The Apache Olio Project

Shanti Subramanyam
Akara Sucharitakul
Performance & Applications Engineering
Sun Microsystems, Inc.

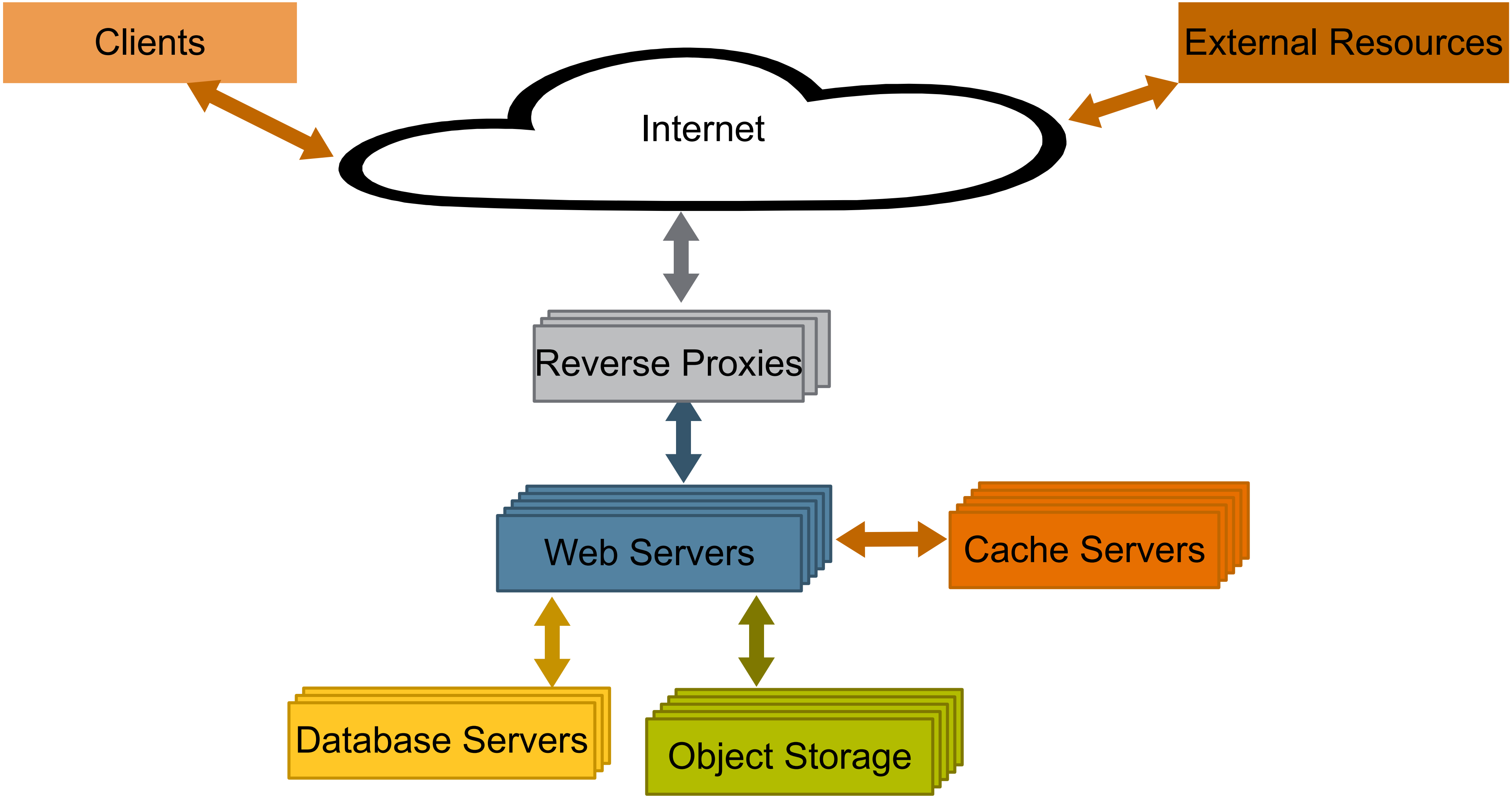
Agenda

- Background
- Application & Infrastructure
- Workload & Scaling
- What's Next?

What is Olio ?

- Reference Architecture to evaluate web2.0 technologies
- Sample web2.0 application
  - 3 implementations – PHP, Java EE and Ruby on Rails (RoR)
- Can be used to :
  - evaluate the differences in the various languages/frameworks for RoR and PHP
  - evaluate the infrastructure technologies for each implementation
  - compare the performance of the various technologies
- http://incubator.apache.org/olio

Typical Web Deployment Architecture



Why Olio ?

