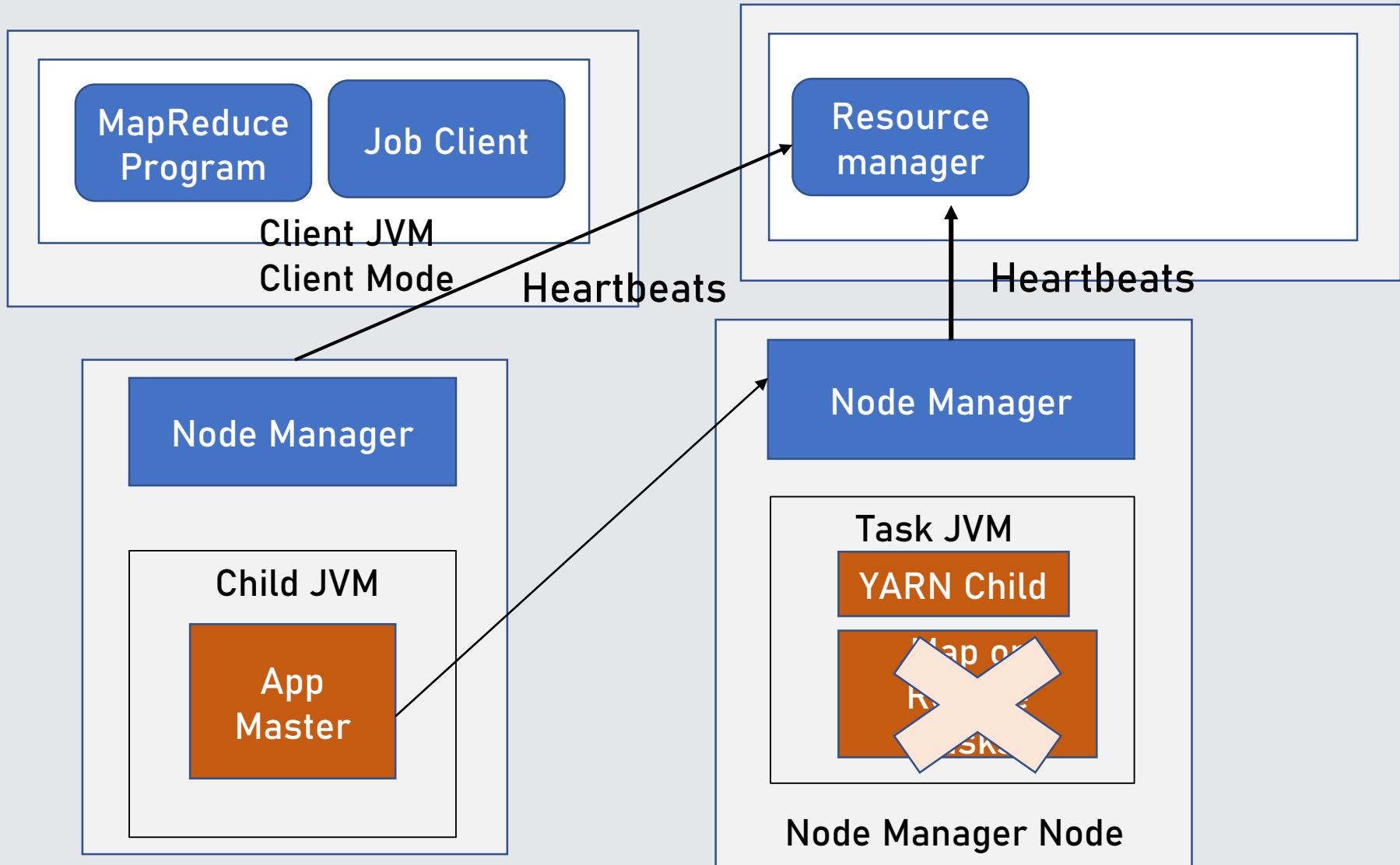
Failures in Yarn

- It can follow failure scenarios in YARN framework.
- Task failure
- Application master failure
- Node Manager Failure
- And last is resource manager failure

