

name value description mapreduce.job.hdfs-servers `${fs.defaultFS}` mapreduce.job.committer.setup.cleanup.needed true true, if job needs job-setup and job-cleanup. false, otherwise mapreduce.task.io.sort.factor 10 The number of streams to merge at once while sorting files. This determines the number of open file handles. mapreduce.task.io.sort.mb 100 The total amount of buffer memory to use while sorting files, in megabytes. By default, gives each merge stream 1MB, which should minimize seeks. mapreduce.map.sort.spill.percent 0.80 The soft limit in the serialization buffer. Once reached, a thread will begin to spill the contents to disk in the background. Note that collection will not block if this threshold is exceeded while a spill is already in progress, so spills may be larger than this threshold when it is set to less than .5 mapreduce.job.maps 2 The default number of map tasks per job. Ignored when mapreduce.framework.name is "local". mapreduce.job.reduces 1 The default number of reduce tasks per job. Typically set to 99% of the cluster's reduce capacity, so that if a node fails the reduces can still be executed in a single wave. Ignored when mapreduce.framework.name is "local". mapreduce.job.running.map.limit 0 The maximum number of simultaneous map tasks per job. There is no limit if this value is 0 or negative. mapreduce.job.running.reduce.limit 0 The maximum number of simultaneous reduce tasks per job. There is no limit if this value is 0 or negative. mapreduce.job.max.map -1 Limit on the number of map tasks allowed per job. There is no limit if this value is negative. mapreduce.job.reducer.preempt.delay.sec 0 The threshold (in seconds) after which an unsatisfied mapper request triggers reducer preemption when there is no anticipated headroom. If set to 0 or a negative value, the reducer is preempted as soon as lack of headroom is detected. Default is 0. mapreduce.job.reducer.unconditional-preempt.delay.sec 300 The threshold (in seconds) after which an unsatisfied mapper request triggers a forced reducer preemption irrespective of the anticipated headroom. By default, it is set to 5 mins. Setting it to 0 leads to immediate reducer preemption. Setting to -1 disables this preemption altogether. mapreduce.job.max.split.locations 10 The maximum number of block locations to store for each split for locality calculation. mapreduce.job.split.metainfo.maxsize 10000000 The maximum permissible size of the split metainfo file. The MapReduce ApplicationMaster won't attempt to read submitted split metainfo files bigger than this configured value. No limits if set to -1. mapreduce.map.maxattempts 4 Expert: The maximum number of attempts per map task. In other words, framework will try to execute a map task these many number of times before giving up on it. mapreduce.reduce.maxattempts 4 Expert: The maximum number of attempts per reduce task. In other words, framework will try to execute a reduce task these many number of times before giving up on it. mapreduce.reduce.shuffle.fetch.retry.enabled `${yarn.nodemanager.recovery.enabled}` Set to enable fetch retry during host restart. mapreduce.reduce.shuffle.fetch.retry.interval-ms 1000 Time of interval that fetcher retry to fetch again when some non-fatal failure happens because of some events like NM restart. mapreduce.reduce.shuffle.fetch.retry.timeout-ms 30000 Timeout value for fetcher to retry to fetch again when some non-fatal failure happens because of some events like NM restart. mapreduce.reduce.shuffle.retry-delay.max.ms 60000 The maximum number of ms the reducer will delay before retrying to download map data. mapreduce.reduce.shuffle.parallelcopies 5 The default number of parallel transfers run by reduce during the copy(shuffle) phase. mapreduce.reduce.shuffle.connect.timeout 180000 Expert: The maximum amount of time (in milli seconds) reduce task spends in trying to connect to a remote node for getting map output. mapreduce.reduce.shuffle.read.timeout 180000 Expert: The maximum amount of time (in milli seconds) reduce task waits for map output data to be available for reading after obtaining connection. mapreduce.shuffle.listen.queue.size 128 The length of the shuffle server listen queue. mapreduce.shuffle.connection-keep-alive.enable false set to true to support keep-alive connections. mapreduce.shuffle.connection-keep-alive.timeout 5 The number of seconds a shuffle client attempts to retain http connection. Refer "Keep-Alive: timeout=" header in Http specification mapreduce.task.timeout 600000 The number of milliseconds before a task will be terminated if it neither reads an input, writes an output, nor updates its status string. A value of 0 disables the timeout. mapreduce.map.memory.mb -1 The amount of memory to request from the scheduler for each map task. If this is not specified or is non-positive, it is inferred from mapreduce.map.java.opts and mapreduce.job.heap.memory-mb.ratio. If java-opts are also not specified, we set it to 1024. mapreduce.map.cpu.vcores 1 The number of virtual cores to request from the scheduler for each map task. mapreduce.reduce.memory.mb -1 The amount of memory to request from the scheduler for each reduce task. If this is not specified or is non-positive, it is inferred from mapreduce.reduce.java.opts and mapreduce.job.heap.memory-mb.ratio. If java-opts are also not specified, we set it to 1024. mapreduce.reduce.cpu.vcores 1 The number of virtual cores to request from the scheduler for each reduce task. mapred.child.java.opts Java opts for the task processes. The following symbol, if present, will be interpolated: @taskid@ is replaced by current TaskID. Any other occurrences of '@' will go unchanged. For example, to enable verbose gc logging to a file named for the taskid in /tmp and to set the heap maximum to be a gigabyte, pass a 'value' of: -Xmx1024m -verbose:gc -Xloggc:/tmp/@taskid@.gc Usage of -Djava.library.path can cause programs to no longer function if hadoop native libraries are used. These values should instead be set as part of LD_LIBRARY_PATH in the map / reduce JVM env using the mapreduce.map.env and mapreduce.reduce.env config settings. If -Xmx is not set, it is inferred from mapreduce.{map|reduce}.memory.mb and mapreduce.job.heap.memory-mb.ratio. mapred.child.env User added environment variables for the task processes. Example : 1) A=foo This will set the env variable A to foo 2) B=\$B:c This is inherit nodemanager's B env variable on Unix. 3) B=%B%;c This is inherit nodemanager's B env variable on Windows. mapreduce.admin.user.env Expert: Additional execution environment entries for map and reduce task processes. This is not an additive property. You must preserve the original value if you want your map and reduce tasks to have access to native libraries (compression, etc). When this value is empty, the command to set execution environment will be OS dependent: For linux, use LD_LIBRARY_PATH=\$HADOOP_COMMON_HOME/lib/native. For windows, use PATH = %PATH%;%HADOOP_COMMON_HOME%\bin. yarn.app.mapreduce.am.log.level INFO The logging level for the MR ApplicationMaster. The allowed levels are: OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE and ALL. The setting here could be overridden if "mapreduce.job.log4j-properties-file" is set. mapreduce.map.log.level INFO The logging level for the map task. The allowed levels are: OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE and ALL. The setting here could be overridden if "mapreduce.job.log4j-properties-file" is set. mapreduce.reduce.log.level INFO The logging level for the reduce task. The allowed levels are: OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE and ALL. The setting here could be overridden if "mapreduce.job.log4j-properties-file" is set. mapreduce.map.cpu.vcores 1 The number of virtual cores required for each map task. mapreduce.reduce.cpu.vcores 1 The number of virtual cores required for each reduce task. mapreduce.reduce.merge.inmem.threshold 1000 The threshold, in terms of the number of files for the in-memory merge process. When we accumulate threshold number of files we initiate the in-memory merge and spill to disk. A value of 0 or less than 0 indicates we want to DON'T have any threshold and instead depend only on the ramfs's memory consumption to trigger the merge. mapreduce.reduce.shuffle.merge.percent 0.66 The usage threshold at which an in-memory merge will be initiated, expressed as a percentage of the total memory allocated to storing in-memory map outputs, as defined by mapreduce.reduce.shuffle.input.buffer.percent. mapreduce.reduce.shuffle.input.buffer.percent 0.70 The percentage of memory to be allocated from the maximum heap size to storing map outputs during the shuffle. mapreduce.reduce.input.buffer.percent 0.0 The percentage of memory- relative to the maximum heap size- to retain map outputs during the reduce. When the shuffle is concluded, any remaining map outputs in memory must consume less than this threshold before the reduce can begin.