- Current web workloads...
  - Do not test architectures common in large scale web sites (AMP, unstructured data, memcached etc.)
  - Do not test web characteristics of modern web interfaces, i.e. Ajax, JavaScript, CSS

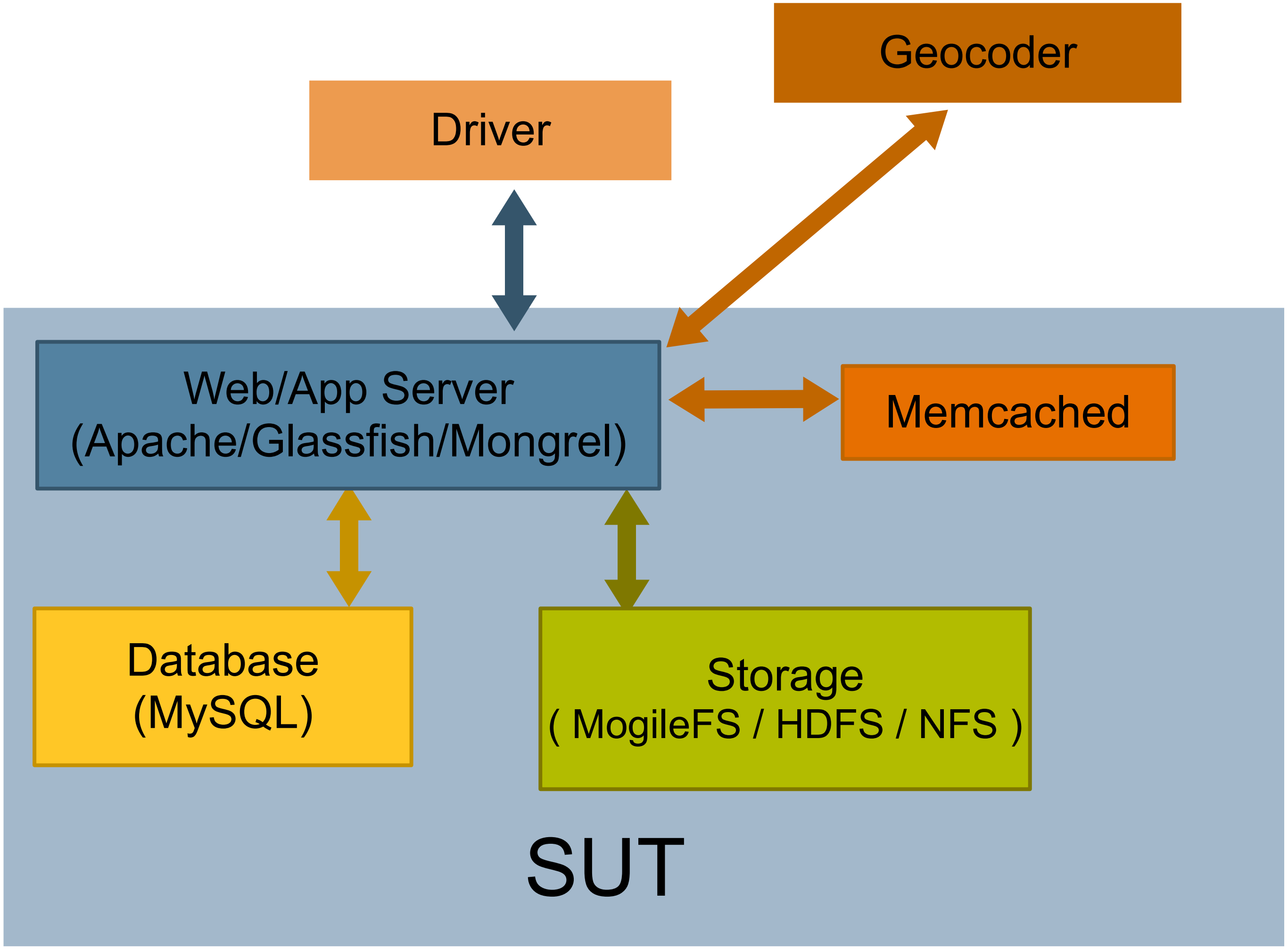
Olio Goals

- Showcase components used in social web sites
  - e.g. distributed object storage, memcache
- Showcase functionality commonly used by those sites
  - Ajax
  - Tagging
  - Comments & Ratings
  - Mashup
  - Unstructured data access
- Simulate a high read-to-write ratio common to these sites

Agenda

- Background
- Application & Infrastructure
- Workload & Scaling
- What's Next?

