HBase Default Configuration

property	default value	description
hbase.rootdir	file:///tmp/hbase-${user.name}/hb	The directory shared by region servers. Should be fully-qualified to include the filesystem to use. E.g: hdfs://NAMENODE_SERVER:PORT/HBASE_RO
hbase.master.port	60000	The port master should bind to.
hbase.cluster.distributed	false	The mode the cluster will be in. Possible values are false: standalone and pseudo-distributed setups with managed Zookeeper true: fully-distributed with unmanaged Zookeeper Quorum (see hbase-env.sh)
hbase.tmp.dir	/tmp/hbase-${user.name}	Temporary directory on the local filesystem.
hbase.master.info.port	60010	The port for the hbase master web UI Set to -1 if you do not want the info server to run.
hbase.master.info.bindAddress	0.0.0.0	The address for the hbase master web UI
hbase.client.write.buffer	2097152	Size of the write buffer in bytes. A bigger buffer takes more memory -- on both the client and server side since server instantiates the passed write buffer to process it -- but reduces the number of RPC. For an estimate of server-side memory-used, evaluate hbase.client.write.buffer * hbase.regionserver.handler.count
hbase.master.meta.thread.rescar	60000	How long the HMaster sleeps

		(in milliseconds) between scans of the root and meta tables.
hbase.master.lease.period	120000	HMaster server lease period in milliseconds. Default is 120 seconds. Region servers must report in within this period else they are considered dead. On loaded cluster, may need to up this period.
hbase.regionserver.port	60020	The port an HBase region server binds to.
hbase.regionserver.info.port	60030	The port for the hbase regionserver web UI Set to -1 if you do not want the info server to run.
hbase.regionserver.info.port.autc	false	Info server auto port bind. Enables automatic port search if hbase.regionserver.info.port is already in use. Useful for testing, turned off by default.
hbase.regionserver.info.bindAddr	0.0.0.0	The address for the hbase regionserver web UI
hbase.regionserver.class	org.apache.hadoop.hbase.ipc.HR	An interface that is assignable to HRegionInterface. Used in HClient for opening proxy to remote region server.
hbase.client.pause	1000	General client pause value. Used mostly as value to wait before running a retry of a failed get, region lookup, etc.
hbase.client.retries.number	10	Maximum retries. Used as maximum for all retryable operations such as fetching of the root region from root region server, getting a cell's value, starting a row update, etc. Default: 10.
hbase.client.scanner.caching	1	Number of rows that will be fetched when calling next on a

		scanner if it is not served from memory. Higher caching values will enable faster scanners but will eat up more memory and some calls of next may take longer and longer times when the cache is empty.
hbase.regionserver.lease.period	60000	HRegion server lease period in milliseconds. Default is 60 seconds. Clients must report in within this period else they are considered dead.
hbase.regionserver.handler.coun	25	Count of RPC Server instances spun up on RegionServers Same property is used by the HMaster for count of master handlers. Default is 25.
hbase.regionserver.msginterval	1000	Interval between messages from the RegionServer to HMaster in milliseconds. Use a high value like 3000 for clusters with more than 10 nodes. Default is 1 second so that HBase seems more 'live'.
hbase.regionserver.flushlogentrie	100	Sync the HLog to the HDFS when it has accumulated this many entries. Default 100. Value is checked on every HLog.sync
hbase.regionserver.optionallogflu	10000	Sync the HLog to the HDFS after this interval if it has not accumulated enough entries to trigger a sync. Default 10 seconds. Units: milliseconds.
hbase.regionserver.logroll.period	3600000	Period at which we will roll the commit log.
hbase.regionserver.thread.splitco	20000	How often a region server runs the split/compaction check.
hbase.regionserver.nbreservatio	4	The number of reservation blocks which are used to prevent unstable region servers

