Panel Discussion
“Benchmarking in the Web 2.0 Era”

Akara Sucharitakul
Performance & Applications Engineering
Sun Microsystems, Inc.

“

PAE Workload Experiences

- We touched almost all relevant workloads
- We were part of developing many important workloads...
  - TPC-B
  - Ecperf, SPECjAppServer2001, 2002, 2004, next
  - SPECweb99, 2005
- And many more custom workloads...

How about Akara?

- Macro-level benchmarking
- Hands dirty on several SPEC (web/java) benchmarks and web services benchmarks
- Project Faban – Open Source benchmarking harness and driver framework
  - > Allowing people to write new macro-workloads for their applications, easily
  - > 'Cause we got tired of doing it all ourselves

Web 2.0 Performance Requirements

- Extremely large user base, >10^6 users
- Reasonably fast response times, <1 sec
- Extremely high aggregate throughput, i.e. 10^9 page views/day

Web 2.0 Application Characteristics

- Massively horizontally scaled
- Data is read mostly → pseudo-static
  - Frequently accessed data pre-generated and cached
- Data access extremely non-uniform
- No single point of failure
- Redundancy implemented in software
- Frequently requires external data sources
  - Server (or client) side mash-ups

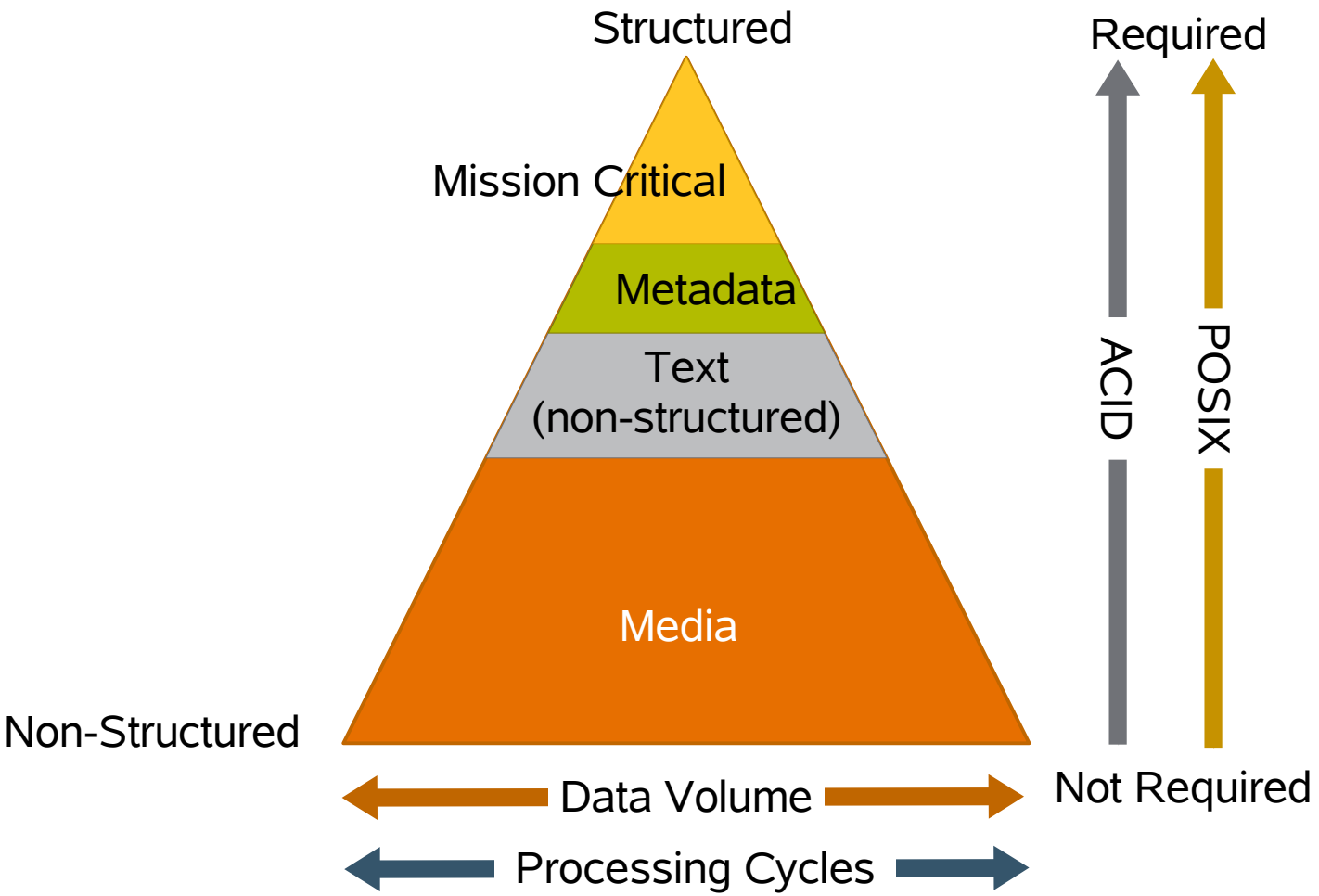
Web 2.0 Network Characteristics

- Wider variety of content and packet sizes

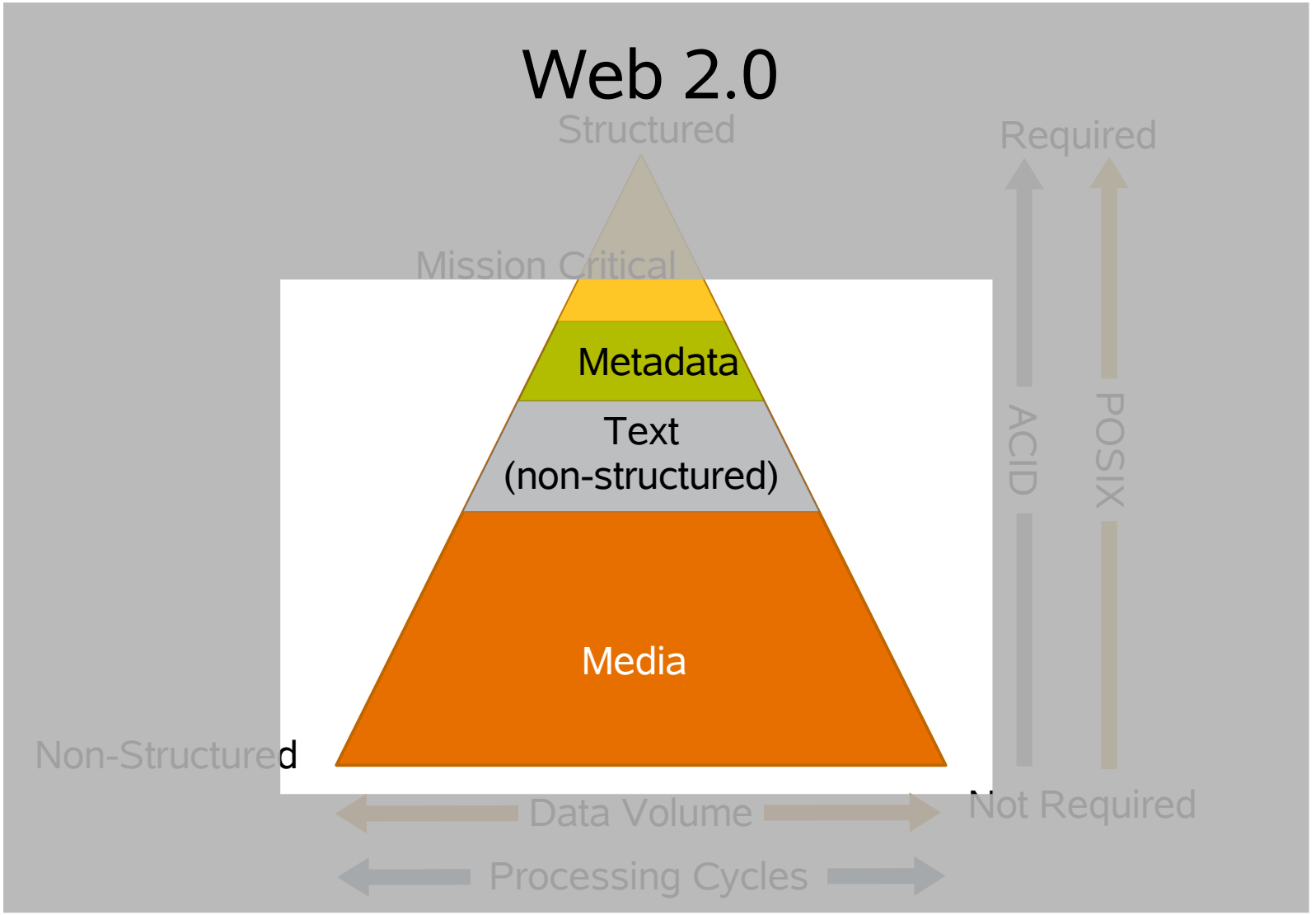
Small Ajax requests/updates	< 1KB
Ajax requests/updates	1 – 100KB
Pages with Ajax libraries	10KB – 5MB
Media and Downloads	>100KB

