

NextGenMapReduceDevTesting

This wiki tracks developer-testing for [NextGenMapReduce](#).

This aim of this document is to capture various failure handling scenarios for MapReduce applications running under YARN and the YARN framework itself.

Failure scenarios

User task error

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
RM is immediately notified of error by NM with appropriate error code/status-msg		
CapacityScheduler releases resources for queue, user and application		
RM notifies AM about status (including error code) of the container		
AM fails the task attempt		
AM re-runs task-attempt before other 'virgin' tasks on a <i>different node</i>		

User task error, same task fails 4 times

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
RM is immediately notified of error by NM with appropriate error code/status-msg		
CapacityScheduler releases resources for queue, user and application		
RM notifies AM about status (including error code) of the container		
AM fails the task attempt		
AM re-runs task-attempt before other 'virgin' tasks on a <i>different node</i>		
AM fails the MapReduce job and exits		

Container failure

Localization error

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
RM is immediately notified of error by NM with appropriate error code/status-msg		
CapacityScheduler releases resources for queue, user and application		
RM notifies AM about status (including error code) of the container		
AM fails the task attempt		
AM re-runs task-attempt before other 'virgin' tasks on a <i>different node</i>		

Exceeding memory or disk limits

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
RM is immediately notified of error by NM with appropriate error code/status-msg		
CapacityScheduler releases resources for queue, user and application		
RM notifies AM about status (including error code) of the container		
AM fails the task attempt		
AM re-runs task-attempt before other 'virgin' tasks on a <i>different node</i>		

Lost map output or faulty NM Netty

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
Reducers report shuffle failure errors to AM		
On sufficient fetch-failure notifications the AM re-runs map		

User fails/kills map or reduce task

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
RM is immediately notified of error by NM with appropriate error code/status-msg		
CapacityScheduler releases resources for queue, user and application		
RM notifies AM about status (including error code) of the container		
AM fails the task attempt		
AM re-runs task-attempt before other 'virgin' tasks on a <i>different node</i>		

Node failure due to timeout or health-check error

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
RM fails all running containers and informs appropriate AMs		
Shuffle failures for completed map containers... handled (aggressively?) by AM		
AM re-runs running task-attempts and completed maps		

MapReduce AM failure

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
NM notifies RM		
CapacityScheduler releases resources for queue, user and application		
ASM recognises AM failure		
ASM kills all running containers		
ASM restarts MapReduce AM		
MapReduce AM recovers and re-runs only non-complete tasks		

ResourceManager bounce

Corrective measures	Developer(s) verifying the corrective measures	Date(s)
RM recovers all running AMs		
RM recovers all running containers		
RM rebuilds CapacityScheduler queue & user capacities		
MapReduce AMs re-runs only non-complete tasks		