

Programming Abstractions in Cloud

Shashikant Ilager

*Cloud Computing and Distributed Systems (CLOUDS) Laboratory,
School of Computing and Information Systems
The University of Melbourne, Australia*

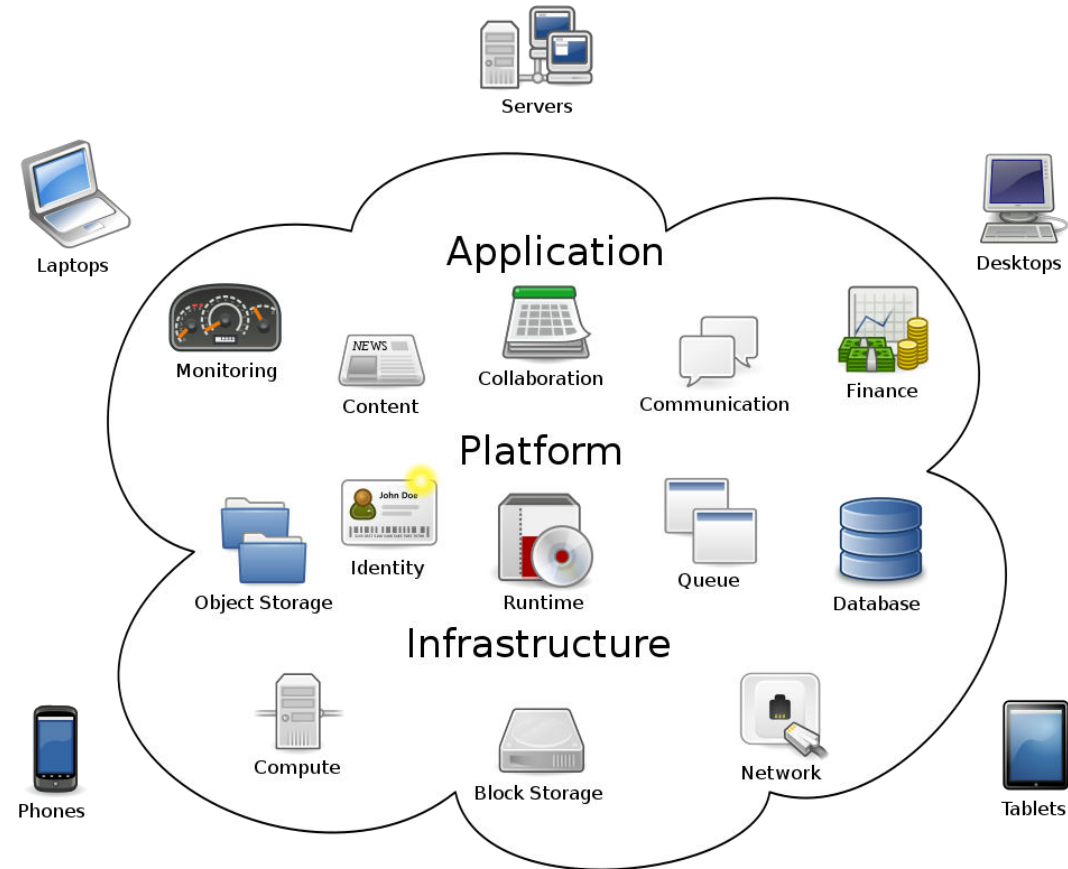


Outline

- Cloud Computing
 - Computing as a utility
- Programming networked computers
 - TCP ports and sockets
 - RPC
- Cloud Application Platforms
 - Platform as a Service (PaaS)
- Aneka: A Cloud Middle-ware Platform
- Summary

Cloud Computing

- Outsource IT facilities to cloud providers
- Avoid expensive up-front investments
- Computing as a utility
- On-demand
- **Pay for what you use**
- **Virtualized** resources



