

クラウド開発演習（課題解決型学習） (C1) SaaS

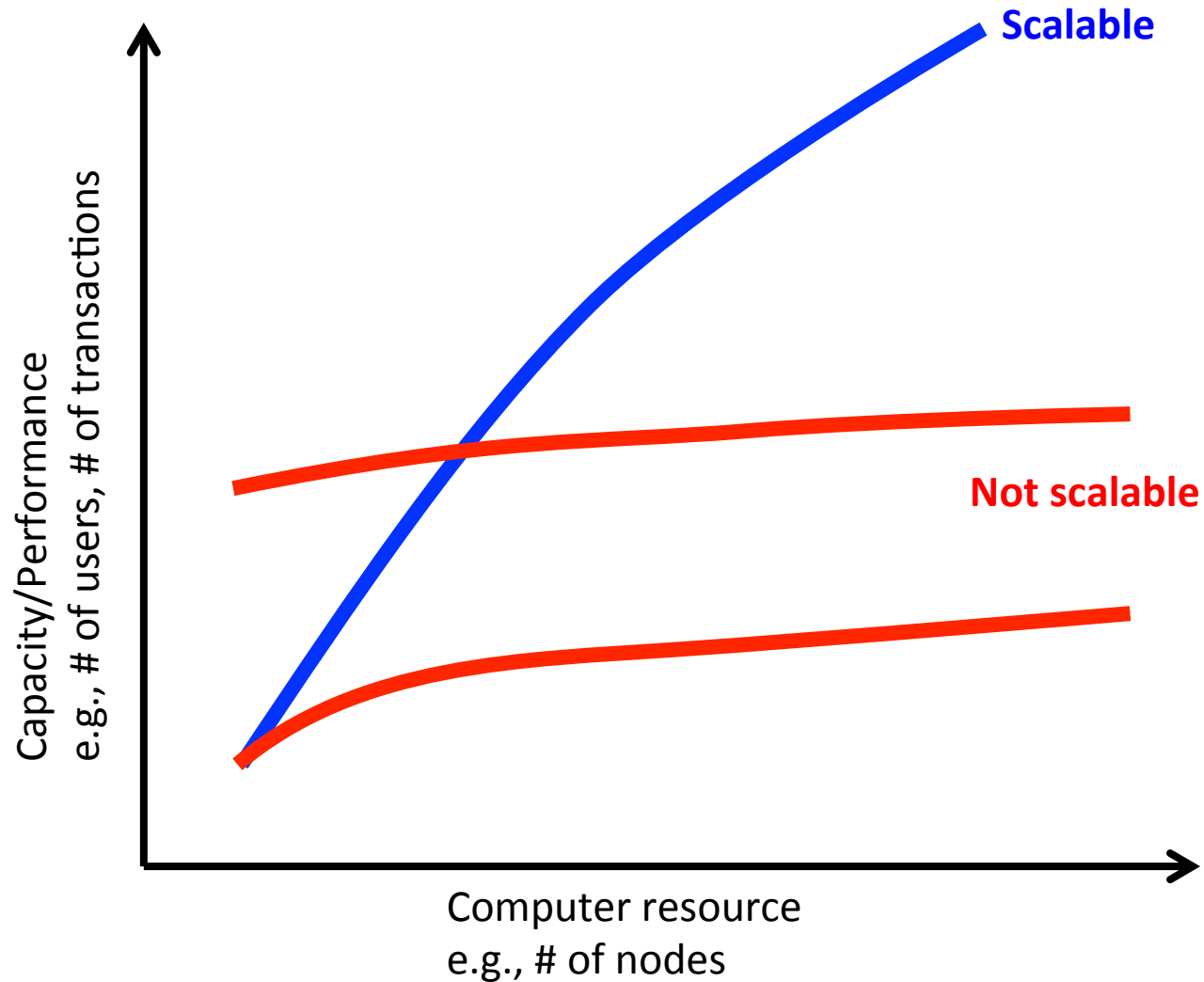
ソフトウェア・クラウド開発プロジェクト実践III

浅井大史
2015年5月22日

(C1) SaaS

- Goal
 - Scalable design of software (as a service)

What does “scalable” mean?

