CMPSCI 677 Operating Systems **Spring 2013**

Lecture 19: April 8

Lecturer: Prashant Shenoy *Scribe: Siddharth Chidambaram*

19.1 Recap of Fault Tolerance

19.1.1 Two Replication techniques to handle fault tolerance

19.1.1.1 Technique 1

Requests are distributed amongst k replicas . In case of failure of a replica , the requests are redistributed . This technique handles crash fault tolerance . It is a simple and the commonly used technique to handle faults.

19.1.1.2 Technique 2

Each request is sent to all the replicas . All replicas process requests and produce results . Then the replicas vote to make a decision . This handles Crash and Byzantine Faults . However it is much harder a nd more expensive(3k replicas) to implement .

19.1.2 Two Phase Commit (2PC) and Three Phase Commit (3PC)

In Two Phase Commit ,The first phase includes the Coordinator quering all the database replicas on whether a transaction has to be committed or aborted . In the decision phase the results from the replicas are used to make a decision . Even if a single replica wants to abort , the transaction is aborted . This ensures the safety property .

However the failure of the Coordinator blocks the system . In order to handle this blocking a technique called Three phase commit is used. The first two phase is similar to the 2PC . In the third phase , the coordinator tabulates the results and sends them to all the replicas . The replicas send an acknowledgement on receiving this message . Only after receiving the acknowledgement does the coordinator send a commit message.

If the coordinator crashes , the replicas ask each other if they have heard from the coordinator and even if a single replica has a tabulated result (in the precommit stage) the replicas can go ahead and commit . If none of the replicas have a result then the abort the transaction .

The state diagram in the slides shows the various messages passed . The 3PC is always safe irrespective of the failure of the coordinator or the replicas

19.1.3 Paxos

Paxos is another technique for handling fault tolerance . Unlike the 2PC or the 3PC , Paxos looks at the results of the transaction and decides based on the result . A comparison of the results obtained by the different replicas make sure that the Byzantine Faults are taken care of . This is a complex technique and much harder to implement.

19.2 Recovery

Once the crashed server recovers , operations must be performed for it to recover to the current state . This is done by resynchronizing with the other servers. In case of databases , all replicas maintains logs and the replica which has just recovered can ask another replica for its log and do a log replay to get itself updated . Checkpointing is another means of recovery . The replicas maintain checkpoints and upon a crash it rolls back to its previous checkpoint and then does a log replay from that point .

Independent Checkpointing can lead to inconsistencies in that case the replicas has to rollback until the last consistent state . This cascading rollback can lead to a domino effect . However use of techniques like distributed snapshot solves the problem.

19.3 Case Study on Data Centers and Cloud Computing

19.3.1 Data Centers

Data centers are collections of servers and storage farms for supporting applications that require distributed storage and computations . They are used by Enterprises and Internet companies for running their website , business applications and other application which require data processing . Spire is an example of an application that can be run on a data center .

Two kinds of Data Center Architecture include Traditional and Modern Architecture . In the Traditional architecture there is a static mapping between the applications and the servers they run on. A system administrator decides the mapping and has to manually manage the servers. However with the fluctuations in the workload this static mapping might not the most effective technique . Modern Data centers provide a flexible mapping and also have increased automation thus handling the needs of these applications

19.3.1.1 Components

- racks of servers
- storage arrays
- networking components like routers
- Cooling infrastructure to handle the heat generated by the servers
- Electricity to power the servers
- USB/battery backup/ generators for backup power.

Another way to handle the fluctuating workloads is to make use of Modular Data Centers . In this technique the containers contain servers and are shipped based on demand . They are much effective since these servers and preinstalled and configured . The costs involved in installation is avoided and the data centers can also expand easily.

One major advantage of Data center is the use of virtualization .A single physical 16 core machine can work as 4 quad core machines working with different platforms and with different functionality. With the VMs adjustments in resource allocations can happen on the fly . The slide includes an animation which depicts this . As discussed in previous lectures VM migration can achieved within a LAN .

The data centers can be used either as Virtual servers or a virtual Desktop. The main advantages of using a data center as a virtual server include faster deployment and easier maintenance . In virtual Desktops, A low end machine is given to the user and a virtual machine on a remote server . The user using the low end machine connects to the desktop on the virtual machine . This is called thin client computing . This is much easier to manage and maintain .

Two major challenges that the data center face include Resource Management and Energy Efficiency . Resource Management include handling unpredictable workloads , working towards higher performance , achieving efficient usage of servers and storage resources. Energy efficiency is one major concern since servers consume huge amounts of energy . Hence use of renewable sources of energy is preferred . However there are issues with the solar or the wind energy since they are not reliable and the system irrespective of the conditions has to cater the needs of the user .

19.3.2 Cloud Computing

Cloud is a data centers were the user can request machine on demand and pay for what is being used. It uses a shared infrastructure were a every server is shared among different users .

19.3.2.1 Types of Cloud

- Software as a Service

  The user's application is hosted on a cloud . They are managed by the provider . Some of the applications include Gmail , Google Docs , Sales Force . In case of gmail , the mail server is provided by google , the user logs in and access the server . The user is abstracted from the hardware or the software used .

- Platform as a Service

  The user gets hardware and some software . Applications can be developed and deployed by the user . The applications are managed by the provider . Based on the needs of the application the provider assigns resources and takes care of the needs of the application . Some examples include Azure and Google App Engine

- Infrastructure as a Service

  The user rents a bare bone machine and storage . No software or application is available . The user can use any software , OS and applications. Some examples include Amazon web services and at and t.

Google App Engine It is an example of a PaaS. The user is provided the hardware , a set of softwares and can build the application in python or Java . The scalability and performance are managed by google

. Google figures out how many threads to allocate for an application based on the need of the application. Its not based on Virtualization

Amazon EC2: The user rents a bare bone machine and storage .The User decides on the kind of machine needed to serve his/her application . The machines can be small/medium/large based on the need. The machines can be provisioned in minutes unlike buying machine which take quite sometime .