



Comparison of Script Characterization of web benchmarks

A presentation at SPECworkshop in Paderborn by the
members of osgweb group

Barry Arndt - IBM

Hansfried Block - Fujitsu

Dean Chandler - Intel

Stephan Gradek - Fujitsu

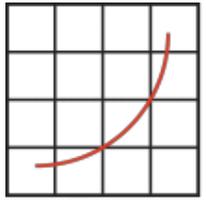
Rema Hariharan - AMD

Sean Wu - Oracle



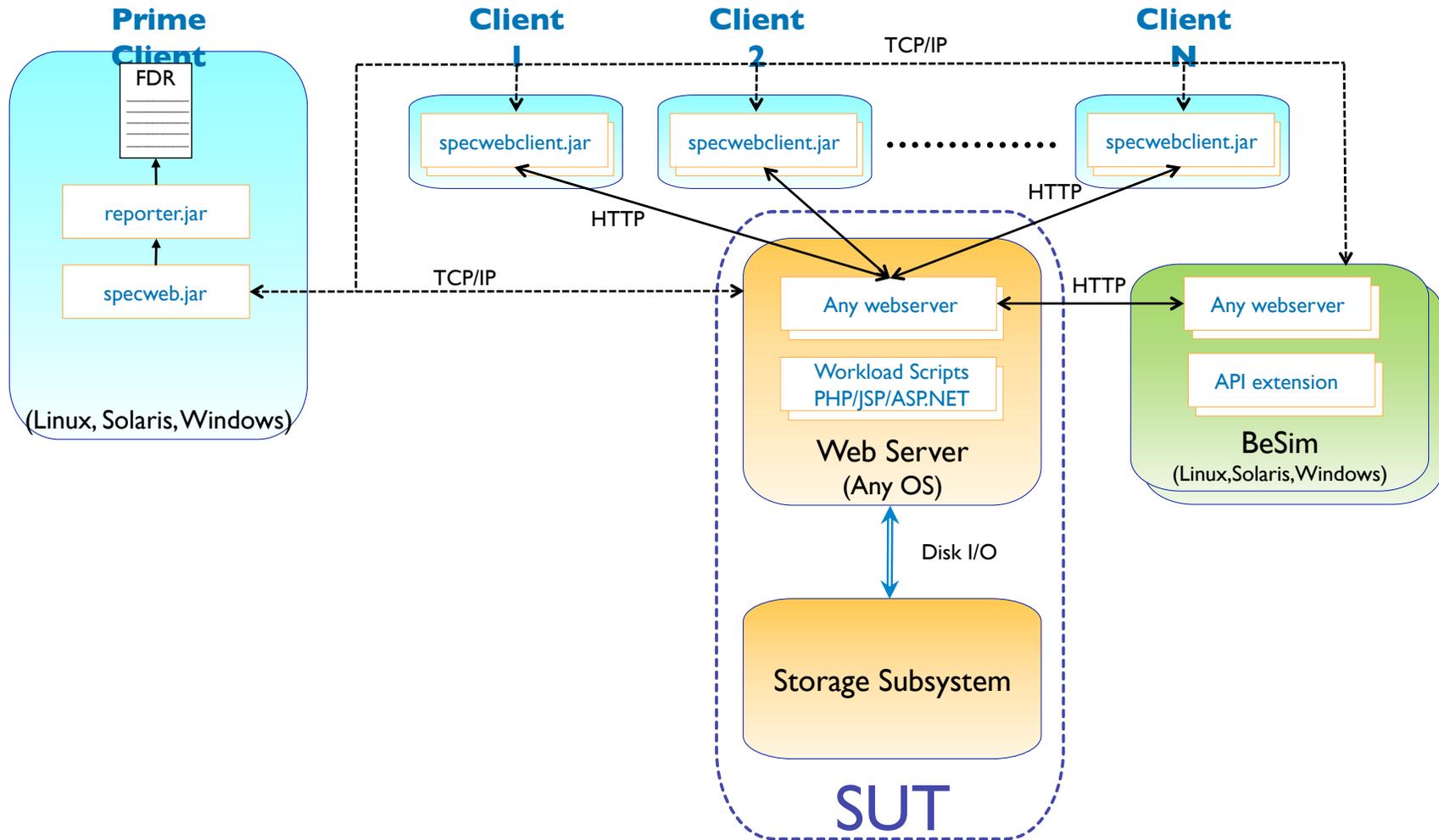
Overview

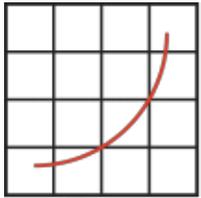
- The benchmark emulates web users accessing an application.
- Internet Banking, Ecommerce and Support download applications.
- Banking workload is 100% secure, Ecommerce workload is partly secure and Support is plain http workload.
- Applications run scripts in JSP/PHP/ASP
- How many user sessions that can be supported while meeting a pre-specified QOS?
- Web2009 also includes a power metric.



