Parallel Merge Sort

Parallel merge sort is a good technique for sorting in the case then you wish to distribute the processing. A good way to think about it is the “divide and conquer” method; where the problem of sorting is broken into several small parts which computers can solve. This might be best illustrated with a thought example.

Imagine a list of unsorted numbers that you wish to sort, in no particular order, you can cut the list in half, creating two sibling lists. If the new sub lists themselves are not sorted, then you can cut them in half again and again until they are. This may lead to every value within the original list to be contained exclusively in its own sub list, but that is adequate because a list of one item is sorted.

Now we can reconstruct the original list in a sorted manner by comparing the siblings to each other. One by one, we compare each item in each child list and populate their parent list according to the the sorting rules being applied. This step is applied back up through the parent-child tree of lists until the final list is returned sorted.

The advantage of parallel merge sort as applied to a distributed system is that computing the sort of a sub list can be done completely independently of any other bud list sorting. This means that you can compute the sort of several sub lists across multiple machines and then put the results back together in the end.