10 Core Architecture Pattern Variations for Achieving Scalability

Monday, November 7, 2011 at 9:10AM

General Chicken in Strategy

Srinath Perera has put together a strong list of architecture patterns based on three meta patterns: distribution, caching, and asynchronous processing. He contends these three are the primal patterns and the following patterns are but different combinations:

1. **LB (Load Balancers) + Shared nothing Units**. Units that do not share anything with each other fronted with a load balancer that routes incoming messages to a unit based on some criteria.
2. **LB + Stateless Nodes + Scalable Storage**. Several stateless nodes talking to a scalable storage, and a load balancer distributes load among the nodes.
3. **Peer to Peer Architectures (Distributed Hash Table (DHT) and Content Addressable Networks (CAN))**. Algorithm for scaling up logarithmically.
4. **Distributed Queues**. Queue implementation (FIFO delivery) implemented as a network service.
5. **Publish/Subscribe Paradigm**. Network publish subscribe brokers that route messages to each other.
6. **Gossip and Nature-inspired Architectures**. Each node randomly pick and exchange information with follow nodes.
7. **Map Reduce/ Data flows**. Scalable pattern to describe and execute Jobs.

8. **Tree of responsibility**. Break the problem down recursively and assign to a tree, each parent node delegating work to children nodes.
9. **Stream processing**. Process data streams, data that is keeps coming.
10. **Scalable Storages**. Ranges from Databases, NoSQL storages, Service Registries, to File systems.

Please read the original article, List of Known Scalable Architecture Templates, for more details.

Article originally appeared on High Scalability (http://highscalability.com/).