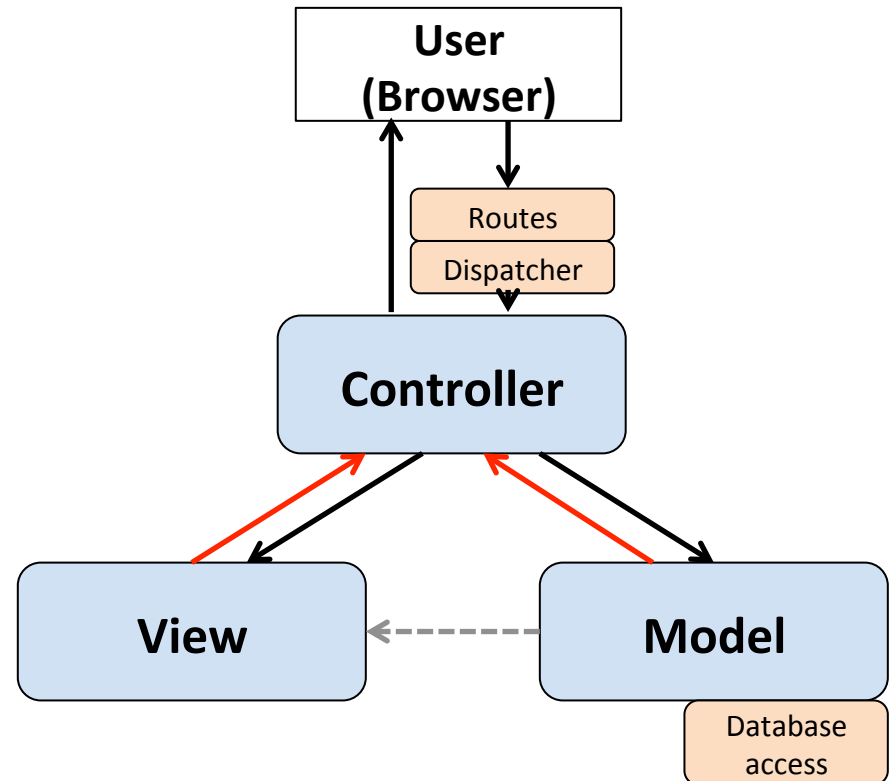


System Design

- Strong coupling (密結合)
- **Loosely coupling (疎結合)**
 - Messaging between components
 - RPC (Remote Procedure Call)
 - REST API
 - Message Queue (e.g., 0MQ)
 - synchronous/asynchronous
 - 1:1/pub-sub

