CLOUD COMPUTING APPLICATIONS

Cloud Services

Prof. Roy Campbell

Contents

- Web Services

- Remote Procedure Calls
    - RMI, SOAP

- HTTP, REST

- JSON, XML

- Load Balancing

Introduction to Web Servers and Load Balancing

- Request enters a router

- Load balancing server determines which web server should serve the request
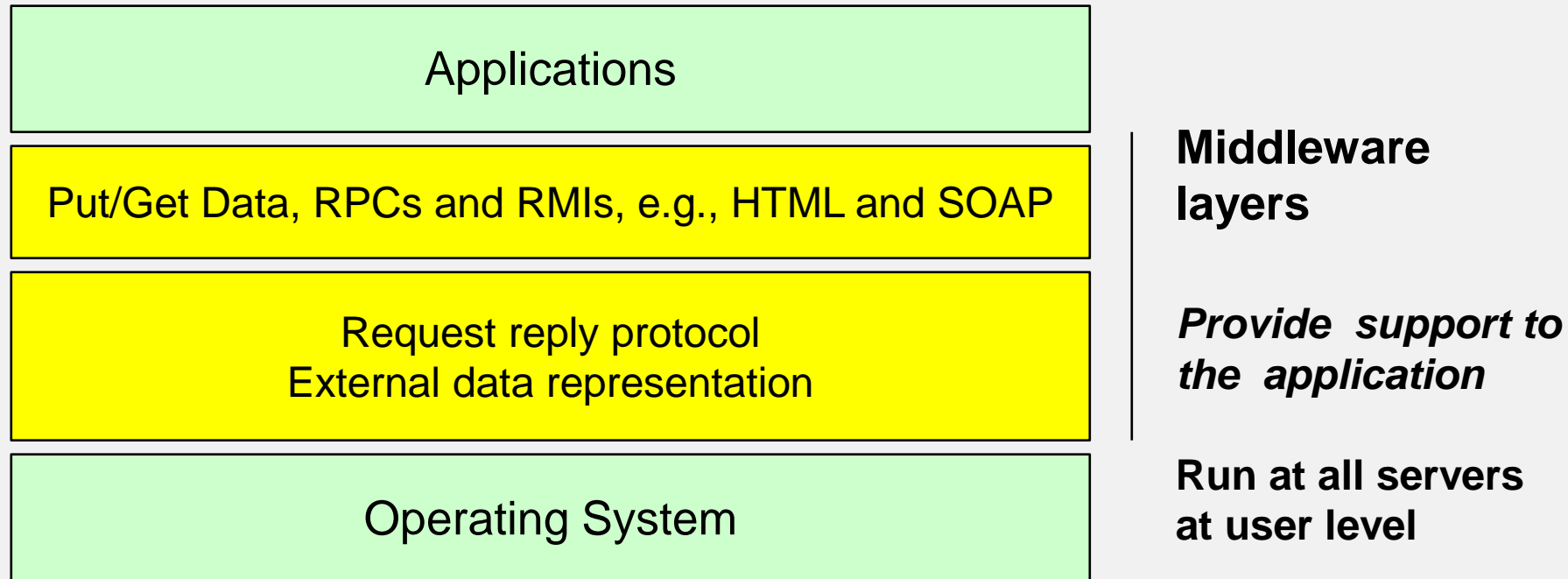
- Sends the request to the appropriate web server