mapreduce.reduce.shuffle.memory.limit.percent 0.25 Expert: Maximum percentage of the in-memory limit that a single shuffle can consume. Range of valid values is [0.0, 1.0]. If the value is 0.0 map outputs are shuffled directly to disk. mapreduce.shuffle.ssl.enabled false Whether to use SSL for for the Shuffle HTTP endpoints. mapreduce.shuffle.ssl.file.buffer.size 65536 Buffer size for reading spills from file when using SSL. mapreduce.shuffle.max.connections 0 Max allowed connections for the shuffle. Set to 0 (zero) to indicate no limit on the number of connections. mapreduce.shuffle.max.threads 0 Max allowed threads for serving shuffle connections. Set to zero to indicate the default of 2 times the number of available processors (as reported by Runtime.availableProcessors()). Netty is used to serve requests, so a thread is not needed for each connection. mapreduce.shuffle.transferTo.allowed This option can enable/disable using nio transferTo method in the shuffle phase. NIO transferTo does not perform well on windows in the shuffle phase. Thus, with this configuration property it is possible to disable it, in which case custom transfer method will be used. Recommended value is false when running Hadoop on Windows. For Linux, it is recommended to set it to true. If nothing is set then the default value is false for Windows, and true for Linux. mapreduce.shuffle.transfer.buffer.size 131072 This property is used only if mapreduce.shuffle.transferTo.allowed is set to false. In that case, this property defines the size of the buffer used in the buffer copy code for the shuffle phase. The size of this buffer determines the size of the IO requests. mapreduce.reduce.markreset.buffer.percent 0.0 The percentage of memory -relative to the maximum heap size- to be used for caching values when using the mark-reset functionality. mapreduce.map.speculative true If true, then multiple instances of some map tasks may be executed in parallel. mapreduce.reduce.speculative true If true, then multiple instances of some reduce tasks may be executed in parallel. mapreduce.job.speculative.speculative-cap-running-tasks 0.1 The max percent (0-1) of running tasks that can be speculatively re-executed at any time. mapreduce.job.speculative.speculative-cap-total-tasks 0.01 The max percent (0-1) of all tasks that can be speculatively re-executed at any time. mapreduce.job.speculative.minimum-allowed-tasks 10 The minimum allowed tasks that can be speculatively re-executed at any time. mapreduce.job.speculative.retry-after-no-speculate 1000 The waiting time(ms) to do next round of speculation if there is no task speculated in this round. mapreduce.job.speculative.retry-after-speculate 15000 The waiting time(ms) to do next round of speculation if there are tasks speculated in this round. mapreduce.job.map.output.collector.class org.apache.hadoop.mapred.MapTask\$MapOutputBuffer The MapOutputCollector implementation(s) to use. This may be a comma-separated list of class names, in which case the map task will try to initialize each of the collectors in turn. The first to successfully initialize will be used. mapreduce.job.speculative.slowtaskthreshold 1.0 The number of standard deviations by which a task's ave progress-rates must be lower than the average of all running tasks' for the task to be considered too slow. mapreduce.job.ubertask.enable false Whether to enable the small-jobs "ubertask" optimization, which runs "sufficiently small" jobs sequentially within a single JVM. "Small" is defined by the following maxmaps, maxreduces, and maxbytes settings. Note that configurations for application masters also affect the "Small" definition - yarn.app.mapreduce.am.resource.mb must be larger than both mapreduce.map.memory.mb and mapreduce.reduce.memory.mb, and yarn.app.mapreduce.am.resource.cpu-vcores must be larger than both mapreduce.map.cpu.vcores and mapreduce.reduce.cpu.vcores to enable ubertask. Users may override this value. mapreduce.job.ubertask.maxmaps 9 Threshold for number of maps, beyond which job is considered too big for the ubertasking optimization. Users may override this value, but only downward. mapreduce.job.ubertask.maxreduces 1 Threshold for number of reduces, beyond which job is considered too big for the ubertasking optimization. CURRENTLY THE CODE CANNOT SUPPORT MORE THAN ONE REDUCE and will ignore larger values. (Zero is a valid max, however.) Users may override this