Programming Networked Computers

- Sockets

- Most familiar communication way for networked applications
- Enables applications to interact with transport layer
- Bound to a local port
- IP address + port = Socket ID/ address
- Socket can be used for sending and receiving data.
- Acts as a programming interface to application code and transport layer
- Each socket is associated with a protocol (UDP or TCP)

Programming Networked Computers

- Transmission Control Protocol (TCP)
 - Connection Oriented
 - Provides an abstraction for a two-way stream (called as packets).
 - Stream provide the basis for producer/consumer communication.
 - Data sent by the producer are queued until the consumer is ready to receive them.
 - Example use cases – http, ftp
- TCP Socket Programming Demo

Programming Networked Computers

- Socket handling becomes complex when applications scale to large number of servers
 - Typical application in cloud spans from few hundred to thousands of servers
 - Individual port to each server
 - Read write buffers
 - Synchronization
 - Exception handling

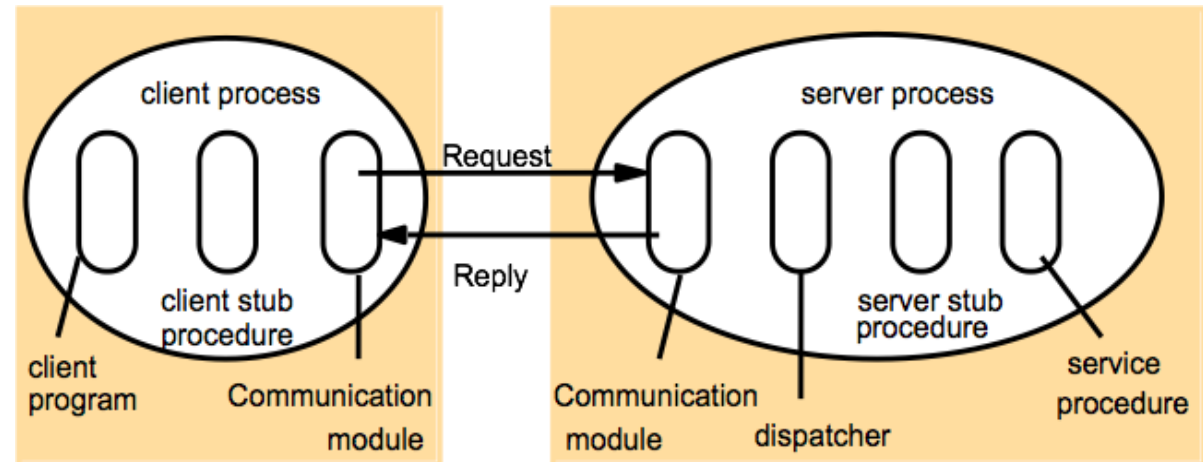
Programming Networked Computers

- Remote Procedure Call (RPC)

- RPCs enable clients to execute procedures in server processes based on a defined service interface (Procedural languages- C, Fortran, Go)
- Remote Method Invocation (RMI) for Object Oriented languages (JAVA)
- Higher abstraction than sockets

- Key components of RPC:

- Communication Module
- Client Stub Procedure
- Dispatcher
- Server stub procedure