spec[®]

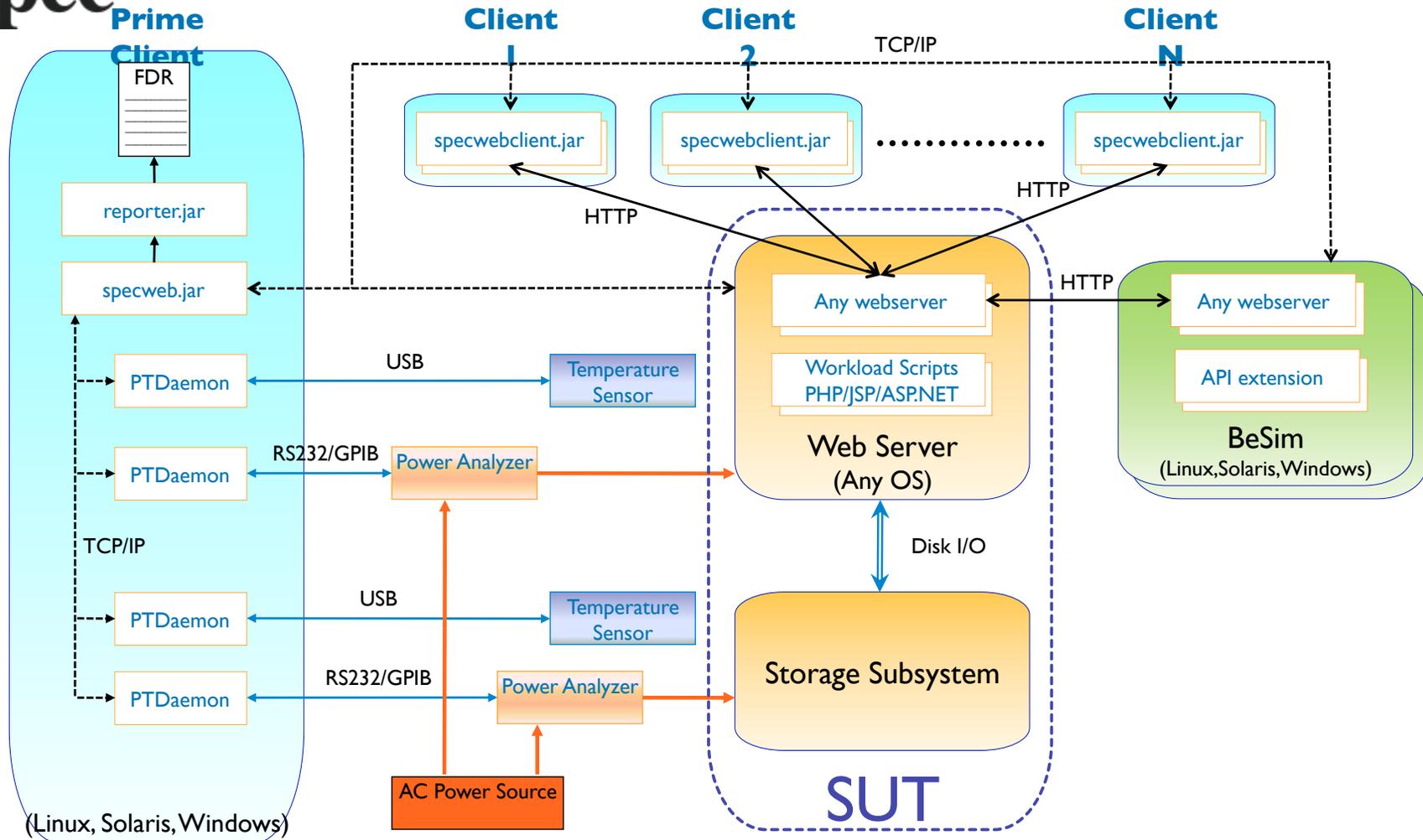
SPECweb2005

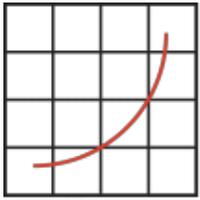




SPECweb2009

spec[®]





spec[®]