MVC pattern

- MVC: Model-View-Controller
 - Model
 - View
 - Controller



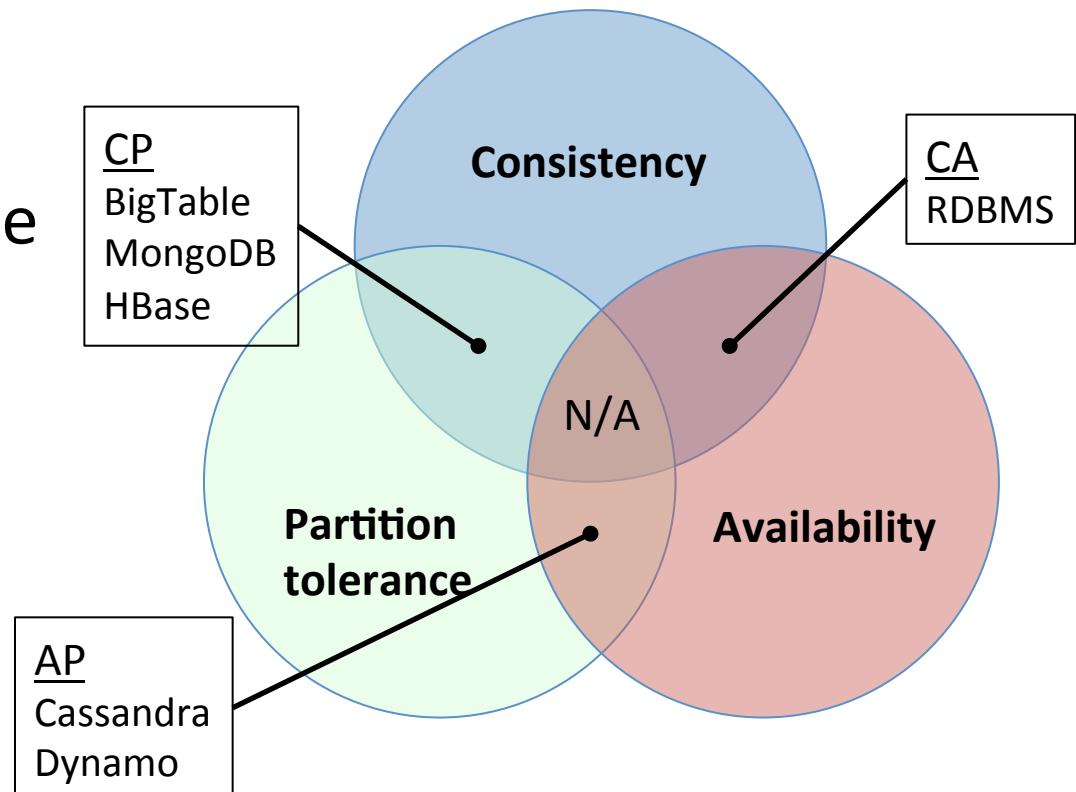
Database

- CAP theorem
- ACID
 - RDBMS
- BASE
 - NoSQL

CAP theorem

Eric A. Brewer, "Towards Robust Distributed Systems," PODC Keynote, 2000

- Impossible to simultaneously provide all of
 - **C**onsistency
 - **A**vailability
 - **P**artition tolerance



ACID

- Properties of a reliable transaction system (e.g., RDBMS)
 - defined by Jim Gray
 - Atomicity (原子性)
 - All transaction successes
 - or no state changes at failure
 - Consistency (一貫性)
 - Ensure valid state after any transaction