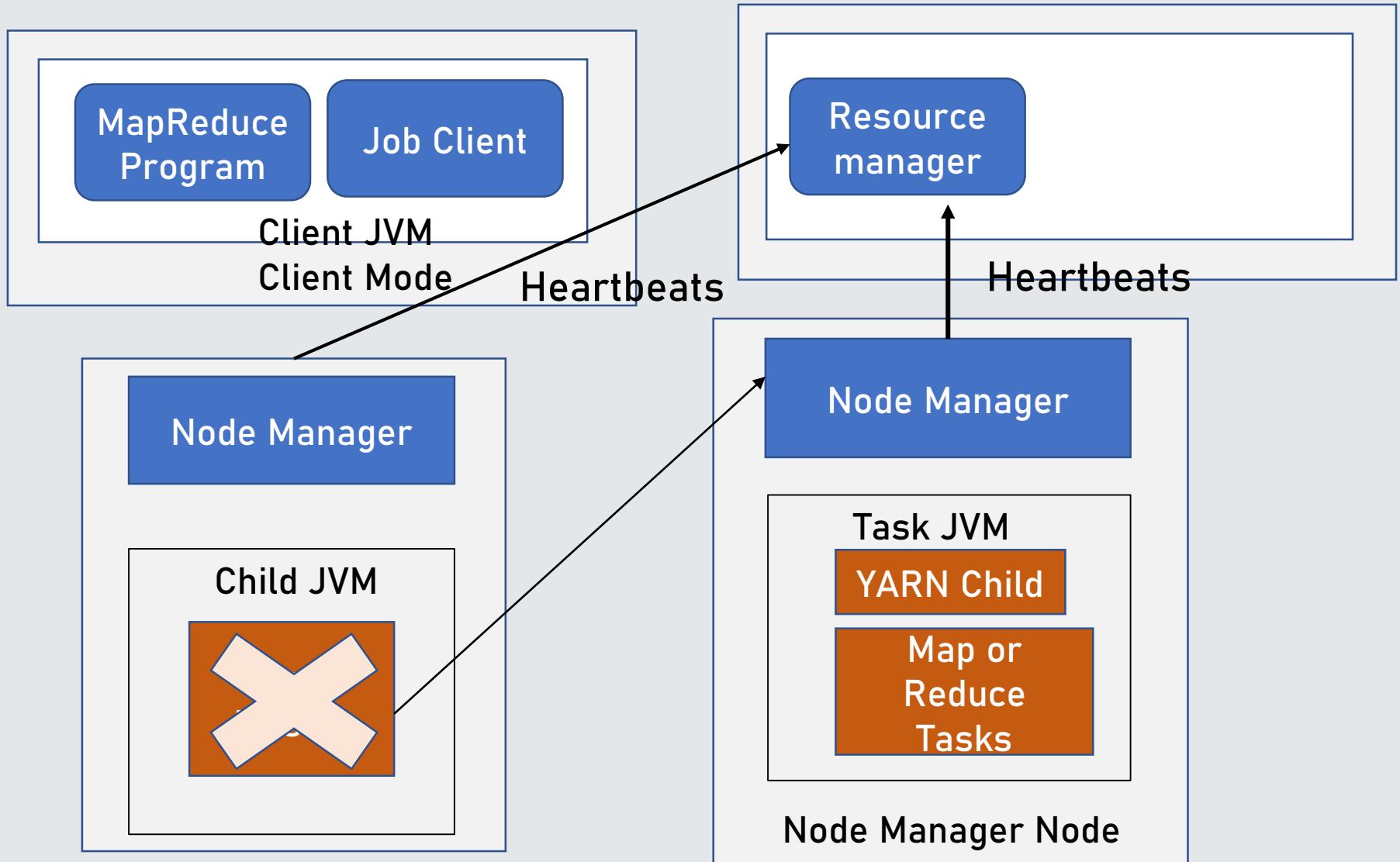
Failure Scenarios in YARN



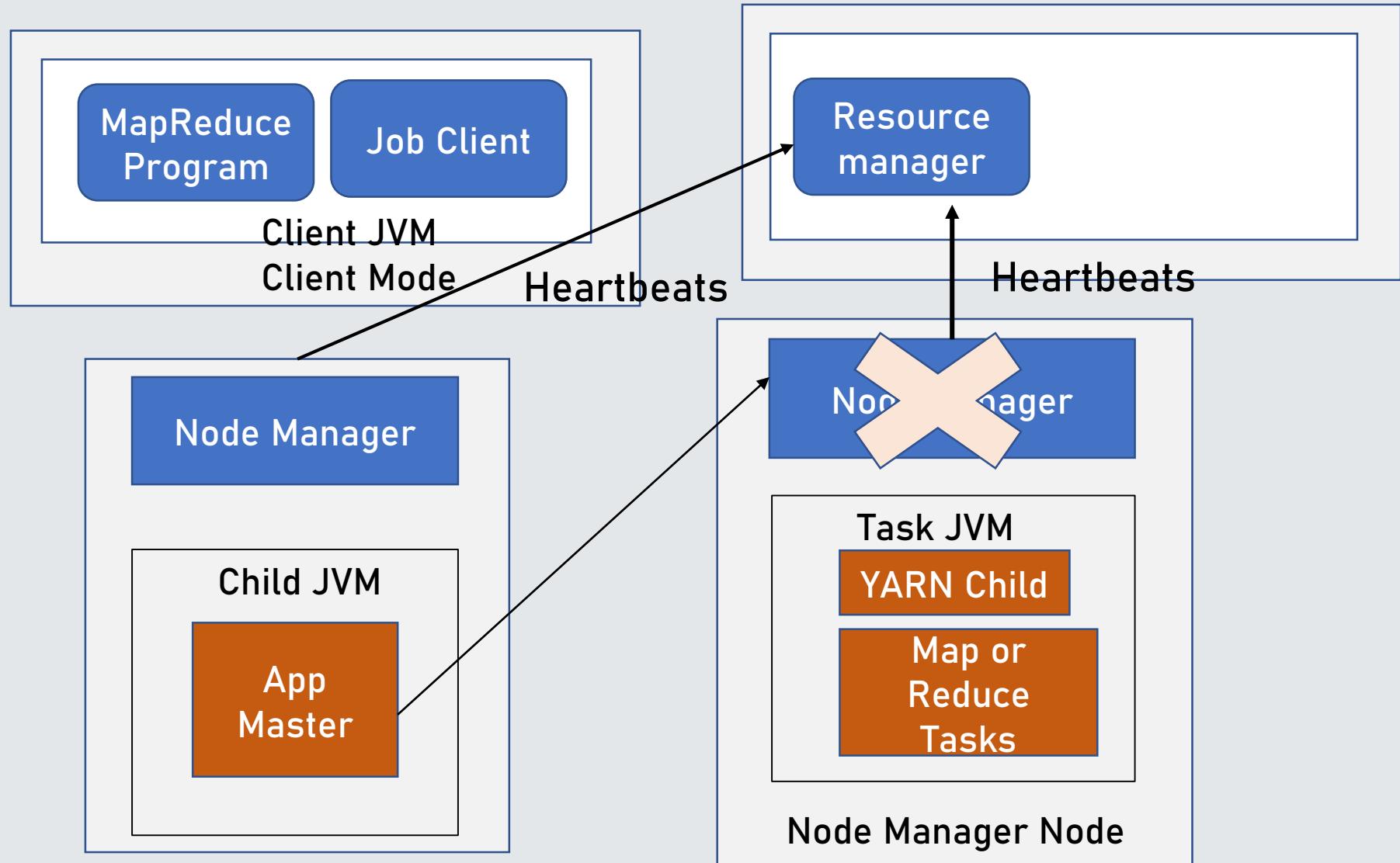
Failure Scenarios in YARN



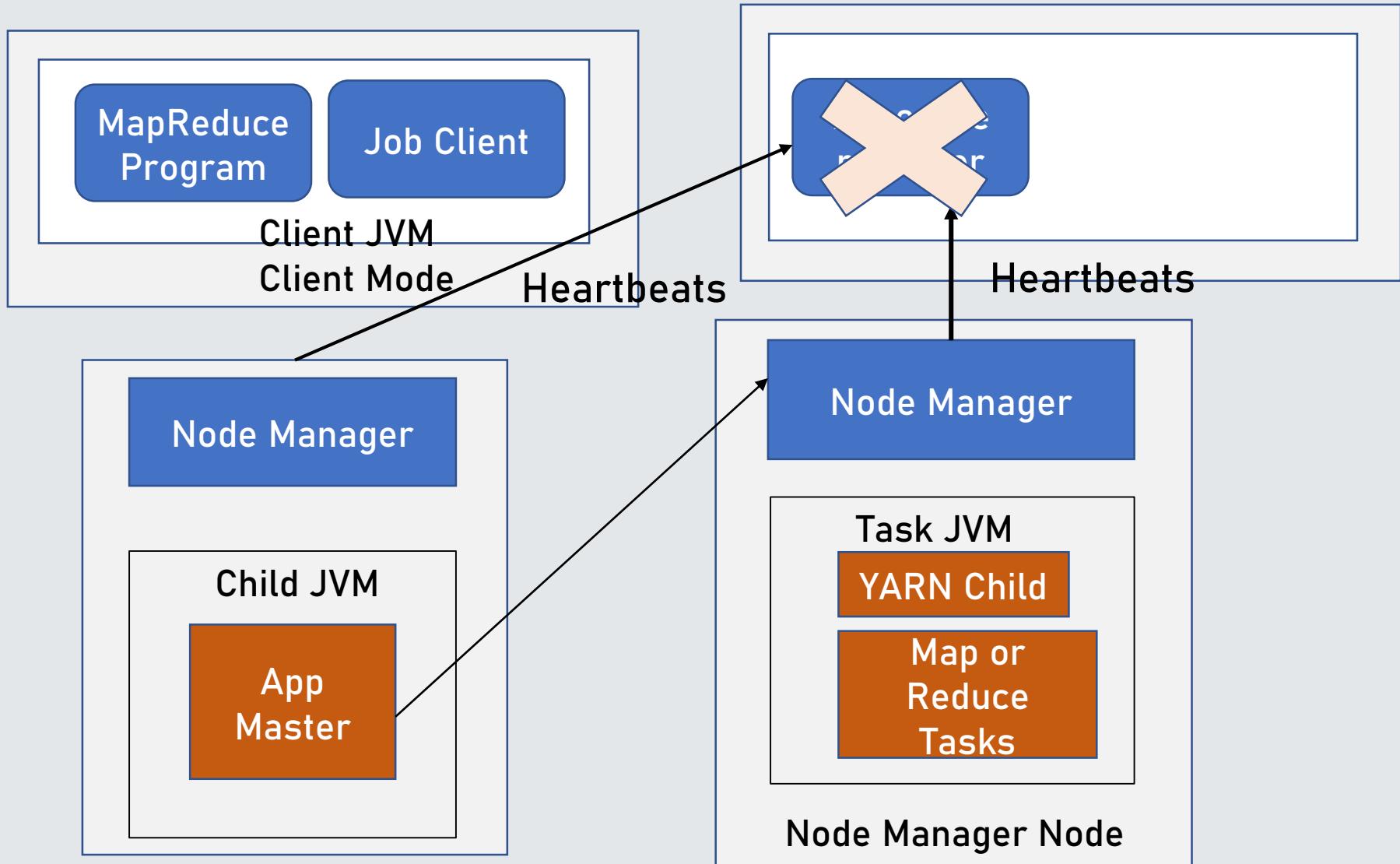
Failure Scenarios in YARN



Failure Scenarios in YARN

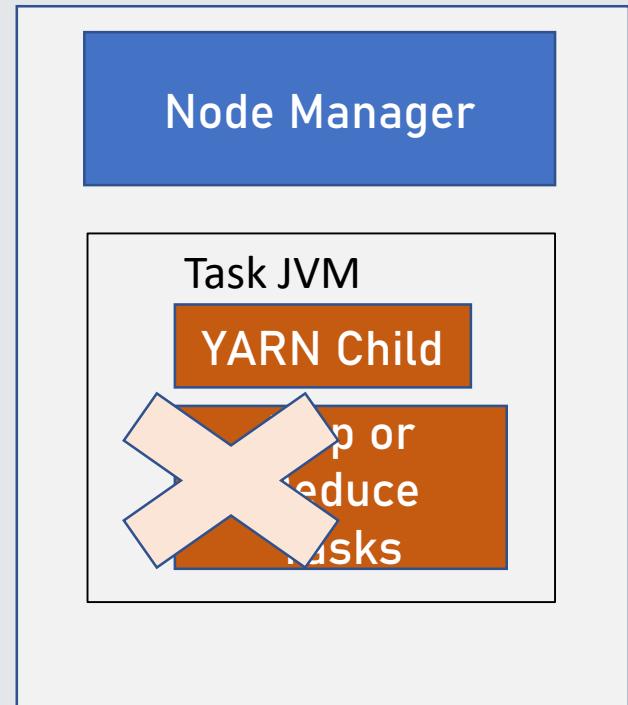


