

Category	Feature List	YARN CS	YARN FS	Comment	FS Property	CS Property	Property mapping comment
Resource Planning / Sharing	Hierarchical Queues	TRUE	TRUE	CS queues are not truly hierarchical			
	Elastic Capacity for better resource sharing	TRUE	TRUE				
	Absolute Resource Configuration in Queues	TRUE	TRUE				
	Percentage Based Resource Configuration in Queues	TRUE	TRUE				
Queue Management	Auto Queue Creation	TRUE	TRUE	Non-existing queues will be auto-created for users FS: Placement rules, we can also pre-create the queues based on FS config CS: flag we can turn on	yarn.scheduler.fair.allow-undeclared-pools	yarn.scheduler.capacity.<queue-path>.auto-create-child-queue.enabled	Test in combo with user as default. Need to also look at no placement rules are in place: provided queue usage
	User Mapping (user/group to queue mapping)	TRUE	TRUE	FS can only dynamically create queues for users	yarn.scheduler.fair.user-as-default-queue	yarn.scheduler.capacity.queue-mappings = u:%user:%user	
	CLI/REST api support to manage queues	TRUE	TRUE	FS does not have REST API to manage queues			
	Move applications between queues	TRUE	TRUE	Admin CLI has a generic move command for all Schedulers			
	Queue Placement Rules	TRUE	TRUE		queuePlacementPolicy 1. Rules: specified, user, primaryGroup, secondaryGroupExistingQueue, nestedUserQueue, default, reject 2. Create Option	1. yarn.scheduler.capacity.queue-mappings = u:user1.queue1.g:group1.queue2.u:%user:%user,u: user2:%primary_group Specific User = u:user1.queue1 Any User = u:%user:%user Primary Group = u:user2:%primary_group NestedUserQueue = u:%user:parent1.%user 2. yarn.scheduler.capacity.queue-mappings-override.enable 3. Auto Create Child Queue Enabled = yarn.scheduler.capacity.<queue-path>.auto-create-child-queue.enabled	1. Create Option per rule level is missing in CS 2. Secondary Group Rule is missing in CS
Queue Management	Dynamic Queue creation/deletion/modification	TRUE	TRUE				
Reservation	Reservation support in Queues	TRUE	TRUE		reservation	yarn.resourcemanager.reservation-system.enable (yarn-site.xml) yarn.scheduler.capacity.<queue-path>.reservable (capacity-scheduler.xml)	
					yarn.scheduler.fair.reservable-nodes	Not Available. Reservation Logic is completely different and CS has different set of configs.	
					yarn.scheduler.reservation-threshold.increment-multiple	Not Available. Reservation Logic is completely different and CS has different set of configs.	
Security	Authorization control (user acl's in Queues for submit/manage/admin etc)	TRUE	TRUE		aclSubmitApps, aclAdministerApps	yarn.scheduler.capacity.root.<queue-path>.acl_submit_applications yarn.scheduler.capacity.root.<queue-path>.acl_administer_queue	
	Third party ACL control (ranger)	TRUE	TRUE				
	Third party ACL control (sentry) - for Hive and Impala	FALSE	TRUE				
Node Labels	Node Labels support	TRUE	FALSE				
Rich Placement support	Hive & Sentry placement integration	FALSE	TRUE				
	Node Attributes support	TRUE	FALSE				
	Placement constraints support (minimal)	TRUE	FALSE		maxChildResources	yarn.scheduler.capacity.<queue-path>.leaf-queue-template.capacity yarn.scheduler.capacity.<queue-path>.leaf-queue-template.maximum-capacity	