		caused by an OOME.
hbase.zookeeper.dns.interface	default	The name of the Network Interface from which a ZooKeeper server should report its IP address.
hbase.zookeeper.dns.nameserve	default	The host name or IP address of the name server (DNS) which a ZooKeeper server should use to determine the host name used by the master for communication and display purposes.
hbase.regionserver.dns.interface	default	The name of the Network Interface from which a region server should report its IP address.
hbase.regionserver.dns.nameser	default	The host name or IP address of the name server (DNS) which a region server should use to determine the host name used by the master for communication and display purposes.
hbase.master.dns.interface	default	The name of the Network Interface from which a master should report its IP address.
hbase.master.dns.nameserver	default	The host name or IP address of the name server (DNS) which a master should use to determine the host name used for communication and display purposes.
hbase.regionserver.global.mems	0.4	Maximum size of all memstores in a region server before new updates are blocked and flushes are forced. Defaults to 40% of heap
hbase.regionserver.global.mems	0.35	When memstores are being forced to flush to make room in memory, keep flushing until we hit this mark. Defaults to 30%

		of heap. This value equal to hbase.regionserver.global.memstore.upperLimit causes the minimum possible flushing to occur when updates are blocked due to memstore limiting.
hbase.hbasemaster.maxregionop	120000	Period to wait for a region open. If regionserver takes longer than this interval, assign to a new regionserver.
hbase.regions.percheckin	10	Maximum number of regions that can be assigned in a single go to a region server.
hbase.server.thread.wakefrequen	10000	Time to sleep in between searches for work (in milliseconds). Used as sleep interval by service threads such as META scanner and log roller.
hbase.hregion.memstore.flush.si	67108864	Memstore will be flushed to disk if size of the memstore exceeds this number of bytes. Value is checked by a thread that runs every hbase.server.thread.wakefrequency.
hbase.hregion.preclose.flush.size	5242880	If the memstores in a region are this size or larger when we go to close, run a "pre-flush" to clear out memstores before we put up the region closed flag and take the region offline. On close, a flush is run under the close flag up to empty memory. During this time the region is offline and we are not taking on any writes. If the memstore content large, this flush could take a long time to complete. The preflush is meant to clean out the bulk of the memstore before putting up the close flag and taking the region offline so the flush that runs under the

		close flag has little to do.
hbase.hregion.memstore.block.m	2	Block updates if memstore has hbase.hregion.block.memstore time hbase.hregion.flush.size bytes. Useful preventing runaway memstore during spikes in update traffic. Without an upper-bound, memstore fills such that when it flushes the resultant flush files take a long time to compact or split, or worse, we OOME.
hbase.hregion.max.filesize	268435456	Maximum HStoreFile size. If any one of a column families' HStoreFiles has grown to exceed this value, the hosting HRegion is split in two. Default: 256M.
hbase.hstore.compactionThresh	3	If more than this number of HStoreFiles in any one HStore (one HStoreFile is written per flush of memstore) then a compaction is run to rewrite all HStoreFiles files as one. Larger numbers put off compaction but when it runs, it takes longer to complete. During a compaction, updates cannot be flushed to disk. Long compactions require memory sufficient to carry the logging of all updates across the duration of the compaction. If too large, clients timeout during compaction.
hbase.hstore.blockingStoreFiles	7	If more than this number of StoreFiles in any one Store (one StoreFile is written per flush of MemStore) then updates are blocked for this HRegion until a compaction is completed, or until hbase.hstore.blockingWaitTime has been exceeded.

hbase.hstore.blockingWaitTime	90000	The time an HRegion will block updates for after hitting the StoreFile limit defined by hbase.hstore.blockingStoreFiles. After this time has elapsed, the HRegion will stop blocking updates even if a compaction has not been completed. Default: 90 seconds.
hbase.hstore.compaction.max	10	Max number of HStoreFiles to compact per 'minor' compaction.
hbase.hregion.majorcompaction	86400000	The time (in miliseconds) between 'major' compactions of all HStoreFiles in a region. Default: 1 day.
hbase.regions.slop	0.3	Rebalance if regionserver has average + (average * slop) regions. Default is 30% slop.
hfile.min.blocksize.size	65536	Minimum store file block size. The smaller you make this, the bigger your index and the less you fetch on a random-access. Set size down if you have small cells and want faster random-access of individual cells.
hfile.block.cache.size	0.2	Percentage of maximum heap (-Xmx setting) to allocate to block cache used by HFile/StoreFile. Default of 0.2 means allocate 20%. Set to 0 to disable.
hbase.hash.type	murmur	The hashing algorithm for use in HashFunction. Two values are supported now: murmur (MurmurHash) and jenkins (JenkinsHash).
zookeeper.session.timeout	60000	ZooKeeper session timeout. HBase passes this to the zk quorum as suggested maximum time for a session.