Failure Scenarios in YARN



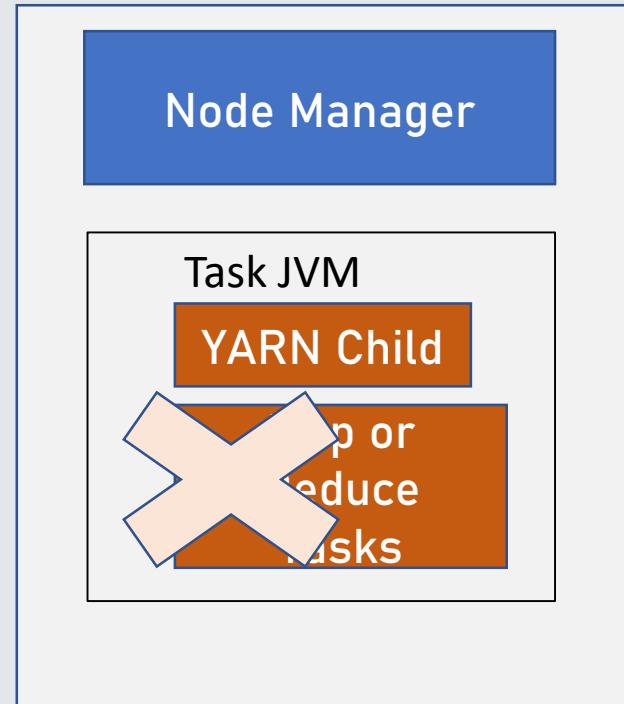
Task failure

- Runtime exceptions and sudden exits of the JVM are propagated



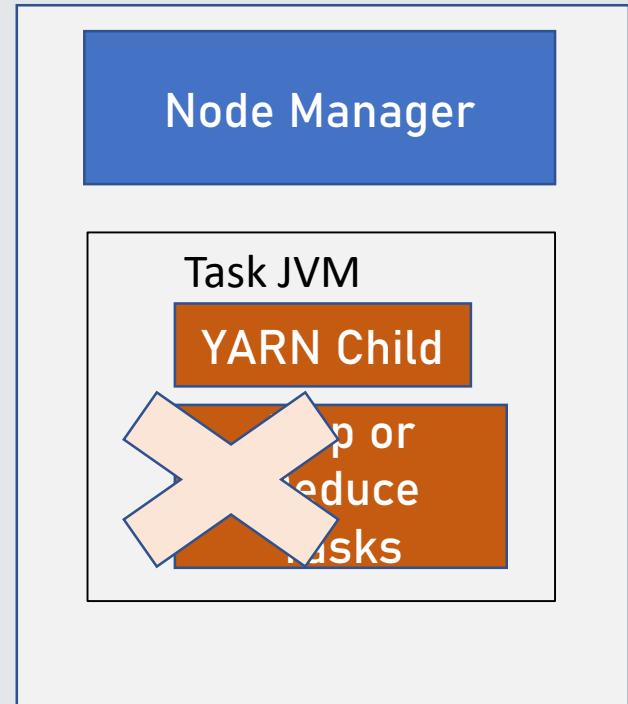
Task failure

- Runtime exceptions and sudden exits of the JVM are propagated