	User limit quota management	TRUE	TRUE	FS: user mapping to queue, queue can have limit only FS can only limit max number of applications for specific user FS can also limit the resources for the dynamically created queues (CS can't do this) FS config: maxChildResources	<user name="sample_user"> <maxRunningApps>30</maxRunningApps> </user>	yarn.scheduler.capacity.<queue-path>.minimum-user-limit-percent, yarn.scheduler.capacity.<queue-path>.user-limit-factor yarn.scheduler.capacity.<queue-path>.user-settings.<user-name>.weight	CS does not have User Level Restriction For Max Running Apps
	AM resource quota management	TRUE	TRUE	FS config: queueMaxAMShareDefault CS config: maximum-am-resource-percent	Default Value for a queue -> queueMaxAMShareDefault Queue Level -> maxAMShare	System Level -> yarn.scheduler.capacity.maximum-am-resource-percent Queue Level -> yarn.scheduler.capacity.<queue-path>.maximum-am-resource-percent	

Scheduler & Queue	Max applicatons limit	TRUE	TRUE	Behaviour is different: hardlimit for FS flexibel limit using AM share in CS	Queue and User Level -> maxRunningApps Default Value for a queue -> queueMaxAppsDefault Default Value for a user -> userMaxAppsDefault	System Level -> yarn.scheduler.capacity.maximum-applications Queue Level -> yarn.scheduler.capacity.<queue-path>.maximum-applications	
	Queue Priority	TRUE	FALSE	FS: queue weight We should figure out how to convert weight to priority	weight	1.Queue Priority + Queue Ordering Policy: yarn.scheduler.capacity.root.<leaf-queue-path>.default-application-priority + yarn.scheduler.capacity.<queue-path>.ordering-policy (priority-utilization)	Weight for a queue in CS missing. https://blog.cloudera.com/untangling-apache-hadoop-yarn-part-3-scheduler-concepts/
					yarn.scheduler.fair.sizebasedweight	yarn.scheduler.capacity.<queue-path>.ordering-policy.fair.enable-size-based-weight	Closely linked to the queue weight used by a number of customers
	Maximum and Minimum allocation limit per container unit	TRUE	TRUE	Minimum / Maximum size of a container	minResources	yarn.scheduler.capacity.<queue-path>.capacity	
					maxResources / queueMaxResourcesDefault	yarn.scheduler.capacity.<queue-path>.maximum-capacity	
					maxContainerAllocation ("X mb, Y vcores")	yarn.scheduler.capacity.<queue-path>.maximum-allocation-mb yarn.scheduler.capacity.<queue-path>.maximum-allocation-vcores	
Round ResourceRequest	TRUE			yarn.resource-types.memory-mb.increment-allocation yarn.resource-types.vcores.increment-allocation yarn.resource-types.<resource>.increment-allocation	yarn.scheduler.minimum-allocation-mb yarn.scheduler.minimum-allocation-vcores	CS rounds the allocation size to the next multiple of minimum allocation size.	
Update Fairness Interval		TRUE	TRUE		yarn.scheduler.fair.update-interval-ms	Not Required. FS recalculate the fair shares and update demand for every update interval. This is specific to FS, CS does not require this.	
"Assign Multiple Containers per NM Heartbeat"	Enable	TRUE	TRUE		yarn.scheduler.fair.assignmultiple	yarn.scheduler.capacity.per-node-heartbeat.multiple-assignments-enabled	
	Number of Containers	TRUE	TRUE		yarn.scheduler.fair.max.assign	yarn.scheduler.capacity.per-node-heartbeat.maximum-container-assignments	
	Dynamically determine resources	TRUE	TRUE		yarn.scheduler.fair.dynamic.max.assign	NA	FS Dynamically determines the number of assignments based on Available Node Resources and assign containers which occupies 50% of Available Node resources in a heartbeat.