 - Isolation (独立性)
 - Serially execute concurrent execution of transactions
 - Durability (永続性)
 - Fault tolerance

※ No guarantee of Availability in ACID

Eventual Consistency

Werner Vogels, "Eventually Consistent," ACM Queue, 2008

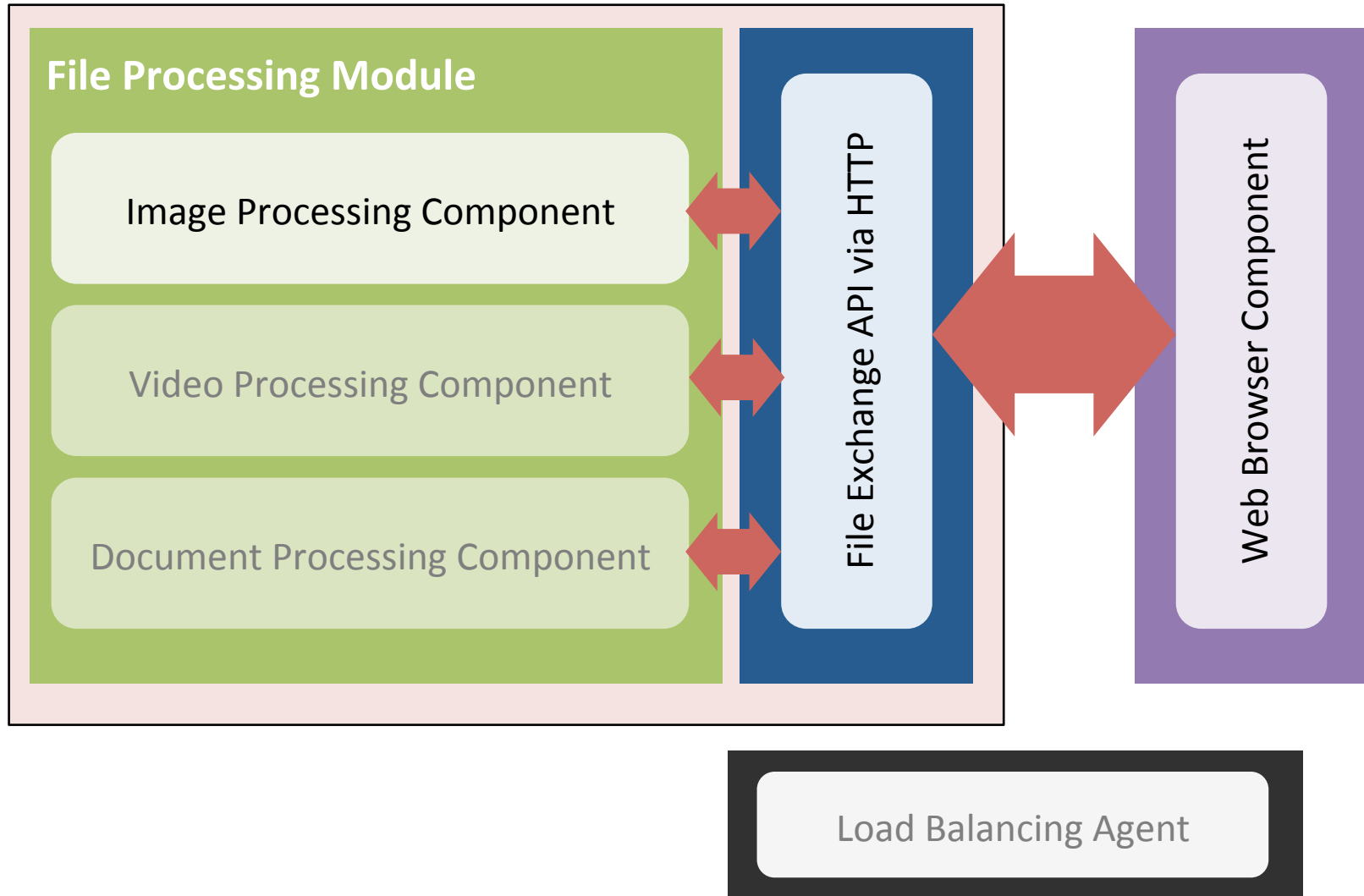
- a.k.a. BASE
 - Basically Available
 - available as the system is operated normally
 - Soft-state
 - hold clients' state but can be regenerated or replaced if needed (state may change over time)
 - between stateless and stateful (hard-state)
 - Eventually consistent
 - guarantee the system will become consistent over time