- Hanging tasks are noticed by the application master



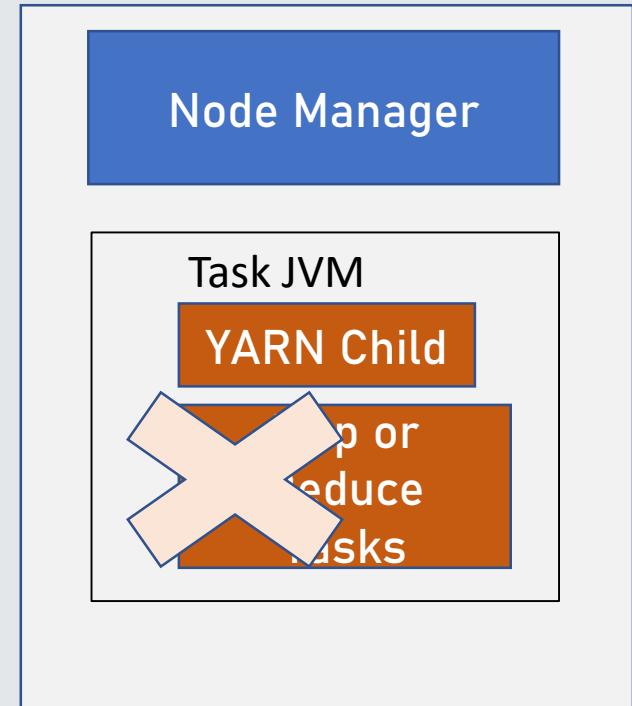
Task failure

- Runtime exceptions and sudden exits of the JVM are propagated
- Hanging tasks are noticed by the application master
- mapreduce.task.time out



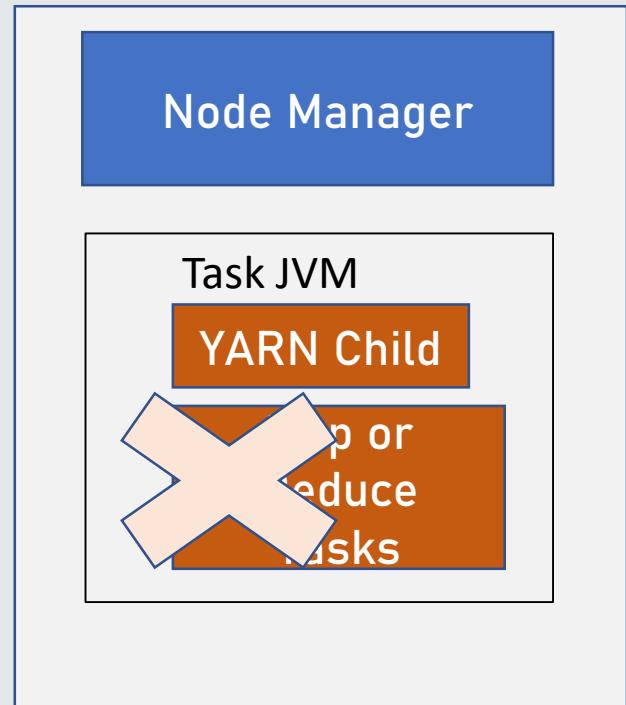
Task failure

- Runtime exceptions and sudden exits of the JVM are propagated
- Hanging tasks are noticed by the application master
- mapreduce.task.time.out
- The configuration properties for determining when a task is considered to be failed are the same as the classic case.