Programming Networked Computers

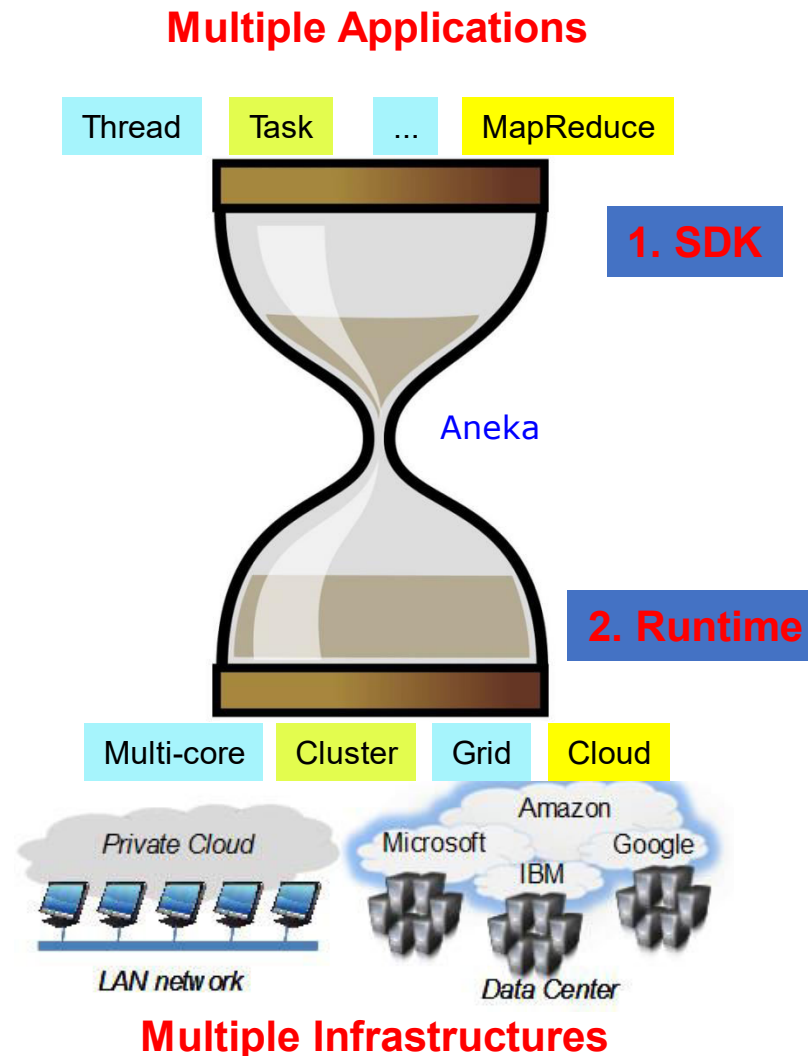
- RPC/ RMI is useful in simple applications where resources are usually static in nature.
- Cloud applications are dynamic that requires:
 - Elasticity – On demand resource provision and de-provision
 - Monitoring – Heartbeat, Budget Constraints, etc.
 - Robustness – Availability, Failure management
- Middleware that seamlessly manages dynamic resources and handles runtime network communication is necessary.