value, but only downward. mapreduce.job.ubertask.maxbytes Threshold for number of input bytes, beyond which job is considered too big for the ubertasking optimization. If no value is specified, dfs.block.size is used as a default. Be sure to specify a default value in mapred-site.xml if the underlying filesystem is not HDFS. Users may override this value, but only downward. mapreduce.job.emit-timeline-data false Specifies if the Application Master should emit timeline data to the timeline server. Individual jobs can override this value. mapreduce.job.sharedcache.mode disabled A comma delimited list of resource categories to submit to the shared cache. The valid categories are: jobjar, libjars, files, archives. If "disabled" is specified then the job submission code will not use the shared cache. mapreduce.input.fileinputformat.split.minsize 0 The minimum size chunk that map input should be split into. Note that some file formats may have minimum split sizes that take priority over this setting. mapreduce.input.fileinputformat.list-status.num-threads 1 The number of threads to use to list and fetch block locations for the specified input paths. Note: multiple threads should not be used if a custom non thread-safe path filter is used. mapreduce.input.lineinputformat.linespermap 1 When using NLineInputFormat, the number of lines of input data to include in each split. mapreduce.client.submit.file.replication 10 The replication level for submitted job files. This should be around the square root of the number of nodes. mapreduce.task.files.preserve.failedtasks false Should the files for failed tasks be kept. This should only be used on jobs that are failing, because the storage is never reclaimed. It also prevents the map outputs from being erased from the reduce directory as they are consumed. mapreduce.output.fileoutputformat.compress false Should the job outputs be compressed? mapreduce.output.fileoutputformat.compress.type RECORD If the job outputs are to compressed as SequenceFiles, how should they be compressed? Should be one of NONE, RECORD or BLOCK. mapreduce.output.fileoutputformat.compress.codec org.apache.hadoop.io.compress.DefaultCodec If the job outputs are compressed, how should they be compressed? mapreduce.map.output.compress false Should the outputs of the maps be compressed before being sent across the network. Uses SequenceFile compression. mapreduce.map.output.compress.codec org.apache.hadoop.io.compress.DefaultCodec If the map outputs are compressed, how should they be compressed? map.sort.class org.apache.hadoop.util.QuickSort The default sort class for sorting keys. mapreduce.task.userlog.limit.kb 0 The maximum size of user-logs of each task in KB. 0 disables the cap. yarn.app.mapreduce.am.container.log.limit.kb 0 The maximum size of the MRAppMaster attempt container logs in KB. 0 disables the cap. yarn.app.mapreduce.task.container.log.backups 0 Number of backup files for task logs when using ContainerRollingLogAppender (CRLA). See org.apache.log4j.RollingFileAppender.maxBackupIndex. By default, ContainerLogAppender (CLA) is used, and container logs are not rolled. CRLA is enabled for tasks when both mapreduce.task.userlog.limit.kb and yarn.app.mapreduce.task.container.log.backups are greater than zero. yarn.app.mapreduce.am.container.log.backups 0 Number of backup files for the ApplicationMaster logs when using ContainerRollingLogAppender (CRLA). See org.apache.log4j.RollingFileAppender.maxBackupIndex. By default, ContainerLogAppender (CLA) is used, and container logs are not rolled. CRLA is enabled for the ApplicationMaster when both yarn.app.mapreduce.am.container.log.limit.kb and yarn.app.mapreduce.am.container.log.backups are greater than zero. yarn.app.mapreduce.shuffle.log.separate true If enabled ('true') logging generated by the client-side shuffle classes in a reducer will be written in a dedicated log file 'syslog.shuffle' instead of 'syslog'. yarn.app.mapreduce.shuffle.log.limit.kb 0 Maximum size of the syslog.shuffle file in kilobytes (0 for no limit). yarn.app.mapreduce.shuffle.log.backups 0 If yarn.app.mapreduce.shuffle.log.limit.kb and yarn.app.mapreduce.shuffle.log.backups are greater than zero then a ContainerRollngLogAppender is used instead of ContainerLogAppender for syslog.shuffle. See org.apache.log4j.RollingFileAppender.maxBackupIndex mapreduce.job.maxtaskfailures.per.tracker 3 The number of task-failures on a