Eventual consistency increases the system complexity though...

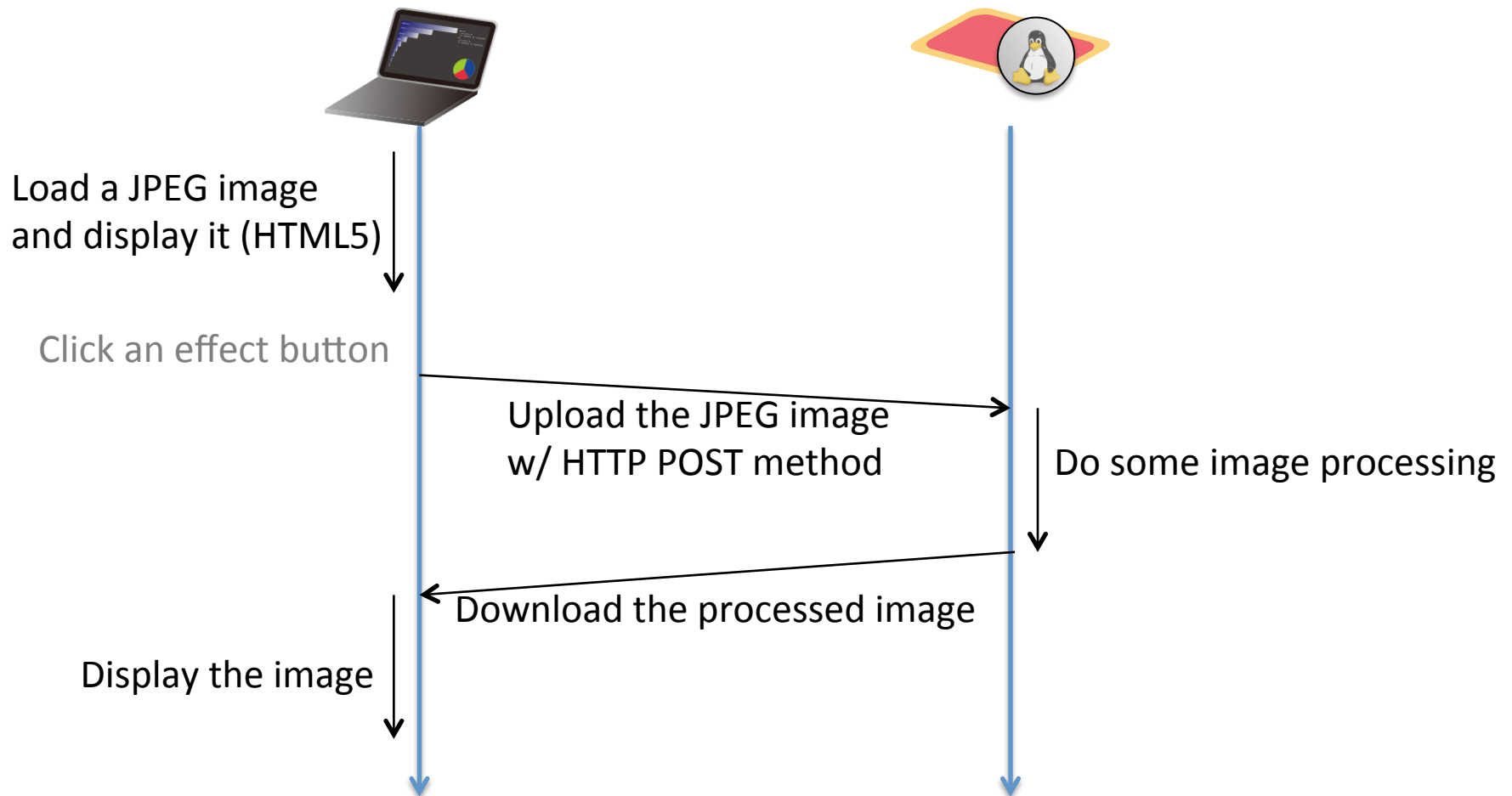
Day 1

- Interim goal: Understand SaaS-related technology
 - DNS-based load balancing
 - Loosely coupled system using REST (in a broad sense)
- Requirements
 - Image processing module
 - Face recognition
 - Adaptive threshold
 - HTML 5
 - Ruby on Rails (MVC pattern)
 - <https://github.com/drpnD/SaaS-Day1>

Day 1: System Design



Day 1: Image Processing Procedure



<https://github.com/hirochika/SaaS-Day1>

Screenshots



Image processing software sample

Load JPEG file: no file selected

Apply

Apply

Input image

Output image

Status: None

Use the information provided on this site on your own responsibility.
(C) 2013 Hirochika Asai, The University of Tokyo. All rights reserved.



Image processing software sample

Load JPEG file: no file selected

Apply

Apply

Input image

Output image

Status: None

Use the information provided on this site on your own responsibility.
(C) 2013 Hirochika Asai, The University of Tokyo. All rights reserved.



