Partitioners

The Partitioner, along with the ReplicationStrategy, is responsible for deciding what data is placed on which nodes in your cluster.

A partition is determined by a *partition key;* in the simplest form, a partition is a single row, but a partition may also be made up of multiple rows.

Rows within a partition are ordered, which is often useful at query time. Rows may be retrieved from a partition in ascending or descending order. This is covered in more detail here: http://www.datastax.com/dev/blog/schema-in-cassandra-1-1. A specific example of doing time series data is here: http://www.rustyrazorblade.com/2012/10/cassandra-cql3-and-time-series-data-with-timeuuid/

Cassandra supports the following partitioners, in order of preference. TLDR: always use Murmur3Partitioner in new clusters.

Murmur3Partitioner

This is the default in Cassandra 1.2. It's functionally the same as RandomPartitioner, but Murmur3 is a much faster hash than MD5. We don't care about MD5's cryptographic properties; all we need is a good distribution over the hash space, which Murmur3 delivers.

RandomPartitioner

Default partitioner in Cassandra 1.1 and earlier. Hashes with MD5. Fine to use on legacy clusters but Murmur3 is faster with no downsides.

ByteOrderedPartitioner

An order-preserving partitioner that operates on partition key bytes lexicographically. Not recommended (see below).

OrderPreservingPartitioner

Assumes keys are UTF8 strings. Not recommended both because of this limitation and because globally ordering all your partitions generates hot spots: some partitions close together will get more activity than others, and the node hosting those will be overloaded relative to others. You can try to mitigate with active load balancing but this works poorly in practice; by the time you can adjust token assignments so that less hot partitions are on the overloaded node, your workload often changes enough that the hot spot is now elsewhere. Remember that preserving global order means you can't just pick and choose hot partitions to relocate, you have to relocate contiguous ranges.