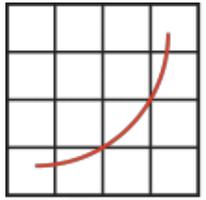
What is being characterized

- Performance of Scripts
- JSP, ASPX, PHP with SPECweb run on Linux and Windows
- Performance data running Olio (a web2.0 benchmark) on Solaris.
- All data collected with 2 processor systems
 - ❖ 8 cores
 - ❖ 1 Gb/s to Backend and 10 Gb/s to client
 - ❖ Local storage drives for data and logs
- Emulated 5000 user sessions



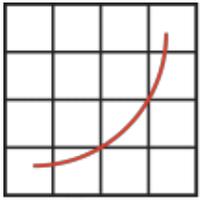
What we hope to achieve through this presentation

- Convince the audience about the performance differences between workloads and scripting methods used.
- Hint at the areas where software improvement might result in heavy performance improvements.
- Illustrate differences between web2.0 Olio based workloads and SPECweb workloads.



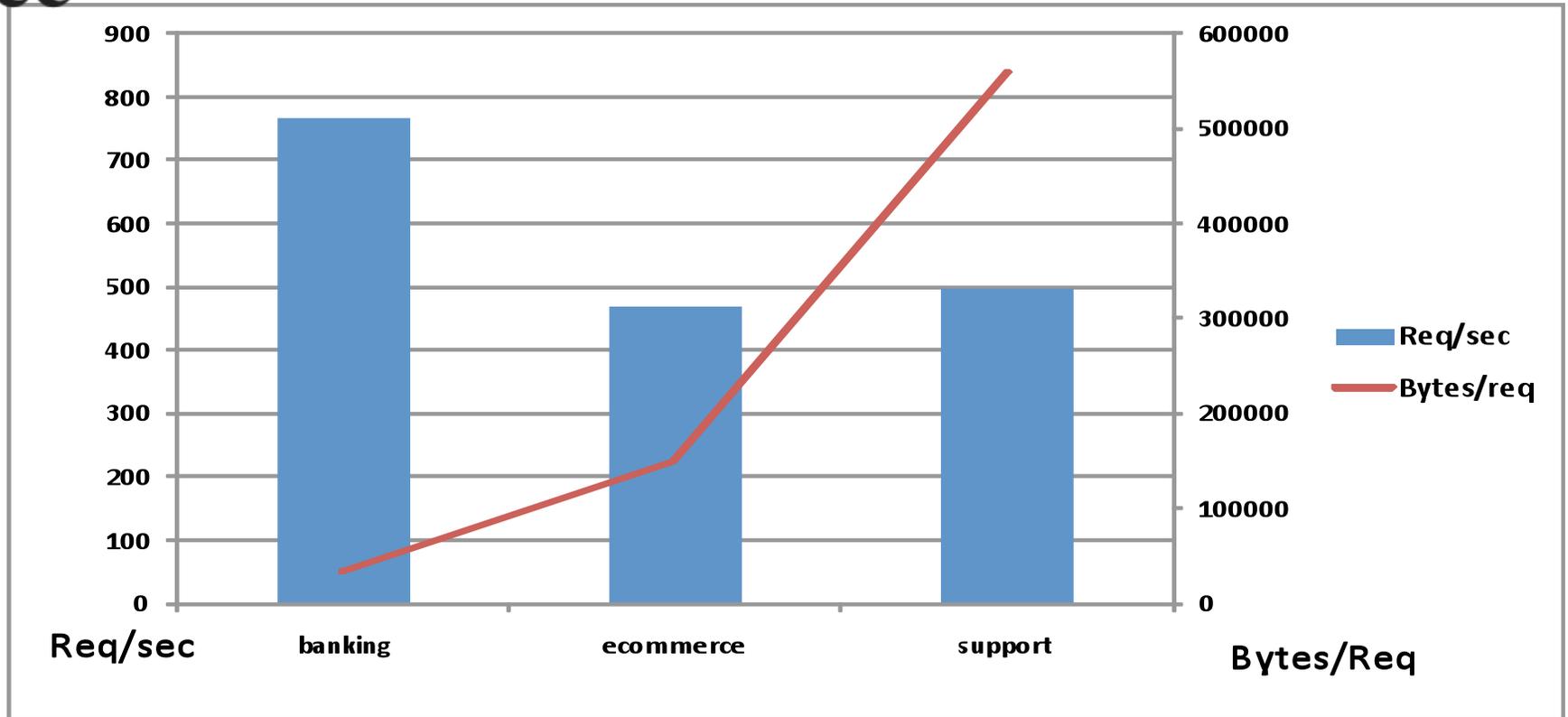
spec[®]