node manager of a given job after which new tasks of that job aren't assigned to it. It MUST be less than `mapreduce.map.maxattempts` and `mapreduce.reduce.maxattempts` otherwise the failed task will never be tried on a different node. `mapreduce.client.output.filter` FAILED The filter for controlling the output of the task's userlogs sent to the console of the JobClient. The permissible options are: NONE, KILLED, FAILED, SUCCEEDED and ALL. `mapreduce.client.completion.pollinterval` 5000 The interval (in milliseconds) between which the JobClient polls the MapReduce ApplicationMaster for updates about job status. You may want to set this to a lower value to make tests run faster on a single node system. Adjusting this value in production may lead to unwanted client-server traffic. `mapreduce.client.progressmonitor.pollinterval` 1000 The interval (in milliseconds) between which the JobClient reports status to the console and checks for job completion. You may want to set this to a lower value to make tests run faster on a single node system. Adjusting this value in production may lead to unwanted client-server traffic. `mapreduce.client.libjars.wildcard` true Whether the libjars cache files should be localized using a wildcarded directory instead of naming each archive independently. Using wildcards reduces the space needed for storing the job information in the case of a highly available resource manager configuration. This property should only be set to false for specific jobs which are highly sensitive to the details of the archive localization. Having this property set to true will cause the archives to all be localized to the same local cache location. If false, each archive will be localized to its own local cache location. In both cases a symbolic link will be created to every archive from the job's working directory. `mapreduce.task.profile` false To set whether the system should collect profiler information for some of the tasks in this job? The information is stored in the user log directory. The value is "true" if task profiling is enabled. `mapreduce.task.profile.maps` 0-2 To set the ranges of map tasks to profile. `mapreduce.task.profile` has to be set to true for the value to be accounted. `mapreduce.task.profile.reduce` 0-2 To set the ranges of reduce tasks to profile. `mapreduce.task.profile` has to be set to true for the value to be accounted. `mapreduce.task.profile.params` - `agentlib:hprof=cpu=samples,heap=sites,force=n,thread=y,verbose=n,file=%s` JVM profiler parameters used to profile map and reduce task attempts. This string may contain a single format specifier `%s` that will be replaced by the path to `profile.out` in the task attempt log directory. To specify different profiling options for map tasks and reduce tasks, more specific parameters `mapreduce.task.profile.map.params` and `mapreduce.task.profile.reduce.params` should be used. `mapreduce.task.profile.map.params` `${mapreduce.task.profile.params}` Map-task-specific JVM profiler parameters. See `mapreduce.task.profile.params` `mapreduce.task.profile.reduce.params` `${mapreduce.task.profile.params}` Reduce-task-specific JVM profiler parameters. See `mapreduce.task.profile.params` `mapreduce.task.skip.start.attempts` 2 The number of Task attempts AFTER which skip mode will be kicked off. When skip mode is kicked off, the tasks reports the range of records which it will process next, to the MR ApplicationMaster. So that on failures, the MR AM knows which ones are possibly the bad records. On further executions, those are skipped. `mapreduce.job.skip.outdir` If no value is specified here, the skipped records are written to the output directory at `_logs/skip`. User can stop writing skipped records by giving the value "none". `mapreduce.map.skip.maxrecords` 0 The number of acceptable skip records surrounding the bad record PER bad record in mapper. The number includes the bad record as well. To turn the feature of detection/skipping of bad records off, set the value to 0. The framework tries to narrow down the skipped range by retrying until this threshold is met OR all attempts get exhausted for this task. Set the value to `Long.MAX_VALUE` to indicate that framework need not try to narrow down. Whatever records(depends on application) get skipped are acceptable. `mapreduce.map.skip.proc-count.auto-incr` true The flag which if set to true, `SkipBadRecords.COUNTER_MAP_PROCESSED_RECORDS` is incremented by MapRunner after invoking the map function. This value must be set to false for applications which process the records asynchronously or buffer the input records. For example streaming. In such cases applications should increment this counter on their own. `mapreduce.reduce.skip.maxgroups` 0 The number of acceptable skip groups surrounding the bad group PER bad group in reducer. The number includes the bad group as well. To turn the feature of detection/skipping of bad groups off, set the value to 0. The framework tries to narrow down the skipped range by retrying until this threshold is met OR all attempts get exhausted for this task. Set the value to `Long.MAX_VALUE` to indicate that framework need not try to narrow down. Whatever groups(depends on application) get skipped are acceptable. `mapreduce.reduce.skip.proc-count.auto-incr` true The flag which if set to true. `SkipBadRecords.COUNTER_REDUCE_PROCESSED_GROUPS` is incremented by framework after invoking the reduce function. This value must be set to false for applications which process the records asynchronously or buffer the input records. For example streaming. In such cases applications should increment this counter on their own. `mapreduce.ifile.readahead` true Configuration key to enable/disable IFile readahead. `mapreduce.ifile.readahead.bytes` 4194304 Configuration key to set the IFile readahead length in bytes. `mapreduce.job.queue` default Queue to which a job is submitted. This must match one of the queues defined in `mapred-queues.xml` for the system. Also, the ACL setup for the queue must allow the current user to submit a job to the queue. Before specifying a queue, ensure that the system is configured with the queue, and access is allowed for submitting jobs to the queue. `mapreduce.job.tags` Tags for the job that will be passed to YARN at submission time. Queries to YARN for applications can filter on these tags. `mapreduce.cluster.local.dir` `${hadoop.tmp.dir}/mapred/local` The local directory where MapReduce stores intermediate data files. May be a comma-separated list of directories on different devices in order to spread disk i/o. Directories that do not exist are ignored. `mapreduce.cluster.acls.enabled` false Specifies whether ACLs should be checked for authorization of users for doing various queue and job level operations. ACLs are disabled by default. If enabled, access control checks are made by MapReduce ApplicationMaster when requests are made by users for queue operations like submit job to a queue and kill a job in the queue and job operations like viewing the job-details (See `mapreduce.job.acl-view-job`) or for modifying the job (See `mapreduce.job.acl-modify-job`) using Map/Reduce APIs, RPCs or via the console and web user interfaces. For enabling this flag, set to true in `mapred-site.xml` file of all MapReduce clients (MR job submitting nodes). `mapreduce.job.acl-modify-job` Job specific access-control list for 'modifying' the job. It is only used if authorization is enabled in Map/Reduce by setting the configuration property `mapreduce.cluster.acls.enabled` to true. This specifies the list of users and/or groups who can do modification operations on the job. For specifying a list of users and groups the format to use is "user1,user2 group1,group". If set to '*', it allows all users/groups to modify this job. If set to ' ' (i.e. space), it allows none. This configuration is used to guard all the modifications with respect to this job and takes care of all the following operations: o killing this job o killing a task of this job, failing a task of this job o setting the priority of this job Each of these operations are also protected by the per-queue level ACL "acl-administer-jobs" configured via `mapred-queues.xml`. So a caller should have the authorization to satisfy either the queue-level ACL or the job-level ACL. Irrespective of this ACL configuration, (a) job-owner, (b) the user who started the cluster, (c) members of an admin configured supergroup configured via `mapreduce.cluster.permissions.supergroup` and (d) queue administrators of the queue to which this job was submitted to configured via `acl-administer-jobs` for the specific queue in `mapred-queues.xml` can do all the modification operations on a job. By default, nobody else besides job-owner, the user who started the cluster, members of supergroup and queue administrators can perform modification operations on a job. `mapreduce.job.acl-view-job` Job specific access-control list for 'viewing' the job. It is only used if authorization is enabled in Map/Reduce by setting the configuration property `mapreduce.cluster.acls.enabled` to true. This specifies the list of users and/or groups who can view private details about the job. For specifying a list of users and groups the format to use is