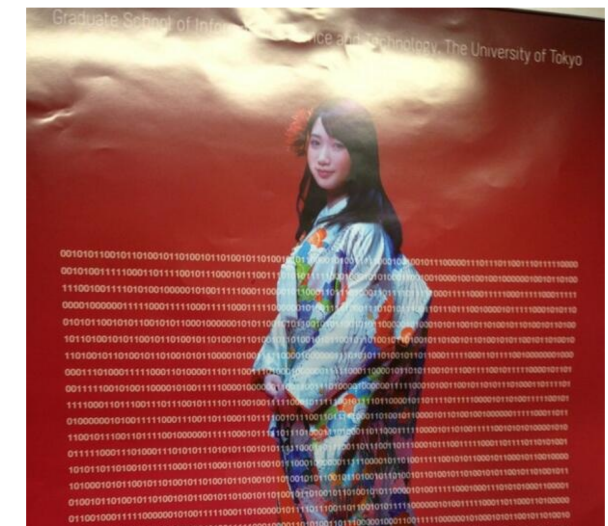
Image processing software sample

Load JPEG file: 1 photo

Apply

Apply

Input image

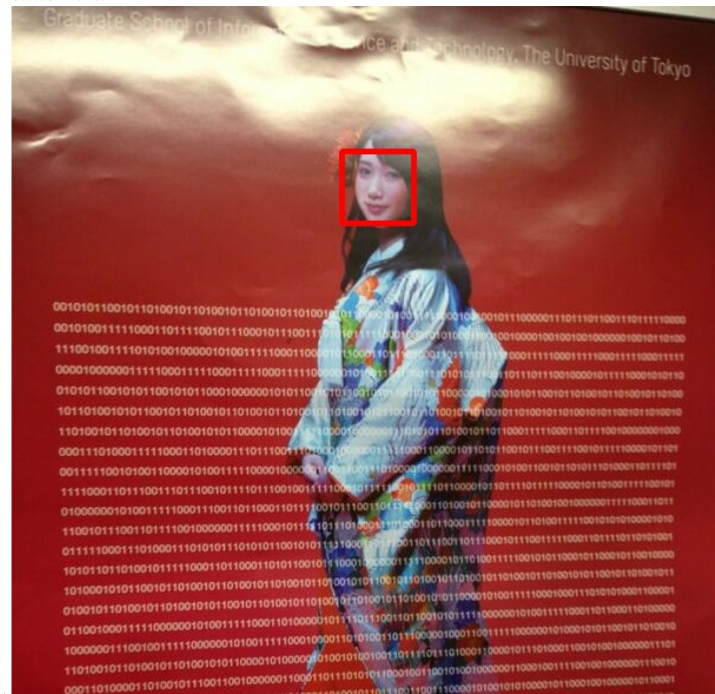


Screenshots



Output image

Status: Done.



Face recognition



Output image

Status: Done.



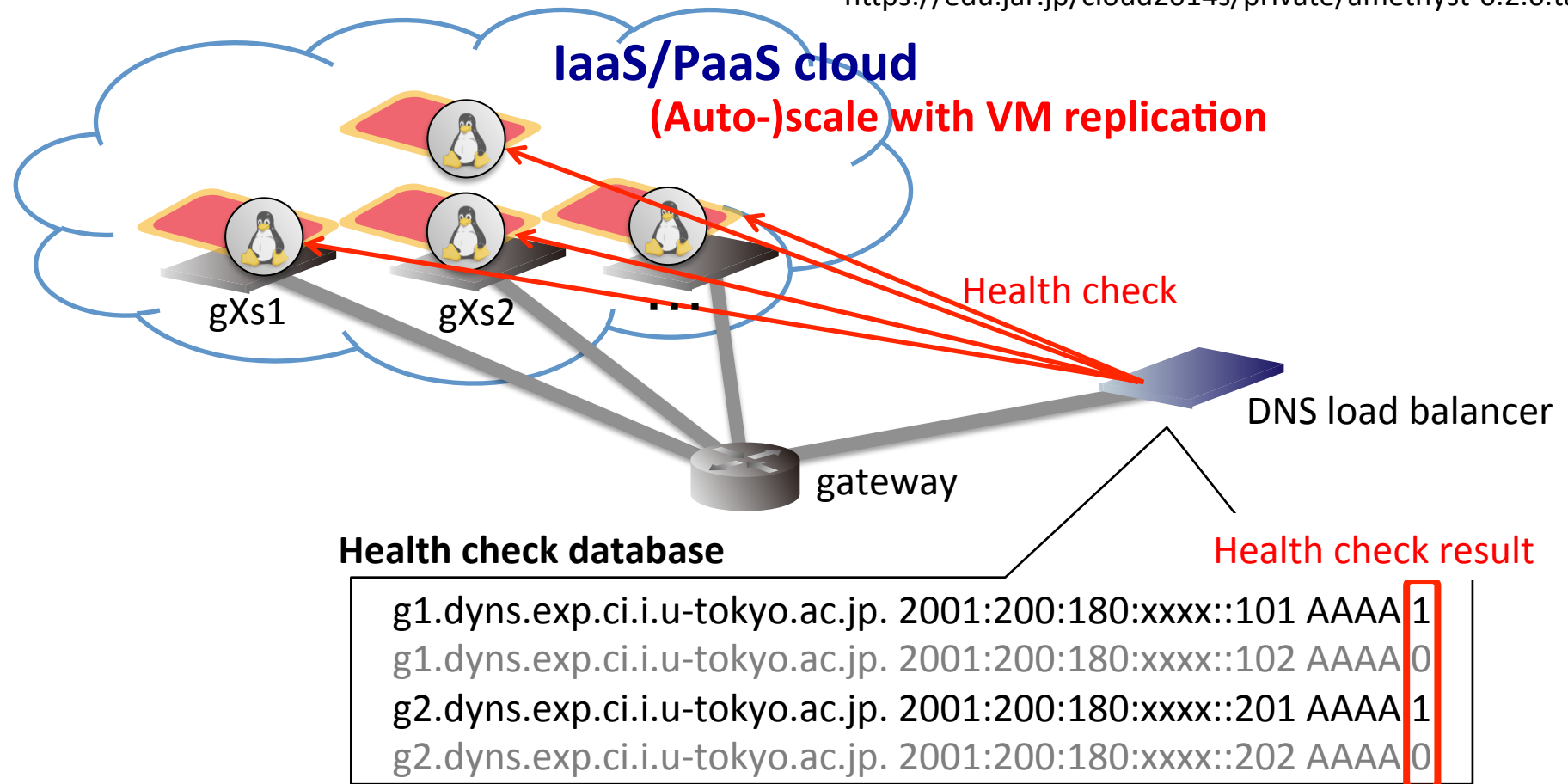
Adaptive thresholding

Day 2-4

- Develop your SaaS application
 - Keywords
 - Scale-out
 - HTML5
 - WebSocket
 - WebRTC
 - iOS
 - etc.