Olio Architecture



The Application

- A Social Event Calendar
  - Allows posting, sharing, tagging/searching, and commenting on social events
  - Events and persons have images, thumbnails
  - Persons have friends
  - Events have event literature (pdf file)
  - You can browse events by date (Ajax)
  - You can sign up to attend events (Ajax)
  - Provides details about the individual events
  - Provides event feeds

Rich Web Clients

- Heavy use of CSS
  - e.g. mouse-over popups showing event detail
- JavaScript/Ajax client components
  - e.g. date picker, event attendee list

Application Functionality

- Server side mashup with Geocoder
  - Geocoder emulated for test environment
- Client side mashup
  - Yahoo maps
- Tagging
  - Tag Clouds
  - Tag Search

Application Functionality (Cont'd)

- Writes/Image uploads (~6% of transactions)
  - Event creation
  - User registration and user profile
  - Comments & Ratings
  - Subscribing to an event
- Data caching using Memcached
  - Generated fragments of home page
  - Tag cloud (planned)
- Feed generation

Application Environments

- Three implementations
  - PHP
    - pdo_mysql, UnixODBC
    - NFS, mogileFS
    - script.aculo.us
  - Ruby on Rails
    - Rails plugins
    - FreeImage
    - NFS
    - script.aculo.us
  - Java
    - Servlets
    - JPA
    - JDBC
    - NFS, HDFS
    - Dojo

Agenda

- Background
- Application & Infrastructure
- Workload & Scaling
- What's Next?

Workload Description

- Workload implemented using Faban
  - > Open source benchmark development framework
- Some parameters :
  - > Scale – number of concurrent users
  - > Metric – ops/sec (# of operations completed/sec)
  - > Operation Mix – Matrix Mix
    - >Finite Markov Chain (aka. Stochastic Matrix)
  - > Inter-arrival times – negative exponential with mean of 5 seconds for all operations

Data

- Loaded Users = 100x Concurrent Users
- Tags start scaling linear at ~25% loaded users but gradually tops off at 5,000
  - cumulative half logistic distribution
- Loaded events = 3x of Tags
- Irregular tag assignment
  - Tag cloud, few large tags early on
- Images – thumbnail and full image for each user and each event
- Event Literature (pdf)

Sample Database Scale
@1000 Concurrent Users

Table	Rows	Kbytes Incl. Indexes
ADDRESS	100,000	8,350
COMMENTS_RATING	693,300	381,850
PERSON	100,000	147,980
PERSON_PERSON	1,500,000	63,320
PERSON_SOCIALEVENT	3,813,150	170,410
SOCIALEVENT	69,330	33,480
SOCIALEVENTTAG	23,110	1,380
SOCIALEVENTTAG_SOCIALEVENT	277,320	10,740
Total	**6,576,210**	**817,510**

Sample Image & Binary File Scale
@1000 Concurrent Users

Type	# of Images	Size (KB)	Replicas	Total (MB)
Person Image	100000	670	3	65,430
Person Thumbnail	100000	7	2	684
Event Image	69330	670	3	45,362
Event Thumbnail	69330	7	2	474
Event Literature	69330	130	3	8,802
Total(GB)				**117.92**

Workload Description

- 7 Operations
  - > Homepage
  - > Login
  - > Tag Search
  - > EventDetail (~8% include AddAttendee)
  - > PersonDetail
  - > AddPerson
  - > AddEvent
- Operation => one entire web page
  - > multiple HTTP requests to complete page

Simulating Browser Caches

- 40% of sessions are simulated with empty cache
  - Full cache => no static downloads
  - Empty cache => static downloads once and cached
- User randomly terminates session
- Images get parsed and loaded
- Images are cached for the length of the session

Current Status & Futures

- PHP and Rails versions in Apache repository
- Java EE version close to completion
- Continue improving application and data model to reflect real sites, but...
  - Keep it simple
  - Minimize resources needed to proof performance/scalability
- Solicit participation

Resources

- http://incubator.apache.org/olio
  - Additional resources listed on Olio site
- Olio Repository
  - https://svn.apache.org/repos/asf/incubator/olio/
- Faban – Load generator for Olio
  - http://faban.sunsource.net/
- http://glassfish.dev.java.net/
- http://lucene.apache.org/hadoop/
- http://developer.yahoo.com/maps/rest/V1/geocode.html

The Apache Olio Project

Shanti Subramanyam

Akara Sucharitakul

Performance & Applications Engineering

Sun Microsystems, Inc.

PHP Application Infrastructure

- Apache 2.2.x
- PHP 5
- MySQL 5
- MogileFS (http://www.danga.com/mogilefs)
- Yahoo maps (client side only, not part of workload)
- Geocoder (http://geocoder.yahoo.com)
  - Emulated as part of workload using Tomcat/servlet

Java Application Infrastructure

- Glassfish, Tomcat, Other Java Apps Servers
- MySQL
- Hadoop HDFS (http://lucene.apache.org/hdfs)
- Yahoo maps (client side only, not part of workload)
- Geocoder/Emulator (http://geocoder.yahoo.com)