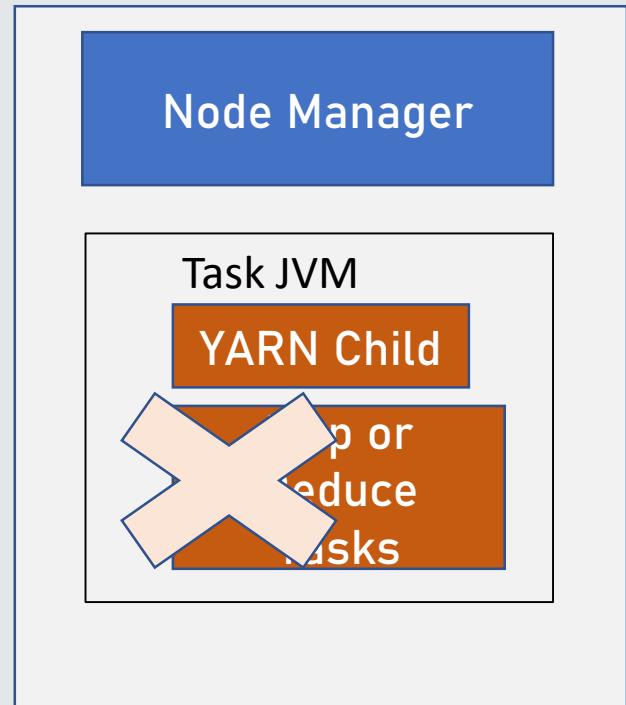
Task failure

- `mapreduce.map.maxattempts`



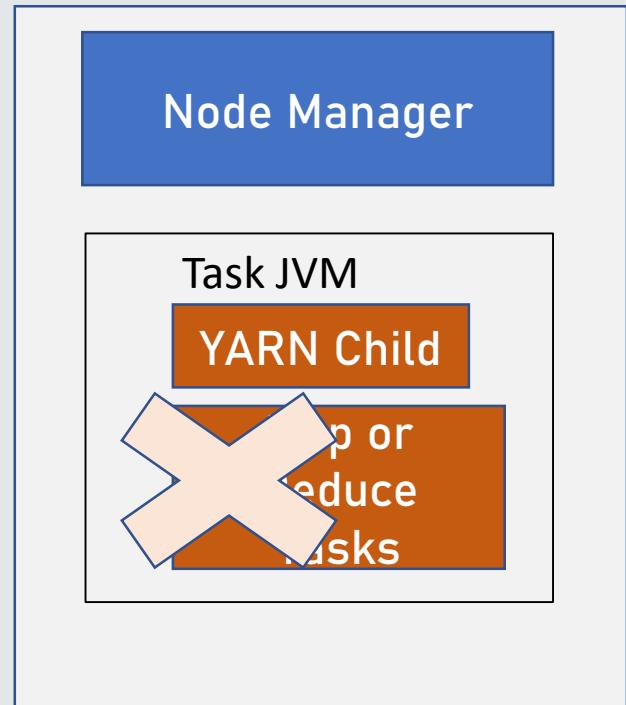
Task failure

- `mapreduce.map.maxattempts`
- `mapreduce.reduce.maxattempts.`



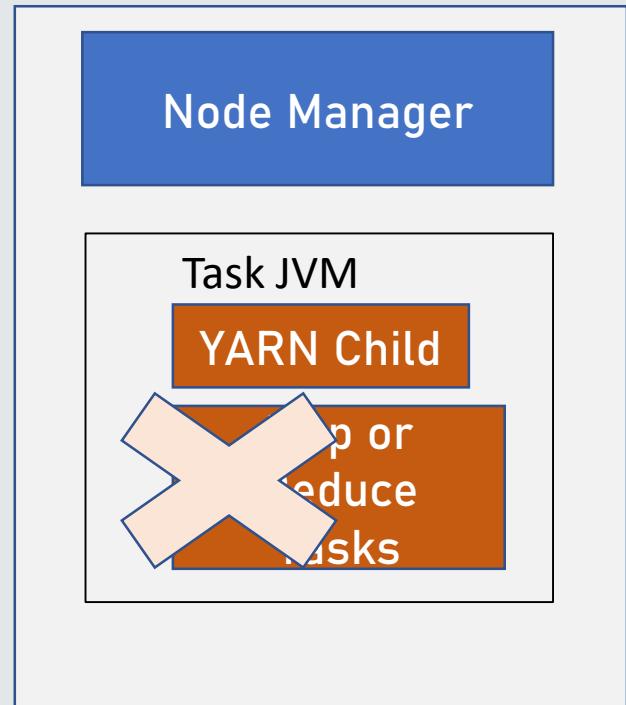
Task failure

- `mapreduce.map.maxattempts`
- `mapreduce.reduce.maxattempts.`
- `mapreduce.map.failures.maxpercent`