		See http://hadoop.apache.org/zookeeper/docs/current "The client sends a requested timeout, the server responds with the timeout that it can give the client. The current implementation requires that the timeout be a minimum of 2 times the tickTime (as set in the server configuration) and a maximum of 20 times the tickTime." Set the zk ticktime with hbase.zookeeper.property.tickTime. In milliseconds.
zookeeper.retries	5	How many times to retry connections to ZooKeeper. Used for reading/writing root region location, checking/writing out of safe mode. Used together with ${zookeeper.pause} in an exponential backoff fashion when making queries to ZooKeeper.
zookeeper.pause	2000	Sleep time between retries to ZooKeeper. In milliseconds. Used together with ${zookeeper.retries} in an exponential backoff fashion when making queries to ZooKeeper.
zookeeper.znode.parent	/hbase	Root ZNode for HBase in ZooKeeper. All of HBase's ZooKeeper files that are configured with a relative path will go under this node. By default, all of HBase's ZooKeeper file path are configured with a relative path, so they will all go under this directory unless changed.
zookeeper.znode.rootserver	root-region-server	Path to ZNode holding root region location. This is written

		by the master and read by clients and region servers. If a relative path is given, the parent folder will be ${zookeeper.znode.parent}. By default, this means the root location is stored at /hbase/root-region-server.
zookeeper.znode.safemode	safe-mode	Path to ephemeral ZNode signifying cluster is out of safe mode. This is created by the master when scanning is done. Clients wait for this node before querying the cluster. If a relative path is given, the parent folder will be ${zookeeper.znode.parent}. By default, this means the safe mode flag is stored at /hbase/safe-mode.
hbase.zookeeper.quorum	localhost	Comma separated list of servers in the ZooKeeper Quorum. For example, "host1.mydomain.com,host2.mydomain.com,host By default this is set to localhost for local and pseudo-distributed modes of operation. For a fully-distributed setup, this should be set to a full list of ZooKeeper quorum servers. If HBASE_MANAGES_ZK is set in hbase-env.sh this is the list of servers which we will start/stop ZooKeeper on.
hbase.zookeeper.peerport	2888	Port used by ZooKeeper peers to talk to each other. See http://hadoop.apache.org/zookeeper/docs/r3.1.1/z for more information.
hbase.zookeeper.leaderport	3888	Port used by ZooKeeper for leader election. See http://hadoop.apache.org/zookeeper/docs/r3.1.1/z for more information.

hbase.zookeeper.property.tickTim	3000	Property from ZooKeeper's config zoo.cfg. The number of milliseconds of each tick. See zookeeper.session.timeout description.
hbase.zookeeper.property.initLim	10	Property from ZooKeeper's config zoo.cfg. The number of ticks that the initial synchronization phase can take.
hbase.zookeeper.property.syncL	5	Property from ZooKeeper's config zoo.cfg. The number of ticks that can pass between sending a request and getting an acknowledgment.
hbase.zookeeper.property.dataD	${hbase.tmp.dir}/zookeeper	Property from ZooKeeper's config zoo.cfg. The directory where the snapshot is stored.
hbase.zookeeper.property.clientP	2181	Property from ZooKeeper's config zoo.cfg. The port at which the clients will connect.
hbase.zookeeper.property.maxC	30	Property from ZooKeeper's config zoo.cfg. Limit on number of concurrent connections (at the socket level) that a single client, identified by IP address, may make to a single member of the ZooKeeper ensemble. Set high to avoid zk connection issues running standalone and pseudo-distributed.