user1,user2 group1,group". If set to '*', it allows all users/groups to modify this job. If set to ' '(i.e. space), it allows none. This configuration is used to guard some of the job-views and at present only protects APIs that can return possibly sensitive information of the job-owner like o job-level counters o task-level counters o tasks' diagnostic information o task-logs displayed on the HistoryServer's web-UI and o job.xml showed by the HistoryServer's web-UI Every other piece of information of jobs is still accessible by any other user, for e.g., JobStatus, JobProfile, list of jobs in the queue, etc. Irrespective of this ACL configuration, (a) job-owner, (b) the user who started the cluster, (c) members of an admin configured supergroup configured via mapreduce.cluster.permissions.supergroup and (d) queue administrators of the queue to which this job was submitted to configured via acl-administer-jobs for the specific queue in mapred-queues.xml can do all the view operations on a job. By default, nobody else besides job-owner, the user who started the cluster, members of supergroup and queue administrators can perform view operations on a job. mapreduce.job.finish-when-all-reducers-done true Specifies whether the job should complete once all reducers have finished, regardless of whether there are still running mappers.

mapreduce.job.token.tracking.ids.enabled false Whether to write tracking ids of tokens to job-conf. When true, the configuration property "mapreduce.job.token.tracking.ids" is set to the token-tracking-ids of the job mapreduce.job.token.tracking.ids When mapreduce.job.token.tracking.ids.enabled is set to true, this is set by the framework to the token-tracking-ids used by the job.

mapreduce.task.merge.progress.records 10000 The number of records to process during merge before sending a progress notification to the MR ApplicationMaster. mapreduce.task.combine.progress.records 10000 The number of records to process during combine output collection before sending a progress notification. mapreduce.job.reduce.slowstart.completedmaps 0.05 Fraction of the number of maps in the job which should be complete before reduces are scheduled for the job. mapreduce.job.complete.cancel.delegation.tokens true if false - do not unregister/cancel delegation tokens from renewal, because same tokens may be used by spawned jobs mapreduce.shuffle.port 13562 Default port that the ShuffleHandler will run on. ShuffleHandler is a service run at the NodeManager to facilitate transfers of intermediate Map outputs to requesting Reducers. mapreduce.job.reduce.shuffle.consumer.plugin.class org.apache.hadoop.mapreduce.task.reduce.Shuffle Name of the class whose instance will be used to send shuffle requests by reducetasks of this job. The class must be an instance of org.apache.hadoop.mapred.ShuffleConsumerPlugin. mapreduce.job.node-label-expression All the containers of the Map Reduce job will be run with this node label expression. If the node-label-expression for job is not set, then it will use queue's default-node-label-expression for all job's containers. mapreduce.job.am.node-label-expression This is node-label configuration for Map Reduce Application Master container. If not configured it will make use of mapreduce.job.node-label-expression and if job's node-label expression is not configured then it will use queue's default-node-label-expression. mapreduce.map.node-label-expression This is node-label configuration for Map task containers. If not configured it will use mapreduce.job.node-label-expression and if job's node-label expression is not configured then it will use queue's default-node-label-expression. mapreduce.reduce.node-label-expression This is node-label configuration for Reduce task containers. If not configured it will use mapreduce.job.node-label-expression and if job's node-label expression is not configured then it will use queue's default-node-label-expression. mapreduce.job.counters.limit 120 Limit on the number of user counters allowed per job. mapreduce.framework.name local The runtime framework for executing MapReduce jobs. Can be one of local, classic or yarn. yarn.app.mapreduce.am.staging-dir /tmp/hadoop-yarn/staging The staging dir used while submitting jobs. yarn.app.mapreduce.am.staging-dir.erasurecoding.enabled false Whether Erasure Coding should be enabled for files that are copied to the MR staging area. This is a job-level setting. mapreduce.am.max-attempts 2 The maximum number of application attempts. It is a application-specific setting. It should not be larger than the global number set by resourcemanager. Otherwise, it will be override. The default number is set to 2, to allow at least one retry for AM. mapreduce.job.end-notification.url Indicates url which will be called on completion of job to inform end status of job. User can give at most 2 variables with URI : \$jobId and \$jobStatus. If they are present in URI, then they will be replaced by their respective values. mapreduce.job.end-notification.retry.attempts 0 The number of times the submitter of the job wants to retry job end notification if it fails. This is capped by mapreduce.job.end-notification.max.attempts

mapreduce.job.end-notification.retry.interval 1000 The number of milliseconds the submitter of the job wants to wait before job end notification is retried if it fails. This is capped by mapreduce.job.end-notification.max.retry.interval mapreduce.job.end-notification.max.attempts 5 The maximum number of times a URL will be read for providing job end notification. Cluster administrators can set this to limit how long after end of a job, the Application Master waits before exiting. Must be marked as final to prevent users from overriding this. mapreduce.job.log4j-properties-file Used to override the default settings of log4j in container-log4j.properties for NodeManager. Like container-log4j.properties, it requires certain framework appenders properly defined in this overridden file. The file on the path will be added to distributed cache and classpath. If no-scheme is given in the path, it defaults to point to a log4j file on the local FS. mapreduce.job.end-notification.max.retry.interval 5000 The maximum amount of time (in milliseconds) to wait before retrying job end notification. Cluster administrators can set this to limit how long the Application Master waits before exiting. Must be marked as final to prevent users from overriding this. yarn.app.mapreduce.am.env User added environment variables for the MR App Master processes. Example : 1) A=foo This will set the env variable A to foo 2) B=\$B:c This is inherit tasktracker's B env variable.