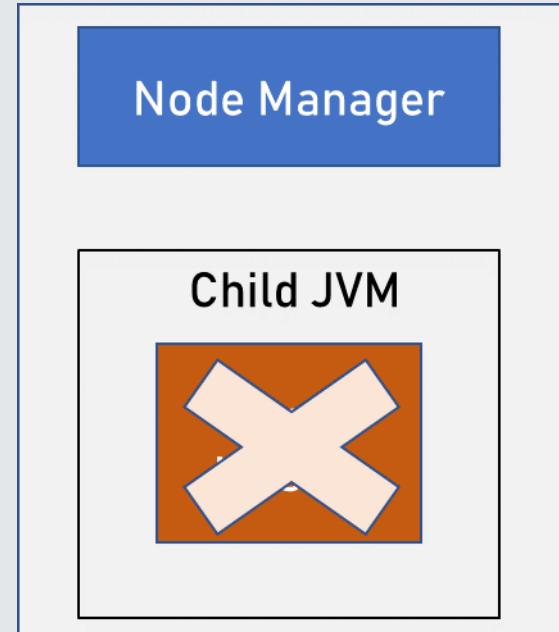
Task failure

- `mapreduce.map.maxattempts`
- `mapreduce.reduce.maxattempts.`
- `mapreduce.map.failures.maxpercent`
- `mapreduce.reduce.failures.maxpercent`



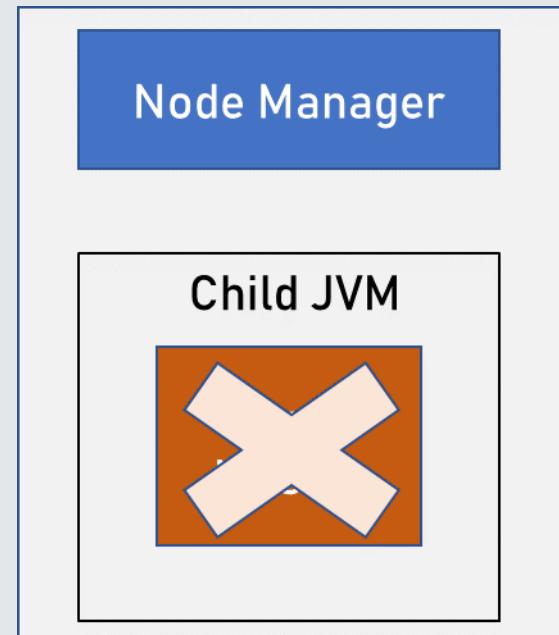
Application Master Failure

- Applications in YARN are tried multiple times in the event of failure.
- `yarn.resourcemanager.am.max-retries`
- An application master sends periodic heartbeats to the resource manager



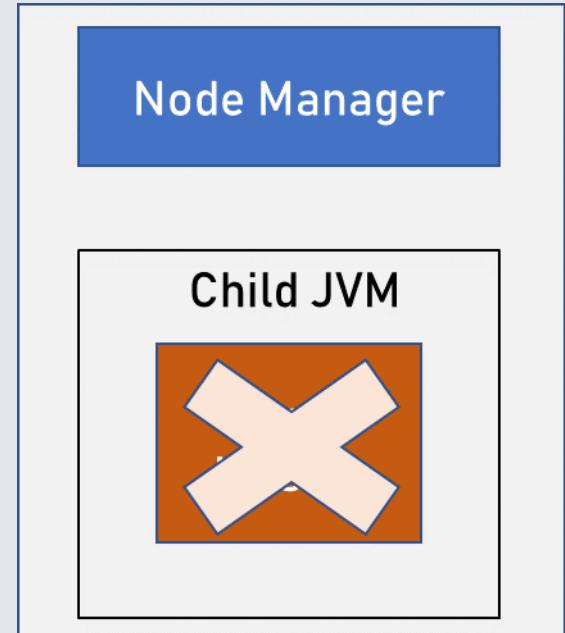
Application Master Failure

- It can recover the state of the tasks that had already been run by the (failed) application.
- `yarn.app.mapreduce.am.job.recovery.enable`



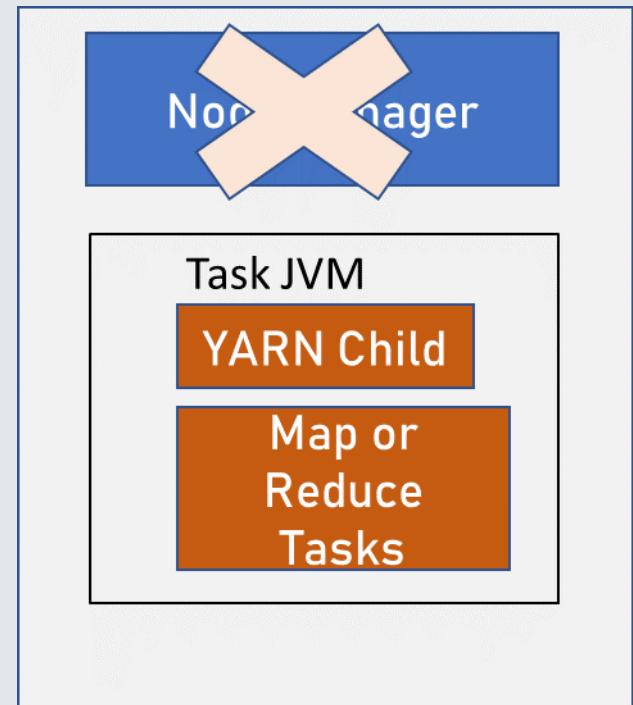
Application Master Failure

- The client polls the application master for progress reports
- it doesn't overload the resource manager with a request every time it needs to poll the application master.
- If the application master fails, however, the client will experience a timeout when it issues a status update



