Seminar - types of distributed systems

Distributed systems - tutor version

Dr Leonardo Mostarda,
School of Science and Technology,
Camerino University, Italy

Version 2.0, 1 March 2013

Question 1

The student should provide a clear distinction between cache and replica. Describe the pros and cons of using replicas.

Solution 1

The first time someone accesses a file, it is downloaded from the appropriate server. Subsequent accesses pull the file from the cache instead. Eventually the cache will fill up, and as new files are downloaded, old files are automatically evicted from the cache.

Caching is not the only way to speed up access to a slow repository. Another option is replication: mirroring the contents of a repository on multiple servers. You might set up a master repository at your main office and create replicas at each remote site. As changes are committed to the master repository, they are mirrored over to the remote replicas.

To put it another way, caching is a pull model data is pulled as it is requested whereas replication is a push model data is pushed as it becomes available, regardless of whether it has been requested.

Replications Advantages Replication has one big performance advantage over caching: it accelerates the first access to a file, not just subsequent accesses. Replication has a number of disadvantages, however, and this advantage is not as clear-cut as it may seem. Combining caching with prefetching has much the same effect.

For example, if your developers start to come in to the office at 8AM, you might kick off a prefetch of all the files they typically use at 7AM, and theyll all be locally cached before anyone arrives. Or, if you work from home, you could start a prefetch each day in the afternoon, and by the time you get home most of the files you need will already be cached. You dont need to prefetch everything just the most important files and this sort of prefetching can be automated using tools like cron.

Another important feature of replication is that it doubles as an efficient way to do backups. If you have an entire copy of your repository offsite, the chances that you will lose all your data are slim. A cache may allow you to recover some files after a loss of data, but it is not a replacement for a real backup system.

Replication also is ideal for disconnected operation. If you lose all network connectivity, having a full replica means you still have access to all the data. In practice, however, disconnected operation is becoming increasingly less important, with Internet and wireless connectivity nearly ubiquitous.

Replications Disadvantages On the flip side, replication has quite a few disadvantages that if you are not using it to perform backups usually outweigh its advantages, especially for large projects.

Building a New Replica

Starting from scratch, it can take a very long time to build a new replica. In effect, you must replay each commit to the repository starting from the beginning. For sufficiently large projects, it may not even be realistic to build an offsite replica purely over a network you may be forced to build a replica at your main site, then physically ship the disks to the remote site.

Caching, on the other hand, has no such upfront costs. The cache can be populated gradually over time, and the speedup from using the cache will grow as more files are populated into it.

Disk Space Cost

Each replica consumes the same amount of disk space as the main repository. The more replicas you need, and the larger your repository, the more you will need to spend on disks to store the replicas. This is usually acceptable for small projects, but once a repository grows large enough that it cannot typically fit on a single commodity, off-the-shelf hard drive, this starts to become troublesome. (Among other things, it becomes impractical for developers who work at home to mirror the repository.)

Caching has no such disk space requiements proportional to the size of the repository. Larger caches can store more files, but even a modestly-sized cache can have large performance benefits. It is practical to set up caches not just at a site-wide level, but also on an individual LAN.

Network Bandwidth Cost

Replication mirrors every change, whether it is needed or not. As such, a replica is constantly consuming network bandwidth. This can overload a remote offices WAN link. In the limit, it is even possible for replication to break down altogether if changes are being committed faster than the data can be mirrored. Also, the mirroring places extra load on the master repositorys server.

Caching, on the other hand, will almost always decrease, not increase, WAN bandwidth usage. A file is not downloaded unless it is really needed.

Replication Lag

Replication is not immediate. It takes time for a change to propagate from the master repository to the replicas. Sometimes the lag may be small, but it may spike if several large changes are committed in a short period of time. When you are working off a replica, you may think you are using the very latest top of tree source code, when in fact you may be any number of changes behind. Depending on how its set up, the replica server might claim that the missing changes dont even exist if you ask it to check out a revision number that hasnt replicated yet, it may give you an error message rather than waiting until the replication catches up to that revision number.

Question 2

When would you advice the use of cluster instead of grid computing?

Solution 2

Hint: you use cluster when you have a huge amount of data to be moved.

Question 3

Provide a short description of the following types of distribute systems: enterprise information systems and pervasive systems.

Solution 3

integration of applications (applications access the database), computing systems everywhere.