Core Scheduler	Node Locality	TRUE	TRUE		yarn.scheduler.fair.locality.threshold.node	yarn.scheduler.capacity.node-locality-delay	Normal scheduling: delay based on node heartbeats
					yarn.scheduler.fair.locality.threshold.rack	yarn.scheduler.capacity.rack-locality-additional-delay	Normal scheduling: delay based on node heartbeats
	Locality Delay control	TRUE	TRUE		yarn.scheduler.fair.locality-delay-node-ms	Not Required. Continous Scheduling is deprecated in FS.	Delay for continuous scheduling only
					yarn.scheduler.fair.locality-delay-rack-ms	Not Required. Continous Scheduling is deprecated in FS.	Delay for continuous scheduling only
	Asynchronous scheduling support	TRUE	TRUE	Comparable to FS Continuous scheduling Since FS cont. scheduling is deprecated, we don't need to spend much time on this			
	Multi Node scheduling compare to continuous scheduling	TRUE	FALSE		yarn.scheduler.fair.continuous-scheduling-enabled	yarn.scheduler.capacity.schedule-asynchronously	
					yarn.scheduler.fair.continuous-scheduling-sleep-ms	yarn.scheduler.capacity.schedule-asynchronously.scheduling-interval-ms	
	Multiple resource types support (CPU, Memory, GPU etc)	TRUE	TRUE				
	Queue Ordering Policies (Fair, Fifo etc)	TRUE	TRUE	Is Fair policy comparable in FS to CS?	Queue Level -> schedulingPolicy = "fifo"/"fair"/"drf" Default -> defaultQueueSchedulingPolicy	1. App Ordering Policy for Leaf Queues: yarn.scheduler.capacity.<queue-path>.ordering-policy = fifo and fair 2. DRF: yarn.scheduler.capacity.resource-calculator = org.apache.hadoop.yarn.util.resource.DominantResourceCalculator	
	Size Based Fairness deep dived	FALSE	TRUE	FS: yarn.scheduler.fair.sizebasedweight TODO: Is there a similar config in CS?			
Multiple container assignments per heartbeat	TRUE	TRUE	FS config: yarn.scheduler.fair.assignmultiple / yarn.scheduler.fair.dynamic.max.assign CS config: yarn.scheduler.capacity.per-node-heartbeat.maximum-container-assignments CS config is different as it limits container assignments per heartbeat but can't do such a thing as FS: taking up 50% of all cluster resources				

Preemption	Enable Preemption	TRUE	TRUE		yarn.scheduler.fair.preemption	yarn.resourcemanager.scheduler.monitor.enable	
	Disable Preemption at Queue Level	TRUE	TRUE		allowPreemptionFrom	yarn.scheduler.capacity.<queue-path>.disable_preemption	
	Wait Time before Kill of containers	TRUE	TRUE		yarn.scheduler.fair.waitTimeBeforeKill	yarn.resourcemanager.monitor.capacity.preemption.max_wait_before_kill	
	Wait Time before Next Starvation Check	TRUE	TRUE		yarn.scheduler.fair.waitTimeBeforeNextStarvationCheck	yarn.resourcemanager.monitor.capacity.preemption.monitoring_interval	FS: Next Starvation Check for an Application to check if preemption is required and to make room for outstanding demand. CS: Interval at which the Preemption Policy will be triggered.
	Preemption Wait	TRUE	TRUE		minSharePreemptionTimeout / defaultMinSharePreemptionTimeout fairSharePreemptionTimeout / defaultFairSharePreemptionTimeout fairSharePreemptionThreshold / defaultFairSharePreemptionThreshold	Not Required. CS immediately tries to serve the starving queue. There is no wait period	
	Preemption Threshold	TRUE	TRUE		yarn.scheduler.fair.preemption.cluster-utilization.threshold	Not Required. CS Preemption Logic will kick in only when there is a starved queue, and there is no available cluster resource to fit the request. And there is a queue which is using beyond it's capacity.	
	Inter Queue preemption support	TRUE	TRUE				
	Intra Queue preemption support	TRUE	TRUE				
	Reservation based preemption	TRUE	TRUE				
	Queue Priority based preemption	TRUE	FALSE				
Application Management	First class Concept of application	TRUE	TRUE				
	Application priority	TRUE	FALSE				
	Application timeout	TRUE	TRUE				
	Moving Application across queues	TRUE	TRUE	Same as Move applications between queues			
High Availability	Stateful	TRUE	TRUE				
	yarn.resourcemanager.scheduler.class	TRUE	TRUE		org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler	org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler	
	Scheduler Config File Location	TRUE	TRUE		yarn.scheduler.fair.allocation.file	1. yarn.scheduler.configuration.store.class 2. yarn.resourcemanager.configuration.provider-class	