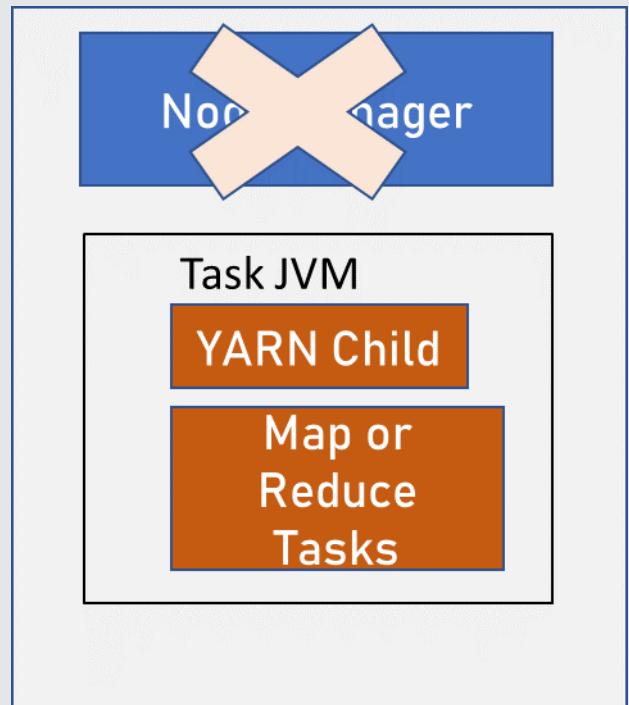
Node Manager Failure

- If a node manager fails, then it will stop sending heartbeats to the resource manager
- `yarn.resourcemanager.nm.liveness-monitor.expiry-intervals`
- Node managers may be blacklisted if the number of failures for the application is high.



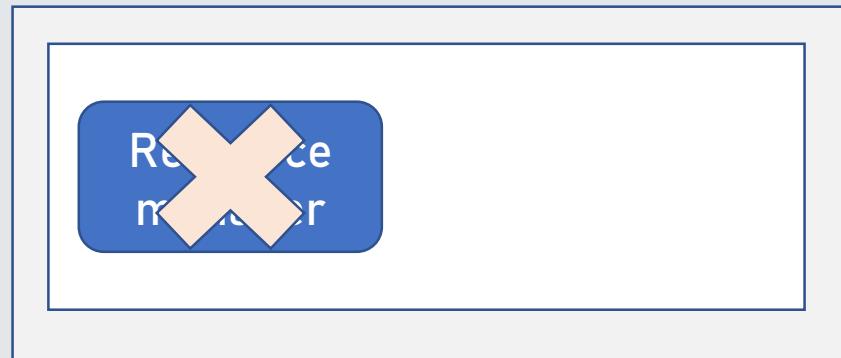
Node Manager Failure

- Blacklisting is done by the application master
- mapreduce.job.maxtaskfailures.per.tracker.



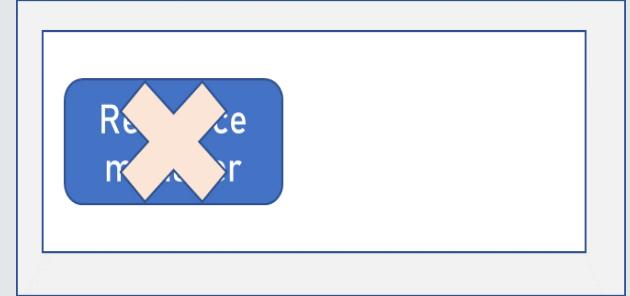
Resource Manager Failure

Failure of the resource manager is serious



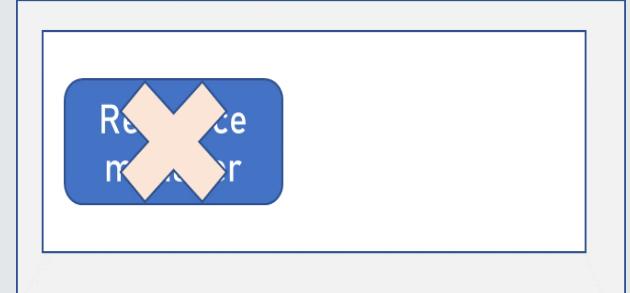
Resource Manager Failure

- After a crash, a new resource manager instance is brought up (by an administrator) and it recovers from the saved state.
- The state consists of the node managers in the system as well as the running applications.



Resource Manager Failure

- `yarn.resourcemanager.store.class`
- `org.apache.hadoop.yarn.server.resource.manager.recovery.MemStore`





That's all for now...