8 Commonly Used Scalable System Design Patterns

Wednesday, December 1, 2010 at 8:36AM

General Chicken in Strategy

Ricky Ho in Scalable System Design Patterns has created a great list of scalability patterns along with very well done explanatory graphics. A summary of the patterns are:

1. **Load Balancer** - a dispatcher determines which worker instance will handle a request based on different policies.
2. **Scatter and Gather** - a dispatcher multicasts requests to all workers in a pool. Each worker will compute a local result and send it back to the dispatcher, who will consolidate them into a single response and then send back to the client.
3. **Result Cache** - a dispatcher will first lookup if the request has been made before and try to find the previous result to return, in order to save the actual execution.
4. **Shared Space** - all workers monitors information from the shared space and contributes partial knowledge back to the blackboard. The information is continuously enriched until a solution is reached.
5. **Pipe and Filter** - all workers connected by pipes across which data flows.
6. **MapReduce** - targets batch jobs where disk I/O is the major bottleneck. It use a distributed file system so that disk I/O can be done in parallel.
7. **Bulk Synchronous Parallel** - a lock-step execution across all workers, coordinated by a master.
8. **Execution Orchestrator** - an intelligent scheduler / orchestrator schedules ready-to-run tasks (based on a dependency graph) across a clusters of dumb workers.