Platform independent and Script
independent characteristics

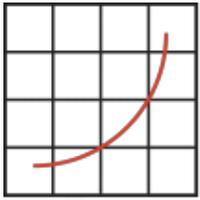


spec[®]

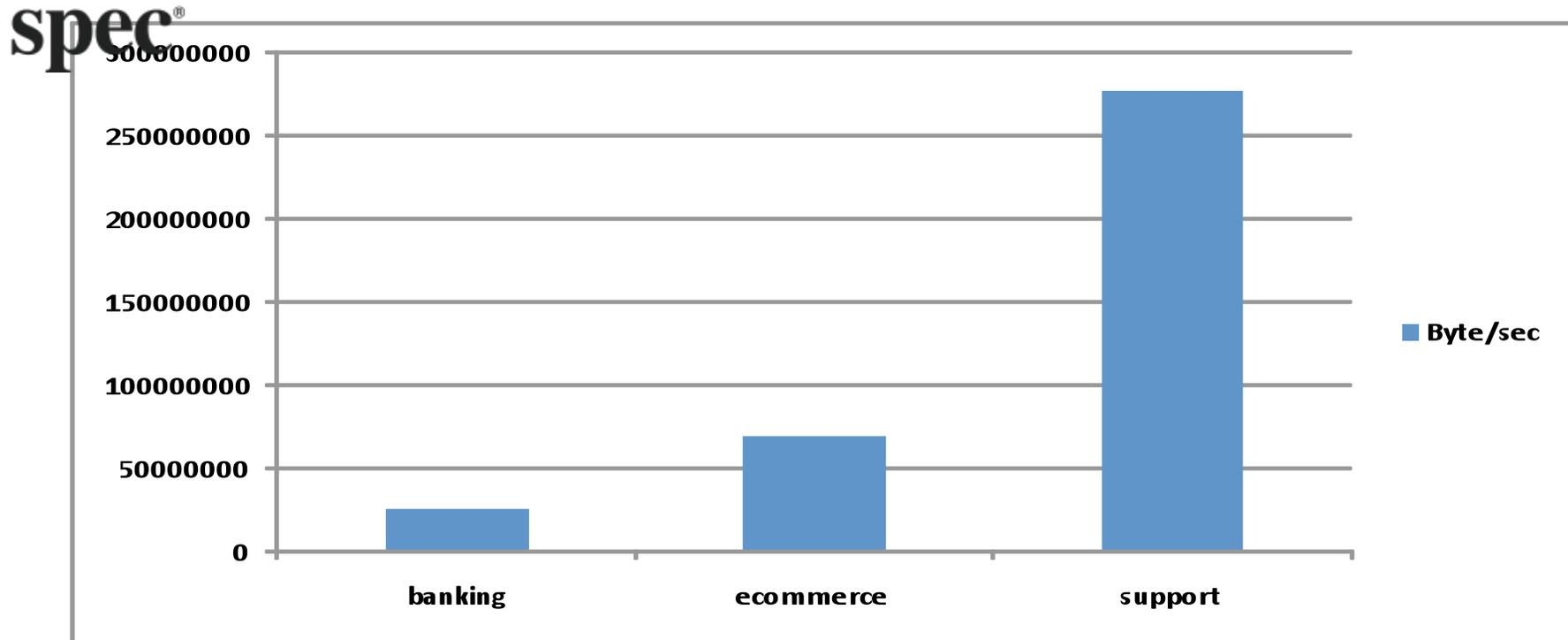
SPECweb Page requests



- Request rate consistent between script types and software stacks
- This is a constant load based on QOS level
- Banking has highest request rate but lowest overall bytes per request



Network Send Bytes/sec

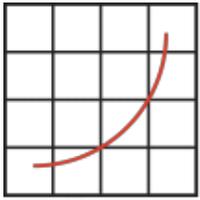


- Passing runs have constant send rate regardless of OS/script type
 - Banking 5058 bytes/session
 - Ecommerce 13908 bytes/session
 - Support 55490 bytes/session

