In MapReduce algorithm design, synchronization and aggregate results from processes on separate nodes is a tricky aspect, where it must be performed during the shuffle and sort stage. However, there are techniques for controlling execution and managing data flow. The first important technique is in-mapper combining: it is a design pattern for local aggregation, to minimize the amount of partial results that need to be copied across the network(Jimmy Lin and Chris Dyer, 2010, p. 40), by letting the mapper aggregates partial results across multiple input records and only emits intermediate key-value pairs after some amount of local aggregation is performed(Jimmy Lin and Chris Dyer, 2010, p. 68). Another strategy for synchronization are pairs and stripes design patterns, which is useful in problems that require joint events tracking in large numbers of observations. In the pairs approach, keep track of each joint

event separately, whereas in the stripes approach keep track of all events that

co-occur with the same event(Jimmy Lin and Chris Dyer, 2010, p. 68). The order inversion pattern converts computation sequence to sorting problem, and clever sorting of special key-values pairs allow computing aggregate statistics on a set of elements before encountering the elements of the set(Jimmy Lin and Chris Dyer, 2010, p. 41). The value-to-key conversion is the technique and solution to secondary sorting, which is the sorting problem for values with a key in the reduce phase, yy moving part of the value into the key, we can exploit the MapReduce execution framework itself for sorting(Jimmy Lin and Chris Dyer, 2010, p. 68).

Inverted index is a data structure that provides full-text search, used by retrieval engines. There are three steps in web search problems: crawling which is gathering web content, inverted index construction and ranking documents by a query which is retrieva and it requires low latency and high throughput. The inverted indexes problem is most suitable for MapReduce because it is a large distributed sort and group by operation. The baseline implementation is a reasonable application, and the revised implementation improves the scalability. The posting lists are more compact and faster to process for inter compression, and the retrieval is accomplished by document partitioning architecture.

Jimmy Lin and Chris Dyer. 2010. *Data-Intensive Text Processing with MapReduce.* Morgan and Claypool Publishers.