Internet

Router

Load-balancing server

→ Request

–·–▸ Response

Web Servers
Traditional Web Cluster
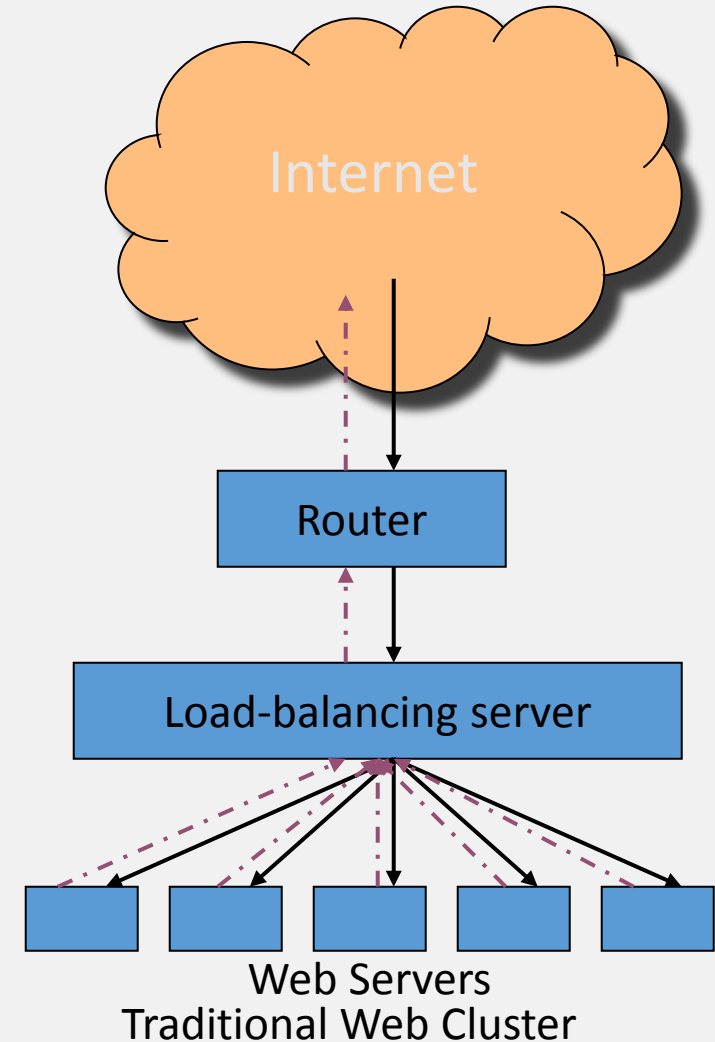
Middleware Layer Definition

- *Software that provides services to applications beyond those generally available at the operating system*

- Middleware implements functionalities that are common across many different applications
  - No need to reinvent the wheel (e.g., message parsing) every time you need to do something

- Building distributed systems while maintaining our code is not very different from a single-node program

Middleware Layers

Applications

Put/Get Data, RPCs and RMIs, e.g., HTML and SOAP

Request reply protocol External data representation

Operating System

Middleware layers

Provide support to the application

Run at all servers at user level

RMI = Remote Method Invocation
CORBA = Common Object Request Brokerage Architecture
SOAP = Simple Object Access Protocol

SOAP —
Simple Object Access Protocol

- Transmitted by HTTP or SMTP (or many others)

- Coded in XML (Can be decoded on any machine)

- Return value: Any XML document

- Underlies Web Services Description Language (WSDL)

Common Abstractions

- In single-node programs, we often rely on concepts such as
  - Procedure calls
  - Objects
  - Shared memory
  - …

- A Middleware Layer can provide the same abstractions to distributed applications!

- We'll start with objects and procedure calls

Local Objects

- Within one process' address space
- Object
  - Consists of a set of data and a set of methods
  - For example, C++ object, Chord object at a client (Chord data structures + functions at a node)
- Object reference
  - An identifier by which objects can be accessed, i.e., a *pointer*
- Interface
  - Provides a definition of the signatures of a set of methods (i.e., the types of their arguments, return values, and exceptions) without specifying their implementation
  - For example, {put(objectname), get(objectname)} interface for Chord object. Same interface also applies to other P2P objects such as Gnutella, Kazaa, etc.

Remote Objects

- May cross multiple process' address spaces
- Remote method invocation
  - Method invocations between objects *in different processes (processes may be on the same or different host)*
  - Remote Procedure Call (RPC): procedure call between functions on different processes in non-object-based system
- Remote objects
  - Objects that can receive remote invocations
- Remote object reference
  - An identifier that can be used globally *throughout a distributed system* to refer to a particular unique remote object
- Remote interface
  - Every remote object has a remote interface that specifies which of its methods can be invoked remotely, e.g., CORBA interface definition language (IDL)