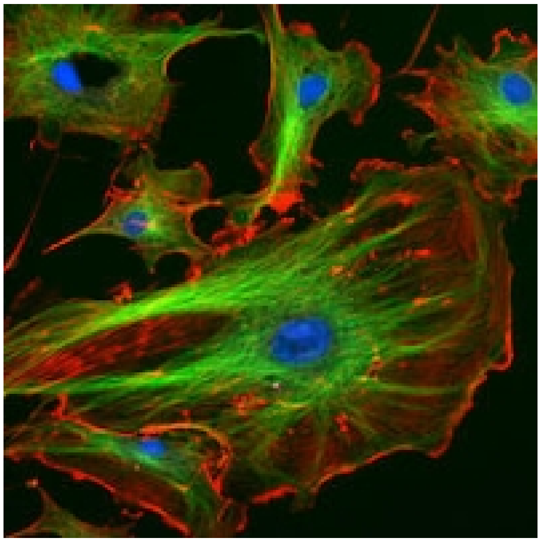
Cell Architectures

Wednesday, May 9, 2012 at 9:15AM

Todd Hoff in Strategy



A consequence of Service Oriented Architectures is the burning need to provide services at scale. The architecture that has evolved to satisfy these requirements is a little known technique called the Cell Architecture.

A Cell Architecture is based on the idea that massive scale requires parallelization and parallelization requires components be isolated from each other. These islands of isolation are called cells. A cell is a self-contained installation that can satisfy all the operations for a shard. A shard is a subset of a much larger dataset, typically a range of users, for example.

Cell Architectures have several advantages:

- Cells provide a unit of parallelization that can be adjusted to any size as the user base grows.
- Cell are added in an incremental fashion as more capacity is required.
- Cells isolate failures. One cell failure does not impact other cells.
- Cells provide isolation as the storage and application horsepower to process requests is independent of other cells.
- Cells enable nice capabilities like the ability to test upgrades, implement rolling upgrades, and test different versions of software.

- Cells can fail, be upgraded, and distributed across datacenters independent of other cells.

A number of startups make use of Cell Architectures:

- Tumblr: Users are mapped into cells and many cells exist per data center. Each cell has an HBase cluster, service cluster, and Redis caching cluster. Users are homed to a cell and all cells consume all posts via firehose updates. Background tasks consume from the firehose to populate tables and process requests. Each cell stores a single copy of all posts.
- Flickr: Uses a federated approach where all a user’s data is stored on a shard which is a cluster of different services.
- Facebook: The Messages service has as the basic building block of their system a cluster of machines and services called a cell. A cell consists of ZooKeeper controllers, an application server cluster, and a metadata store.
- Salesforce: Salesforce is architected in terms of pods. Pods are self-contained sets of functionality consisting of 50 nodes, Oracle RAC servers, and Java application servers. Each pod supports many thousands of customers. If a pod fails only the users on that pod are impacted.

The key to the cell is you are creating a scalable and robust MTBF friendly service. A service than can be used as a bedrock component in a system of other services coordinated by a programmable orchestration layer. It works just as well in a data center as in a cloud. If you are looking for a higher level organization pattern, the Cell Architecture is a solid choice.

Related Articles

http://highscalability.com/blog/2012/5/9/cell-architectures.html

- Startups Are Creating A New System Of The World For IT

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