TK1 1st Exercise

Kecen Li, Shule Liu, Zhen Chen, Letian Feng, Xin Zhang

9. November 2016

1 Definition Distributed Systems

A distributed system consists a set of computing devices and the communication network connecting them. Each computing device works independently with shared or their own memories and collaborate with the other computers through message exchange to accomplish sophisticated tasks that is impossible for any single computer to complete. And the absence of a fraction of the machines in the system doesn't necessarily make itself paralysed, in fact, it will continue to function in most of such scenarios. However, from the aspect outside the system, it is viewed as if it is a single computer with super computing capability.

Basic Problems:
1. Lack of Global State
2. Lack of Common Clock
3. Indeterministic Behaviour

Requirements:
1. Heterogeneity Support
2. Scalability
3. Security

2 Abstraction Levels

Level 1: Physical Configuration
Level 2: Logical Configuration
Level 3: Process Network (Logical Distribution)
Level 4: Distributed Algorithm

3 System Models

C/S:
Client and Server play different roles during the communication, Client will initialise a connection whenever requesting a service, server as the service provider will answer requests from clients. Server has to be always running to maintain high availability of the its services.

P2P:
Is a communication network without designated servers and clients, without central control and without reliable partners.

4 Programming Abstractions

DOS Approach: In the price of compromising the autonomy of individual computing nodes, a distributed operating system, which is one layer up and in control of these homogeneous computers, is able to provide functionalities through concurrent/parallel programming.

DDBMS Approach: As a top level management system, DDBMS incorporates multiple database systems and provides general database features. In addition, isolated sequential programming with simple reading and writing from the distributed database system becomes feasible without exchanging messages among distributed applications. But this is also the bottleneck that makes implementation of certain distributed algorithm difficult and more importantly, it gets increasingly hard to scale as the number of databases grows.

Protocol Approach: This approach is designed to tackle specific isolated problems with standard protocol for communication thus not a viable large scale solution.

Distributed Programming Approach: With sequential programming and extensions, distributed programming is achieved through distributed programs that runs on different computers throughout the network, possibly with aid from a distributed runtime system. There is no enforcement on operating systems, databases and programming languages as compared to other approaches. Comparing to the original approach, compiler cannot see the entire distributed programs, thus not able to providing benefits such as avoiding race conditions.

5 Distributed Programing Language Approach

Pros:
1. The DRTS works closely with the programming languages to retrieve information concerning the distributed programs from compiler, thus an increase in efficiency.
2. Absence of enforcement on language leads to competition and advancement in the field.
3. Distributed programming languages are designed and implemented to address issues such as heterogeneity, migration and autonomy which to some extend, are compromised in other distributed system approaches.
4. Global scale support for distributed programming is possible.

Cons:
1. Have to deal with multiple DRTSes at a time under different environments and configurations since they are written in different languages.
2. It is implemented with little consideration of the operating system that distributed programs will run on, thus not optimising the efficiency in this regard.
3. From the programmers's perspective, learning new language constructs is not always desired since it evolves very fast.

4. The support that programmer get from compiler today is not comparable to the original approach which makes distributed programming nowadays much more difficult.