Cloud Application Platforms

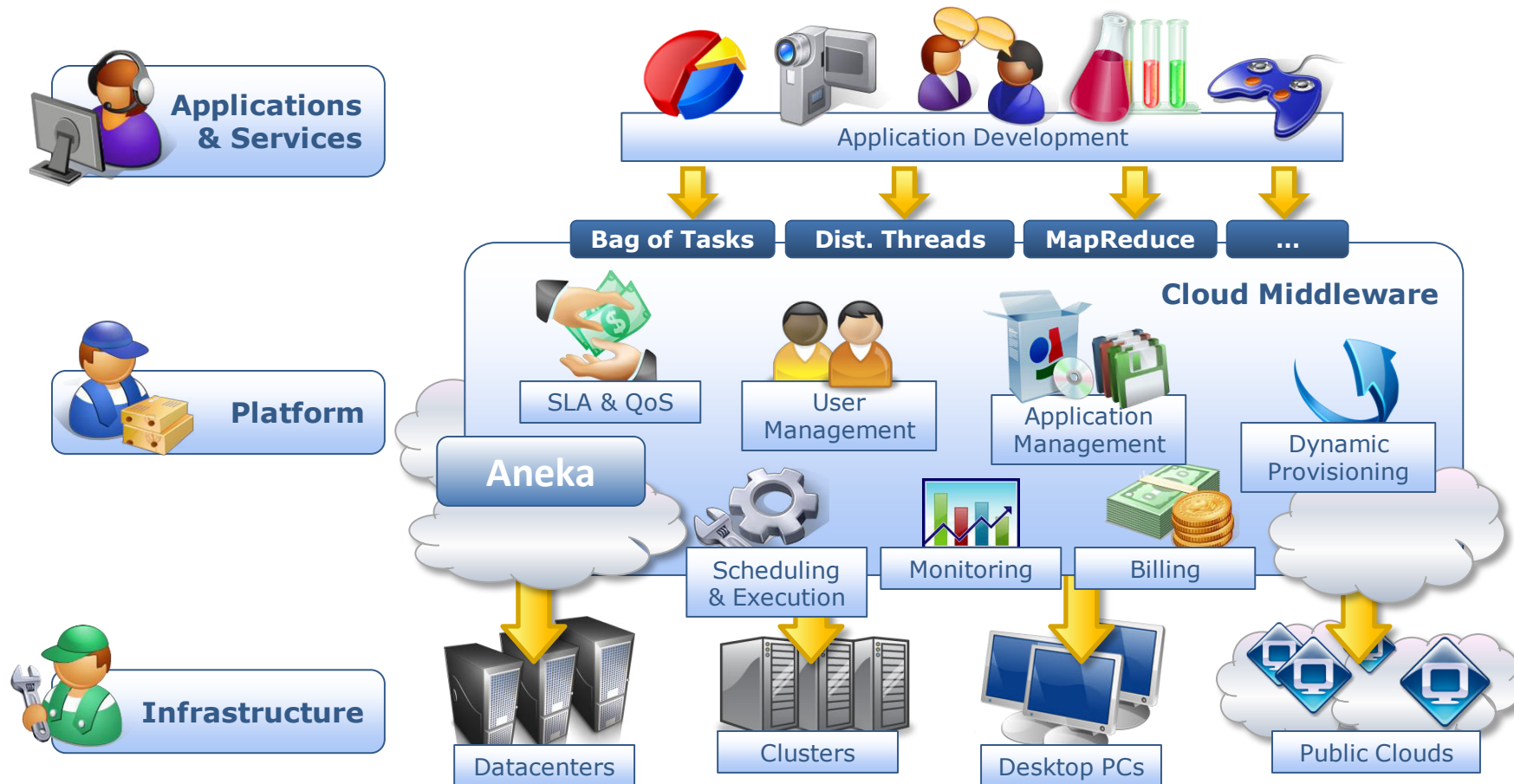
- Cloud Application Platforms (CAP) provide highest level of abstractions to build the applications
- Abstraction through SDKs, APIs and middle-ware platforms
- Less efforts on resource management and more focus on application
- Examples include
 - Google App Engine
 - Deploy web applications on the fly
 - Microsoft Azure PaaS platforms
 - Highly suitable platform for enterprise application developments
 - Aneka (Manjrasoft, CLOUDS Lab, UniMelb)
 - Cloud application platform with multiple programming models

Aneka: Cloud Application Platform(CAP) for Resource-Intensive/Elastic Apps

- Platform supporting multiple Cloud programming models (Task, Thread, MapReduce)
- **SDK (Software Development Kit)** containing APIs for multiple programming models and tools
- **Runtime** Environment for managing application execution on Clouds
- Suitable for
 - Development of Enterprise Cloud Applications
 - Cloud enabling legacy applications
- **Portability for Customer Apps:**
 - Enterprise ↔ Public Clouds
 - .NET/Win ↔ Mono/Linux



Aneka as a Cloud Application Platform



Aneka as a Cloud Application Platform

- Aneka Demo

Summary

- Cloud Platforms like Aneka simplifies the process to build large scale distributed systems
- Although middle-ware platforms provide abstraction to programmer, the underlying complex operations are still need to be implemented inside these platforms.
- Acquiring skills to build such tools and platforms is crucial.

Questions?

For any queries, please contact me:

shashi.ilager@unimelb.edu.au