# A description of the configuration options for Swift

**Table 1: Description of the 27 Flink configuration parameters.**

| Configuration Parameters—Description | Abbr. | Range | Default |
|---|---|---|---|
| **parallelism.default**-The default parallelism to use for programs that have no parallelism specified. | PLDT | 1-12 | 1 |
| **jobmanager.heap.mb** - JVM heap size (in megabytes) for the JobManager. | JMHP | 1024-6144 | 1024 |
| **taskmanager.heap.mb** - JVM heap size (in megabytes) for the TaskManager. | TMHP | 2048-6144 | 1024 |
| **taskmanager.memory.off-heap** - the task manager allocates memory which is used for sorting, hash tables, and caching of intermediate results outside of the JVM heap. | TMOH | false, true | false |
| **taskmanager.memory.fraction** - The relative amount of memory that the task manager reserves for sorting, hash tables, and caching of intermediate results. | TMFT | 0.1-0.9 | 0.7 |
| **taskmanager.memory.segment-size** - The size of memory buffers used by the memory manager and the network stack in bytes. | TMSS | 2KB - 2M | 32KB |
| **taskmanager.runtime.hashjoin-bloom-filters** - Flag to activate/ deactivate bloom filters in the hybrid hash join implementation. | TRHB | false, true | false |
| **taskmanager.runtime.sort-spilling-threshold** - A sort operation starts spilling when this fraction of its memory budget is full. | TRSS | 0.1 - 0.9 | 0.8 |
| **taskmanager.runtime.max-fan** - The maximal fan-in for external merge joins and fan-out for spilling hash tables. | TRMF | 80 - 300 | 128 |
| **taskmanager.network.memory.fraction** - Fraction of JVM memory to use for network buffers. | TNMF | 0.1 - 0.9 | 0.1 |
| **taskmanager.net.num-arenas** - the number of Netty arenas (same to taskmanager. numberOfTaskSlots). | TNNA | 1 - 4 | #slot |
| **taskmanager.net.server.numThreads** - The number of Netty server thread (same to taskmanager. numberOfTaskSlots). | TNSN | 1 - 4 | #slot |
| **taskmanager.net.client.numThreads** - The number of Netty client threads (same to taskmanager. numberOfTaskSlots). | TNCN | 1 - 4 | #slot |
| **blob.fetch.num-concurrent** - The number concurrent BLOB fetches (such as JAR file downloads) that the JobManager serves . | BFNC | 50 - 200 | 50 |
| **blob.fetch.retries** - The number of retries for the TaskManager to download BLOBs (such as JAR files) from the JobManager. | BFRT | 50 - 200 | 50 |
| **akka.framesize** - Maximum size of messages which are sent between the JobManager and the TaskManagers. | AKFS | 6M - 21M | 10M |
| **akka.watch.threshold** - Threshold for the DeathWatch failure detector. | AKWT | 8 - 21 | 12 |
| **taskmanager.network.memory.min** - Minimum memory size for network buffers in bytes. | TNMI | 32M - 1024M | 64M |
| **taskmanager.network.memory.max** - Minimum memory size for network buffers in bytes. | TNMA | 1024M - 4096M | 1024M |
| **taskmanager.net.sendReceiveBufferSize** - The Netty send and receive buffer size(defaul to the system buffer size). | TNRB | 763659- 1527317 | #system buffer sizes |
| **blob.fetch.backlog** - The maximum number of queued BLOB fetches (such as JAR file downloads) that the JobManager allows. | BFBL | 500 - 3000 | 1000 |
| **jobmanager.tdd.offload.minsize** - Maximum size of the TaskDeploymentDescriptor's serialized task and job information to still transmit them via RPC. | JTOM | 900 - 4096 | 1024 |
| **fs.overwrite-files** - Specifies whether file output writers should overwrite existing files by default. | FSOF | false, true | false |
| **fs.output.always-create-directory** - File writers running with a parallelism larger than one create a directory for the output file path and put the different result files (one per parallel writer task) into that directory. | FOAC | false, true | false |
| **compiler.delimited-informat.max-line-samples** - The maximum number of line samples taken by the compiler for delimited inputs. | CDIA | 9 - 20 | 10 |
| **compiler.delimited-informat.min-line-samples** - The minimum number of line samples taken by the compiler for delimited inputs. | CDII | 2 - 8 | 2 |
| **compiler.delimited-informat.max-sample-len** - The maximal length of a line sample that the compiler takes for delimited inputs. | CDAL | 1M - 10M | 2M |