yarn.app.mapreduce.am.admin.user.env Environment variables for the MR App Master processes for admin purposes. These values are set first and can be overridden by the user env (yarn.app.mapreduce.am.env) Example : 1) A=foo This will set the env variable A to foo 2) B=\$B:c This is inherit app master's B env variable. yarn.app.mapreduce.am.command-opts -Xmx1024m Java opts for the MR App Master processes. The following symbol, if present, will be interpolated: @taskid@ is replaced by current TaskID. Any other occurrences of '@' will go unchanged. For example, to enable verbose gc logging to a file named for the taskid in /tmp and to set the heap maximum to be a gigabyte, pass a 'value' of: -Xmx1024m -verbose:gc -Xloggc:/tmp/@taskid@.gc Usage of -Djava.library.path can cause programs to no longer function if hadoop native libraries are used. These values should instead be set as part of LD_LIBRARY_PATH in the map / reduce JVM env using the mapreduce.map.env and mapreduce.reduce.env config settings. yarn.app.mapreduce.am.admin-command-opts Java opts for the MR App Master processes for admin purposes. It will appears before the opts set by yarn.app.mapreduce.am.command-opts and thus its options can be overridden user. Usage of -Djava.library.path can cause programs to no longer function if hadoop native libraries are used. These values should instead be set as part of LD_LIBRARY_PATH in the map / reduce JVM env using the mapreduce.map.env and mapreduce.reduce.env config settings. yarn.app.mapreduce.am.job.task.listener.thread-count 30 The number of threads used to handle RPC calls in the MR AppMaster from remote tasks yarn.app.mapreduce.am.job.client.port-range Range of ports that the MapReduce AM can use when binding. Leave blank if you want all possible ports. For example 50000-50050,50100-50200 yarn.app.mapreduce.am.webapp.port-range Range of ports that the MapReduce AM can use for its webapp when binding. Leave blank if you want all possible ports. For example 50000-50050,50100-50200 yarn.app.mapreduce.am.job.committer.cancel-timeout 60000 The amount of time in milliseconds to wait for the output committer to cancel an operation if the job is killed

yarn.app.mapreduce.am.job.committer.commit-window 10000 Defines a time window in milliseconds for output commit operations. If contact with the RM has occurred within this window then commits are allowed, otherwise the AM will not allow output commits until

contact with the RM has been re-established. `mapreduce.fileoutputcommitter.algorithm.version` 2 The file output committer algorithm version valid algorithm version number: 1 or 2 default to 2, which is the original algorithm In algorithm version 1, 1. `commitTask` will rename directory `$joboutput/_temporary/$appAttemptID/_temporary/$taskAttemptID/` to `$joboutput/_temporary/$appAttemptID/$taskID/` 2. `recoverTask` will also do a rename `$joboutput/_temporary/$appAttemptID/$taskID/` to `$joboutput/_temporary/($appAttemptID + 1)/$taskID/` 3. `commitJob` will merge every task output file in `$joboutput/_temporary/$appAttemptID/$taskID/` to `$joboutput/`, then it will delete `$joboutput/_temporary/` and write `$joboutput/_SUCCESS` It has a performance regression, which is discussed in MAPREDUCE-4815. If a job generates many files to commit then the `commitJob` method call at the end of the job can take minutes. the commit is single-threaded and waits until all tasks have completed before commencing. algorithm version 2 will change the behavior of `commitTask`, `recoverTask`, and `commitJob`. 1. `commitTask` will rename all files in `$joboutput/_temporary/$appAttemptID/_temporary/$taskAttemptID/` to `$joboutput/` 2. `recoverTask` actually doesn't require to do anything, but for upgrade from version 1 to version 2 case, it will check if there are any files in `$joboutput/_temporary/($appAttemptID - 1)/$taskID/` and rename them to `$joboutput/` 3. `commitJob` can simply delete `$joboutput/_temporary` and write `$joboutput/_SUCCESS` This algorithm will reduce the output commit time for large jobs by having the tasks commit directly to the final output directory as they were completing and `commitJob` had very little to do.

`yarn.app.mapreduce.am.scheduler.heartbeat.interval-ms` 1000 The interval in ms at which the MR AppMaster should send heartbeats to the ResourceManager `yarn.app.mapreduce.client-am.ipc.max-retries` 3 The number of client retries to the AM - before reconnecting to the RM to fetch Application Status. `yarn.app.mapreduce.client-am.ipc.max-retries-on-timeouts` 3 The number of client retries on socket timeouts to the AM - before reconnecting to the RM to fetch Application Status. `yarn.app.mapreduce.client.max-retries` 3 The number of client retries to the RM/HS before throwing exception. This is a layer above the ipc. `yarn.app.mapreduce.am.resource.mb` 1536 The amount of memory the MR AppMaster needs. `yarn.app.mapreduce.am.resource.cpu-vcores` 1 The number of virtual CPU cores the MR AppMaster needs. `yarn.app.mapreduce.am.hard-kill-timeout-ms` 10000 Number of milliseconds to wait before the job client kills the application. `yarn.app.mapreduce.client.job.max-retries` 3 The number of retries the client will make for `getJob` and dependent calls. This is needed for non-HDFS DFS where additional, high level retries are required to avoid spurious failures during the `getJob` call. 30 is a good value for WASB `yarn.app.mapreduce.client.job.retry-interval` 2000 The delay between `getJob` retries in ms for retries configured with `yarn.app.mapreduce.client.job.max-retries`. `mapreduce.application.classpath` CLASSPATH for MR applications. A comma-separated list of CLASSPATH entries. If `mapreduce.application.framework` is set then this must specify the appropriate classpath for that archive, and the name of the archive must be present in the classpath. If `mapreduce.app-submission.cross-platform` is false, platform-specific environment variable expansion syntax would be used to construct the default CLASSPATH entries. For Linux: `$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*`, `$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*`. For Windows: `%HADOOP_MAPRED_HOME%/share/hadoop/mapreduce/*`, `%HADOOP_MAPRED_HOME%/share/hadoop/mapreduce/lib/*`. If `mapreduce.app-submission.cross-platform` is true, platform-agnostic default CLASSPATH for MR applications would be used: `{{HADOOP_MAPRED_HOME}}/share/hadoop/mapreduce/*`, `{{HADOOP_MAPRED_HOME}}/share/hadoop/mapreduce/lib/*` Parameter expansion marker will be replaced by NodeManager on container launch based on the underlying OS accordingly.