Example: DNS-based Load Balancing

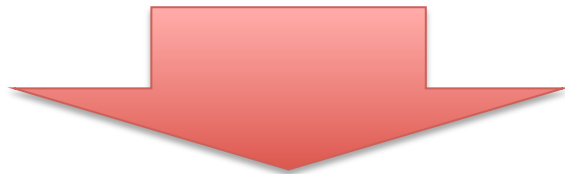
<https://edu.jar.jp/cloud2014s/private/amethyst-0.2.0.tar.gz>



For scaling out and down: Need to add *enable flag* or a function to add/delete records (N.B., Health check and TTL take some minutes to disable the record.)

Example: DNS-based Load Balancing

```
gx.dyns.exp.ci.i.u-tokyo.ac.jp. 2001:200:0:1cde:5054:ff:feff:a01 AAAA 1
gx.dyns.exp.ci.i.u-tokyo.ac.jp. 2001:200:0:1cde:5054:ff:feff:a02 AAAA 1
gx.dyns.exp.ci.i.u-tokyo.ac.jp. 2001:200:0:1cde:5054:ff:feff:a03 AAAA 1
gx.dyns.exp.ci.i.u-tokyo.ac.jp. 2001:200:0:1cde:5054:ff:feff:a04 AAAA 1
```



**Answer at most 3 of active records
(random order $\hat{=}$ DNS round robin)**

```
$ dig gx.dyns.exp.ci.i.u-tokyo.ac.jp AAAA +short
2001:200::1cde:5054:ff:feff:a04
2001:200::1cde:5054:ff:feff:a01
2001:200::1cde:5054:ff:feff:a02
```

How to use amethyst (DNS load balancer)

Download <https://edu.jar.jp/cloud2014s/private/amethyst-0.2.0.tar.gz>

```
$ tar -zxvf amethyst-0.2.0.tar.gz  
$ cd amethyst-0.2.0
```

Compile the package

```
$ ./configure  
$ make
```

Launch the daemon

```
$ sudo ./amethyst dnssd start
```

How to use amethyst (DNS load balancer)

Configure the server

```
$ vi conf/amethyst.conf
## Laste changed: 2011-01-26 xx:xx:xx JST
version 1.0.0;
dns {
    directory  "/home/yourname/amethyst-0.2.0";
    listen {
        *;
    };
    logfile    "/var/log/atomos.log";
    timeout 1.0;
    zone dyns.exp.ci.i.u-tokyo.ac.jp. {
        type master;
        file "to be implemented";
        selector {
            type cgi;
            script "cgi/dyns.cgi";
        };
    };
    default-selector {
        type cgi;
        script "cgi/dyns.cgi";
    };
};
```

Replace the zone with appropriate one.

How to use amethyst (DNS load balancer)

Setup the script

File: <https://edu.jar.jp/cloud2014s/private/dyns.cgi.txt>

```
#!/usr/bin/env python
import os
import sys
import csv
import random
import re

dbfile = "/home/yourname/amethyst-0.2.0/health.db"

"""
Print normal record
"""
def print_rec(qname, rtype, rcls, ttl, ans):
    print "%s %s %s %u %s" % (qname, rtype, rcls, ttl, ans)

"""
Print SOA record
"""
def print_soa(qname, rtype, rcls, ttl, mname, rname, serial, refresh, retry, \
              expire, minimum):
    print "%s %s %s %u %s %s %u %u %u %u" \
          % (qname, rtype, rcls, ttl, mname, rname, serial, refresh, retry, \
             expire, minimum)

"""
Print empty line to go to next section
"""
def next_section():
    print ""

"""
Main routine
"""
def start():
    """
    Check the arguments
    """
    if len(sys.argv) != 4:
        sys.exit(os.EX_USAGE)
    qname = sys.argv[1]
    qtype = sys.argv[2]
    qcls = sys.argv[3]
    ttl = 10
    if os.getenv("REQUEST_ZONE"):
        zone = os.getenv("REQUEST_ZONE")
    else:
        zone = "."
    if os.getenv("REMOTE_ADDR"):
        srcns = os.getenv("REMOTE_ADDR")
    else:
        srcns = None

    if qname == zone and qtype == "NS":
        nsname = "dyns1.exp.ci.i.u-tokyo.ac.jp."
        print_rec(qname, qtype, "IN", 3600, nsname)
        next_section()
        ## No NS
        next_section()