# Table 2: Description of the 34 Spark configuration parameters.

| Configuration Parameters—Description | Range | Default |
|---|---|---|
| **spark.reducer.maxSizeInFlight**—Maximum size of map outputs to fetch simultaneously from each reduce task, in MB. | 2–128 | 48 |
| **spark.shuffle.file.buffer**—Size of the in-memory buffer for each shuffle file output stream, in KB. | 2–128 | 32 |
| **spark.shuffle.sort.bypassMergeThreshold**—Avoid merge-sorting data if there is no map-side aggregation. | 100–1000 | 200 |
| **spark.speculation.interval**—How often Spark will check for tasks to speculate, in millisecond. | 10–100 | 100 |
| **spark.speculation.multiplier**—How many times slower a task is than the median to be considered for speculation. | 1–5 | 1.5 |
| **spark.speculation.quantile**—Percentage of tasks which must be complete before speculation is enabled. | 0–1 | 0.75 |
| **spark.broadcast.blockSize**—Size of each piece of a block for TorrentBroadcastFactory, in MB. | 2–128 | 4 |
| **spark.io.compression.codec**—The codec used to compress internal data such as RDD partitions, and so on. | snappy, lzf,lz4 | snappy |
| **spark.io.compression.lz4.blockSize**—Block size used in LZ4 compression, in KB. | 2–128 | 32 |
| **spark.io.compression.snappy.blockSize**—Block size used in snappy, in KB. | 2–128 | 32 |
| **spark.kryo.referenceTracking**—Whether to track references to the same object when serializing data with Kryo | true,false | true |
| **spark.kryoserializer.buffer.max**—Maximum allowable size of Kryo serialization buffer, in MB. | 8–128 | 64 |
| **spark.kryoserializer.buffer**—Initial size of Kryo's serialization buffer, in KB. | 2–128 | 64 |
| **spark.driver.cores**—Number of cores to use for the driver process. | 1–30 | 1 |
| **spark.executor.cores**—The number of cores to use on each executor. | 4–30 | core # |
| **spark.executor.instances**—The number of executors for static allocation. | 7–35 | 2 |
| **spark.driver.memory**—Amount of memory to use for the driver process, in MB. | 1024–33792 | 1024 |
| **spark.executor.memory**—Amount of memory to use per executor process, in MB. | 1024–10240 | 1024 |
| **spark.storage.memoryMapThreshold**—Size of a block above which Spark maps when reading a block from disk, in MB. | 50–500 | 2 |
| **spark.network.timeout**—Default timeout for all network interactions, in second. | 20–500 | 120 |
| **spark.locality.wait**—How long to launch a data-local task before giving up, in second. | 1–10 | 3 |
| **spark.scheduler.revive.interval**—The interval length for the scheduler to revive the worker resource, in second. | 2–50 | 1 |
| **spark.task.maxFailures**—Number of task failures before giving up on the job. | 1–8 | 4 |
| **spark.shuffle.compress**—Whether to compress map output files. | true,false | true |
| **spark.memory.fraction**—Fraction of (heap space - 300 MB) used for execution and storage. | 0–1 | 0.75 |
| **spark.shuffle.spill.compress**—Whether to compress data spilled during shuffles. | true,false | true |
| **spark.speculation**—If set to "true", performs speculative execution of tasks. | true,false | false |
| **spark.broadcast.compress**—Whether to compress broadcast variables before sending them. Generally a good idea. | true,false | true |
| **spark.rdd.compress**—Whether to compress serialized RDD partitions. | true,false | false |
| **spark.serializer**—Class to use for serializing objects that are sent over the network or need to be cached in serialized form. | java,kryo | java |
| **spark.memory.storageFraction**—Amount of storage memory immune to eviction, expressed as a fraction of the size of the region set aside by *spark.memory.fraction*. | 0–1 | 0.5 |
| **spark.default.parallelism**—The largest number of partitions in a parent RDD for distributed shuffle operations. | 8–50 | # |
| **spark.memory.offHeap.enabled**—If true, Spark will attempt to use off-heap memory for certain operations. | true,false | false |
| **spark.memory.offHeap.size**—The absolute amount of memory which can be used for off-heap allocation, in MB. | 10–1000 | 0 |