`mapreduce.app-submission.cross-platform` false If enabled, user can submit an application cross-platform i.e. submit an application from a Windows client to a Linux/Unix server or vice versa. `mapreduce.application.framework.path` Path to the MapReduce framework archive. If set, the framework archive will automatically be distributed along with the job, and this path would normally reside in a public location in an HDFS filesystem. As with distributed cache files, this can be a URL with a fragment specifying the alias to use for the archive name. For example, `hdfs:/mapred/framework/hadoop-mapreduce-2.1.1.tar.gz#mrframework` would alias the localized archive as "mrframework". Note that `mapreduce.application.classpath` must include the appropriate classpath for the specified framework. The base name of the archive, or alias of the archive if an alias is used, must appear in the specified classpath. `mapreduce.job.classloader` false Whether to use a separate (isolated) classloader for user classes in the task JVM. `mapreduce.job.classloader.system.classes` Used to override the default definition of the system classes for the job classloader. The system classes are a comma-separated list of patterns that indicate whether to load a class from the system classpath, instead from the user-supplied JARs, when `mapreduce.job.classloader` is enabled. A positive pattern is defined as: 1. A single class name 'C' that matches 'C' and transitively all nested classes 'C\$*' defined in C; 2. A package name ending with a '.' (e.g., "com.example.") that matches all classes from that package. A negative pattern is defined by a '-' in front of a positive pattern (e.g., "-com.example."). A class is considered a system class if and only if it matches one of the positive patterns and none of the negative ones. More formally: A class is a member of the inclusion set I if it matches one of the positive patterns. A class is a member of the exclusion set E if it matches one of the negative patterns. The set of system classes $S = I \setminus E$. `mapreduce.jvm.system-properties-to-log` `os.name`,`os.version`,`java.home`,`java.runtime.version`,`java.vendor`,`java.version`,`java.vm.name`,`java.class.path`,`java.io.tmpdir`,`user.dir`,`user.name` Comma-delimited list of system properties to log on `mapreduce JVM start` `mapreduce.jobhistory.address` 0.0.0.0:10020 MapReduce JobHistory Server IPC host:port `mapreduce.jobhistory.webapp.address` 0.0.0.0:19888 MapReduce JobHistory Server Web UI host:port `mapreduce.jobhistory.webapp.https.address` 0.0.0.0:19890 The https address the MapReduce JobHistory Server WebApp is on. `mapreduce.jobhistory.keytab` /etc/security/keytab/jhs.service.keytab Location of the kerberos keytab file for the MapReduce JobHistory Server. `mapreduce.jobhistory.principal` jhs/_HOST@REALM.TLD Kerberos principal name for the MapReduce JobHistory Server. `mapreduce.jobhistory.intermediate-done-dir` `${yarn.app.mapreduce.am.staging-dir}/history/done_intermediate` `mapreduce.jobhistory.done-dir` `${yarn.app.mapreduce.am.staging-dir}/history/done` `mapreduce.jobhistory.cleaner.enable` true `mapreduce.jobhistory.cleaner.interval-ms` 86400000 How often the job history cleaner checks for files to delete, in milliseconds. Defaults to 86400000 (one day). Files are only deleted if they are older than `mapreduce.jobhistory.max-age-ms`. `mapreduce.jobhistory.max-age-ms` 604800000 Job history files older than this many milliseconds will be deleted when the history cleaner runs. Defaults to 604800000 (1 week). `mapreduce.jobhistory.client.thread-count` 10 The number of threads to handle client API requests `mapreduce.jobhistory.datestring.cache.size` 200000 Size of the date string cache. Effects the number of directories which will be scanned to find a job. `mapreduce.jobhistory.joblist.cache.size` 20000 Size of the job list cache `mapreduce.jobhistory.loadedjobs.cache.size` 5 Size of the loaded job cache. This property is ignored if the property `mapreduce.jobhistory.loadedtasks.cache.size` is set to a positive value. `mapreduce.jobhistory.loadedtasks.cache.size` Change the job history cache limit to be set in terms of total task count. If the total number of tasks loaded exceeds this value, then the job cache will be shrunk down until it is under this limit (minimum 1 job in cache). If this value is empty or nonpositive then the cache reverts to using the property `mapreduce.jobhistory.loadedjobs.cache.size` as a job cache size. Two recommendations for the `mapreduce.jobhistory.loadedtasks.cache.size` property: 1) For every 100k of cache size, set the heap size of the Job History Server to 1.2GB. For example, `mapreduce.jobhistory.loadedtasks.cache.size=500000`, heap size=6GB. 2) Make sure that the cache size is larger than the number of tasks required for the largest job run on the cluster. It might be a good idea to set the value slightly higher (say, 20%) in order to allow for job size growth. `mapreduce.jobhistory.move.interval-ms` 180000 Scan for history files to move from intermediate done

dir to done dir at this frequency. `mapreduce.jobhistory.move.thread-count` 3 The number of threads used to move files.