```

```

        ## add
        print_rec(nsname, "A", "IN", 3600, "192.0.2.1")
    next_section()
    sys.exit(0)
elif qname == zone and qtype == "SOA":
    ## Print SOA
    print_soa(qname, "SOA", "IN", 3600, "dyns1.exp.ci.i.u-tokyo.ac.jp.", \
              "postmaster.ci.i.u-tokyo.ac.jp.", \
              1, 3600, 3600, 604800, 60)
    next_section()
    next_section()
    next_section()
    sys.exit(0)

have = False
candidate = []
for ln in open(dbfile):
    [hostname, ipaddr, ctype, flag] = ln.split()
    if hostname == qname:
        have = True
        if flag=="1" and qtype==ctype:
            candidate.append([hostname, ipaddr, ctype, flag])

if len(candidate) == 0 and not have:
    sys.exit(os.EX_USAGE)

random.shuffle(candidate)
num = 0
for x in candidate:
    print_rec(x[0], x[2], "IN", ttl, x[1])
    num += 1
    if num > 2:
        break

    next_section()
    ## No NS
    next_section()
    ## No add
    next_section()

if __name__ == "__main__":
    ## Initialize the random seed
    random.seed(None)
    ## Go into the main routing
    start()

sys.exit(0)

```

Replace NS and records with appropriate ones.

How to use amethyst (DNS load balancer)

Health check script

```
import socket

dbfile = "/home/yourname/amethyst-0.2.0/health.db"

lst = []
for ln in open(dbfile):
    [hostname, ipaddr, ctype, flag] = ln.split()
    lst.append([hostname, ipaddr, ctype, flag]);

for i in xrange(len(lst)):
    if lst[i][2] == "AAAA":
        s = socket.socket(socket.AF_INET6, socket.SOCK_STREAM, 0)
    else:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0)
    s.settimeout(0.5)
    lst[i][3] = 1
    try:
        s.connect((lst[i][1], 80))
        s.close()
    except socket.error, e:
        lst[i][3] = 0

f = open(dbfile, "w")
for arr in lst:
    s = "%s %s %s %d\n" % tuple(arr)
    f.write(s)
f.close()
```

How to use amethyst (DNS load balancer)

Setup the database /home/yourname/amethyst-0.2.0/health.db
(NOTE: this file will not be locked...)

```
g0.dyns.exp.ci.i.u-tokyo.ac.jp. 192.0.2.1 A 0  
g0.dyns.exp.ci.i.u-tokyo.ac.jp. 192.0.2.2 A 0  
g0.dyns.exp.ci.i.u-tokyo.ac.jp. 192.0.2.3 A 0  
g0.dyns.exp.ci.i.u-tokyo.ac.jp. 192.0.2.4 A 0
```

Replace records with appropriate ones.

Day 1: Prepared virtual machine template

- Installed packages
 - apache, libopencv-dev, libcv-dev, libcvaux-dev, libhighgui-dev
- Installed one HTML5 front view
 - `/var/www/upload.html`
 - Supporting only Safari and Google Chrome (for simplicity)
- Installed two sample image processing programs
 - `/var/www/facerecog.cgi` : Face recognition
 - Source code: `/home/cloud/facerecog.c`
 - Config XML: `/var/haarcascade_frontalface_default.xml`
 - `/var/www/adaptive.cgi` : Adaptive thresholding algorithm (gaussian)
 - Source code: `/home/cloud/adaptive.c`
- Setup serial console
 - Added the following line to `/etc/inittab`
`s0:2345:respawn:/sbin/getty -L ttyS0 9600 vt100`
- Some magic in `/etc/rc.local`
 - just configuring hostname...