- Some media is time critical
- Lots is solved by effective buffering at client

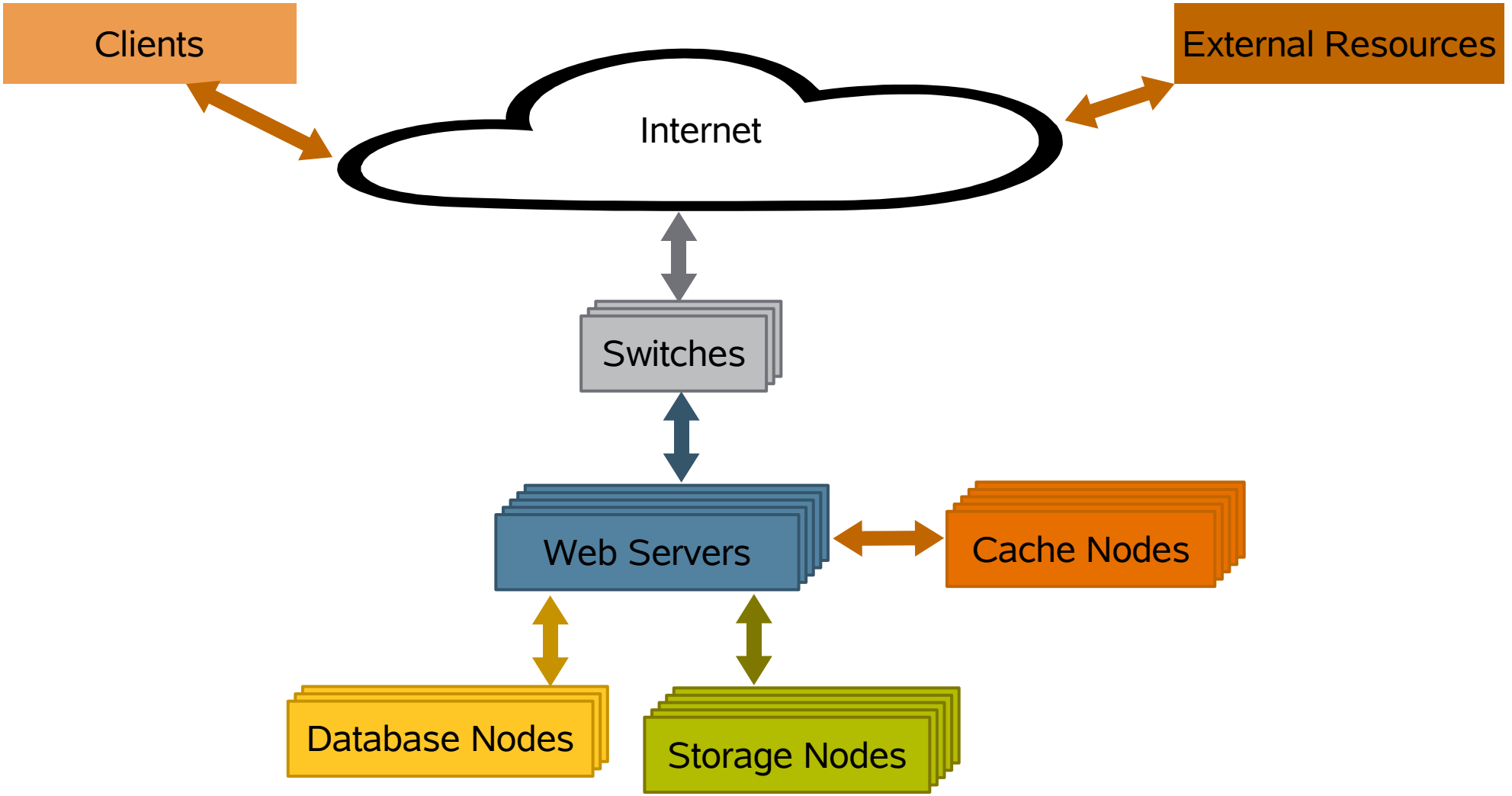
Web 2.0 Data Characteristics



Web 2.0 Data Characteristics



Scalable Web Deployment Architecture



Hardware Challenges

- Virtually limitless horizontal scaling, but... power challenges
- More power-efficient systems → same or lower number of systems for more bandwidth
- cores↑ memory bandwidth↑ network bandwidth↑ I/O bandwidth↑ IOPS↑ packet rates↑ single NIC
- Moore's law is on our side, although we might just abuse it

Software Challenges

- Common software (LAMP stack) developed for single-core, bad vertical scaling
- Virtualization - the short term lifesaver
  - Tools lag behind, increase management cost
  - Manage 10 boxes cheaper than 10 virtual systems
- Software infrastructures will catch up or applications will migrate to more scalable architectures
  - May take another generation of infrastructure
  - More efficient technology is available, but not widespread and perceived expensive, i.e. Java-based systems

Workloads, why yet another one?

