Parallel Database System

* The ideal parallel database system has two key properties:
  + Linear speedup
  + Linear scaleup
* Generic barrier to linear speedup and scaleup: startup, interference, skew
* In order to build scalable multi-processor systems without sacrificing effective power: Shared-disks and Shared-nothing
* Shared-nothing architecture reduces **interference** by minimizing resource sharing, only moves question and answer through the network instead of large quantities of data.
* As parallelism increases, interference on shared resources limits performance, as each processor is given large private cache, which debase performance loading and flushing.
* Consensus architecture for parallel database systems: multiple processors

Algorithmic Aspects of Parallel Data Processing

* Most parallel data management systems compute joins using the parallel hash-join
* When a database is skew-free (each value has a degree less than IN/p, parallel hash-join implies a linear speed-up. But when we increase the number of processors that compute the join, it’s more likely the database will be skewed, and the performance deteriorates
* Hash-based Joins algorithm is the most commonly used in practice, which includes random hashing and range hashing.
* When one of the input relations is significantly smaller than the other one, we normally use Broadcast Join.
* Brent’s theorem provides a general model when transforming the inherent parallelism of a problem to a parallel computational algorithm.