`mapreduce.jobhistory.store.class` The HistoryStorage class to use to cache history data. `mapreduce.jobhistory.minicluster.fixed.ports` false Whether to use fixed ports with the minicluster

`mapreduce.jobhistory.admin.address` 0.0.0.0:10033 The address of the History server admin interface. `mapreduce.jobhistory.admin.acl` * ACL of who can be admin of the History server. `mapreduce.jobhistory.recovery.enable` false Enable the history server to store server state and recover server state upon startup. If enabled then `mapreduce.jobhistory.recovery.store.class` must be specified. `mapreduce.jobhistory.recovery.store.class` `org.apache.hadoop.mapreduce.v2.hs.HistoryServerFileSystemStateStoreService` The HistoryServerStateStoreService class to store history server state for recovery. `mapreduce.jobhistory.recovery.store.fs.uri` `${hadoop.tmp.dir}/mapred/history/recoverystore` The URI where history server state will be stored if HistoryServerFileSystemStateStoreService is configured as the recovery storage class.

`mapreduce.jobhistory.recovery.store.leveldb.path` `${hadoop.tmp.dir}/mapred/history/recoverystore` The URI where history server state will be stored if HistoryServerLeveldbSystemStateStoreService is configured as the recovery storage class. `mapreduce.jobhistory.http.policy` HTTP_ONLY This configures the HTTP endpoint for JobHistoryServer web UI. The following values are supported: - HTTP_ONLY : Service is provided only on http - HTTPS_ONLY : Service is provided only on https

`mapreduce.jobhistory.jobname.limit` 50 Number of characters allowed for job name in Job History Server web page. `mapreduce.jobhistory.jhist.format` binary File format the AM will use when generating the .jhist file. Valid values are "json" for text output and "binary" for faster parsing. `mapreduce.job.heap.memory-mb.ratio` 0.8 The ratio of heap-size to container-size. If no -Xmx is specified, it is calculated as $(\text{mapreduce}\{\text{maplreduce}\}\text{memory.mb} * \text{mapreduce.heap.memory-mb.ratio})$. If -Xmx is specified but not `mapreduce}\text{memory.mb}`, it is calculated as $(\text{heapSize} / \text{mapreduce.heap.memory-mb.ratio})$.

`yarn.app.mapreduce.am.containerlauncher.threadpool-initial-size` 10 The initial size of thread pool to launch containers in the app master. `mapreduce.task.exit.timeout` 60000 The number of milliseconds before a task will be terminated if it stays in finishing state for too long. After a task attempt completes from TaskUmbilicalProtocol's point of view, it will be transitioned to finishing state. That will give a chance for the task to exit by itself. `mapreduce.task.exit.timeout.check-interval-ms` 20000 The interval in milliseconds between which the MR framework checks if task attempts stay in finishing state for too long. `mapreduce.job.encrypted-intermediate-data` false Encrypt intermediate MapReduce spill files or not default is false

`mapreduce.job.encrypted-intermediate-data-key-size-bits` 128 Mapreduce encrypt data key size default is 128

`mapreduce.job.encrypted-intermediate-data.buffer.kb` 128 Buffer size for intermediate encrypt data in kb default is 128

`mapreduce.task.local-fs.write-limit.bytes` -1 Limit on the byte written to the local file system by each task. This limit only applies to writes that go through the Hadoop filesystem APIs within the task process (i.e.: writes that will update the local filesystem's BYTES_WRITTEN counter). It does not cover other writes such as logging, sideband writes from subprocesses (e.g.: streaming jobs), etc. Negative values disable the limit. default is -1

`mapreduce.jobhistory.webapp.rest-csrf.enabled` false Enable the CSRF filter for the job history web app

`mapreduce.jobhistory.webapp.rest-csrf.custom-header` X-XSRF-Header Optional parameter that indicates the custom header name to use for CSRF protection. `mapreduce.jobhistory.webapp.rest-csrf.methods-to-ignore` GET,OPTIONS,HEAD Optional parameter that indicates the list of HTTP methods that do not require CSRF protection

`mapreduce.job.cache.limit.max-resources` 0 The maximum number of resources a map reduce job is allowed to submit for localization via files, libjars, archives, and jobjar command line arguments and through the distributed cache. If set to 0 the limit is ignored.

`mapreduce.job.cache.limit.max-resources-mb` 0 The maximum size (in MB) a map reduce job is allowed to submit for localization via files, libjars, archives, and jobjar command line arguments and through the distributed cache. If set to 0 the limit is ignored.

`mapreduce.job.cache.limit.max-single-resource-mb` 0 The maximum size (in MB) of a single resource a map reduce job is allow to submit for localization via files, libjars, archives, and jobjar command line arguments and through the distributed cache. If set to 0 the limit is ignored.

`mapreduce.jobhistory.webapp.xfs-filter.xframe-options` SAMEORIGIN Value of the xframe-options

`mapreduce.jobhistory.loadedjob.tasks.max` -1 The maximum number of tasks that a job can have so that the Job History Server will fully parse its associated job history file and load it into memory. A value of -1 (default) will allow all jobs to be loaded.

`mapreduce.job.redacted-properties` The list of job configuration properties whose value will be redacted. `mapreduce.job.send-token-conf` This configuration is a regex expression. The list of configurations that match the regex expression will be sent to RM. RM will use these configurations for renewing tokens. This configuration is added for below scenario: User needs to run distcp jobs across two clusters, but the RM does not have necessary hdfs configurations to connect to the remote hdfs cluster. Hence, user relies on this config to send the configurations to RM and RM uses these configurations to renew tokens. For example the following regex expression indicates the minimum required configs for RM to connect to a remote hdfs cluster: `dfs.nameservices|^dfs.namenode.rpc-address.*$|^dfs.ha.namenodes.*$|^dfs.client.failover.proxy.provider.*$|dfs.namenode.kerberos.principal`