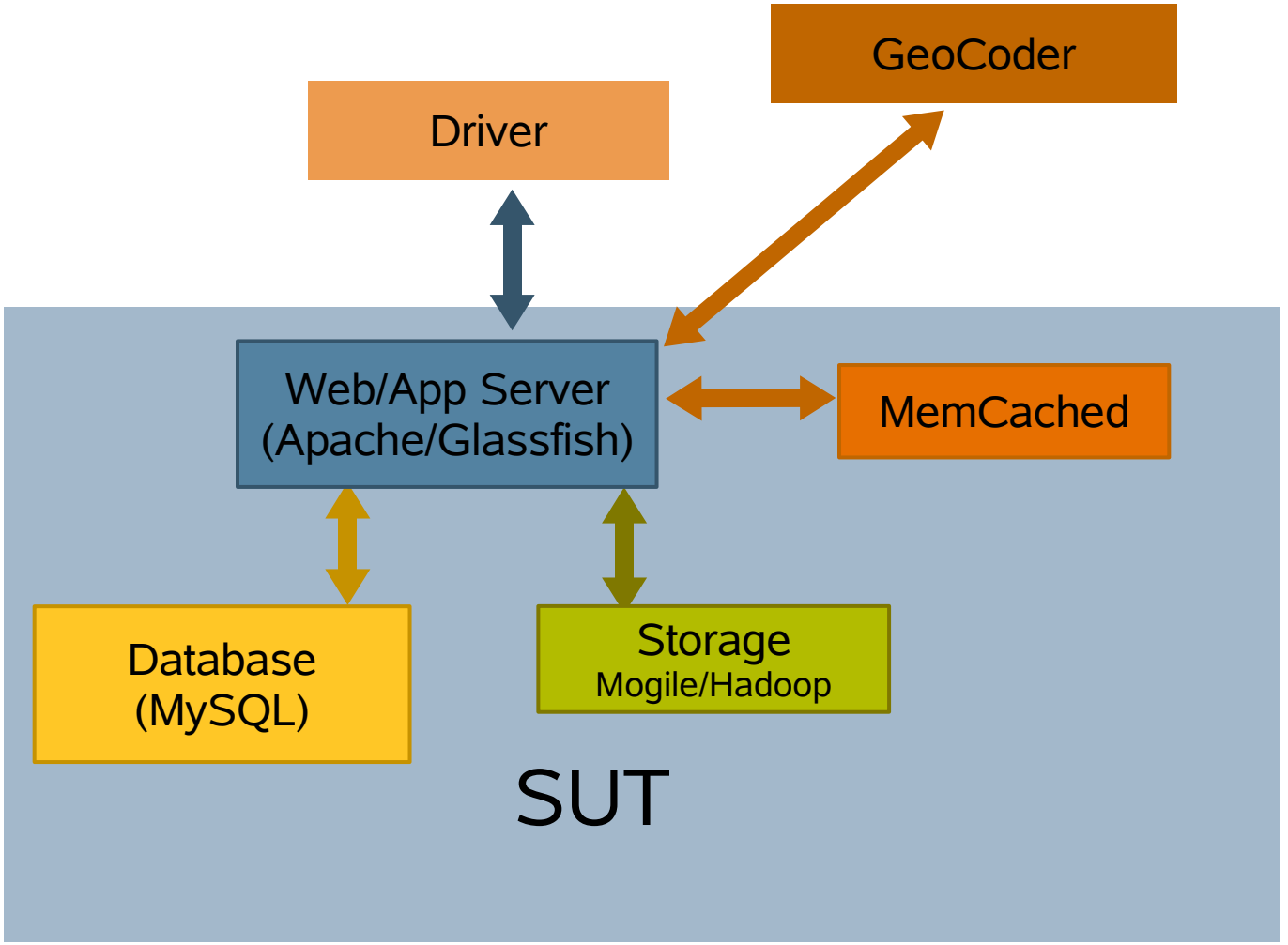
- No workload simulates the characteristics of modern web architectures
- Current web benchmarks...
  - Do not test architectures common in large scale web sites (AMP, memcached etc.)
  - Do not test web characteristics of modern web interfaces, i.e. Ajax, JavaScript, CSS
  - Provide peak throughput in highly optimized environments, unlike real web sites
- So we came up with Web20Bench (for the lack of a better name)

Web20Bench Goals

- Test components used in large scale social web sites
  - i.e. Memcache, distributed and redundant storage
- Simulate functionality commonly used by those sites
  - Ajax
  - Tagging
  - Comments & Ratings
  - Mashup
  - Media-heavy
- Simulate a high read-to-write ratio (95% to 5%) common to these sites

Benchmark Architecture

An attempt to scale down the test environment while still making sense



Web20Bench Status

- PHP version is up and running
- Java version following close behind
- Ruby version, future
- It's very new – we still need to iron out the wrinkles
- We learn what people do and add them to the benchmark as we know more
- And yes, we intend to make it publicly available

Interesting Links

- http://httpd.apache.org/
- http://tomcat.apache.org/
- http://www.danga.com/memcached/
- http://www.danga.com/mogilefs/
- http://glassfish.dev.java.net/
- http://lucene.apache.org/hadoop/
- http://developer.yahoo.com/maps/rest/V1/geocode.html
- http://faban.sunsource.net/

Panel Discussion
“Benchmarking in the Web 2.0 Era”

Akara Sucharitakul

http://faban.sunsource.net/