Principles of Computer System Design - Assignment 1

Thorbjørn Christensen
Steffen Karlsson
Kai Ejler Rasmussen

November 20, 2014

Exercises

Question 1: Fundamental Abstractions

1. We split the address used in READ and WRITE into a group of X least significant bits and a group of $Bits_{max} - X$ bits. The most significant group will refer to a machine using a name resolution service, in the client, and the least significant group will refer to a memory address on the specified server. Each server is added sequentially in the bit domain to allow for a sequential memory domain. The scalability of this solution depends on the distribution between the two groups in the naming. A possible distribution in a 64 bit domain could be $2^{56_{bits}}$ memory addresses and $2^{8_{bits}}$ possible machines.

   This design, if configured correctly, will automatically change server when the address rolls over in the most significant group.

2. The READ and WRITE API are both using a name resolution service (ie. lookup table or other service) to resolve the ip of the receiving server. The value is then read or written using RPC.

```
READ(name)
   ip <- lookupServer(name & SERVER_BITMASK)
   memory <- name & MEMORY_BITMASK
   return getValueFromServer(ip, memory)
```

```
WRITE(name, value)
   ip <- lookupServer(name & SERVER_BITMASK)
   memory <- name & MEMORY_BITMASK
   writeValueToServer(ip, memory, value)
```

3. The READ/WRITE API of the abstraction layer should be atomic to keep consistency, since generic READ/WRITE operations are atomic. To achieve this, a lock is obtained during the operations on the remote machine.

4.

Question 2: Techniques for Performance

1.
2.
3.

Questions for Discussion on Architecture