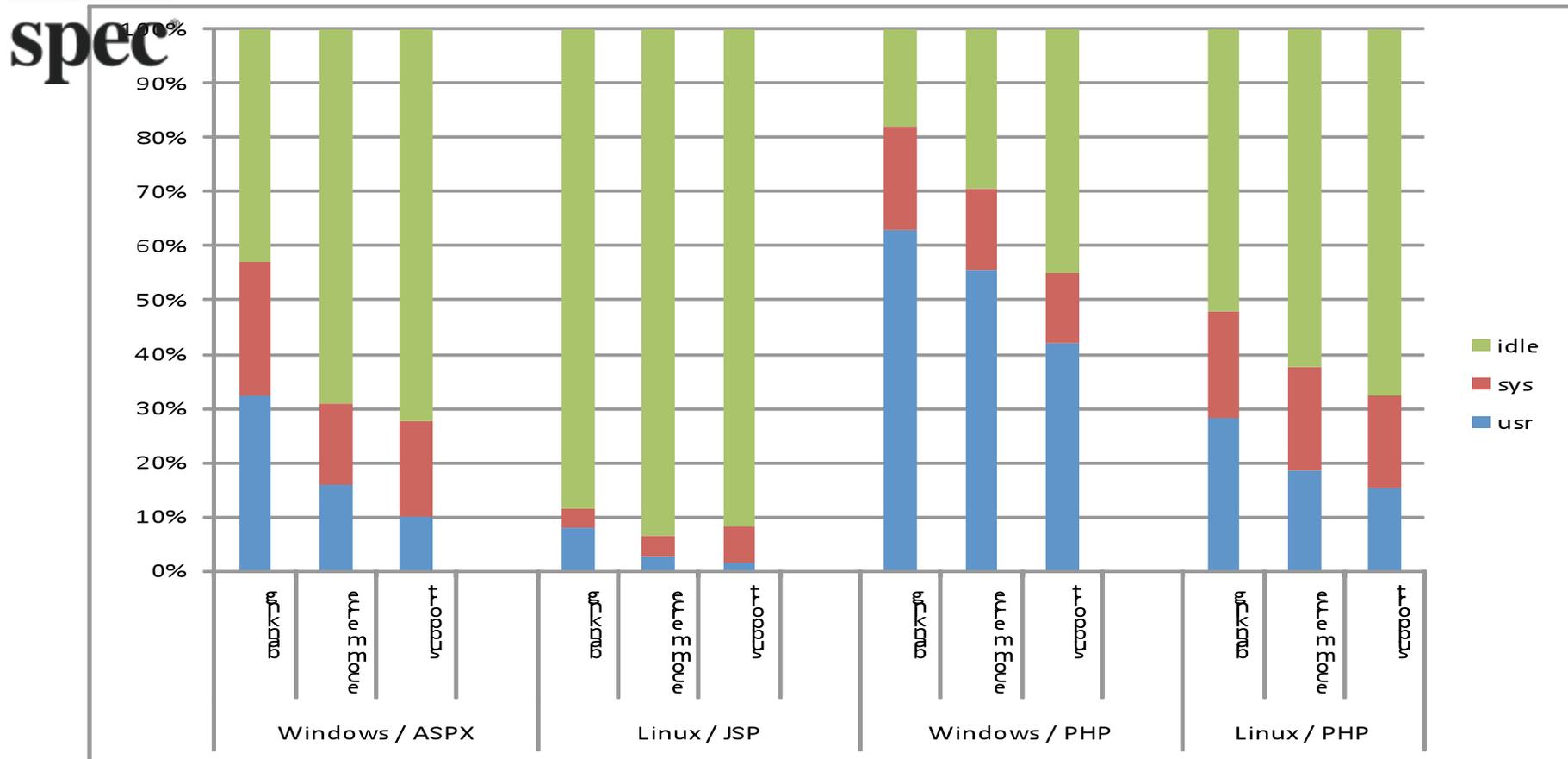


Script Dependent characteristics

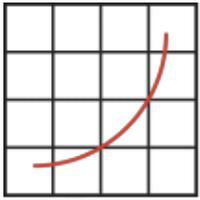
- CPU usage pattern
- Interrupts
- Context Switches
- DRAM usage
- Disk usage



CPU usage for various scripts

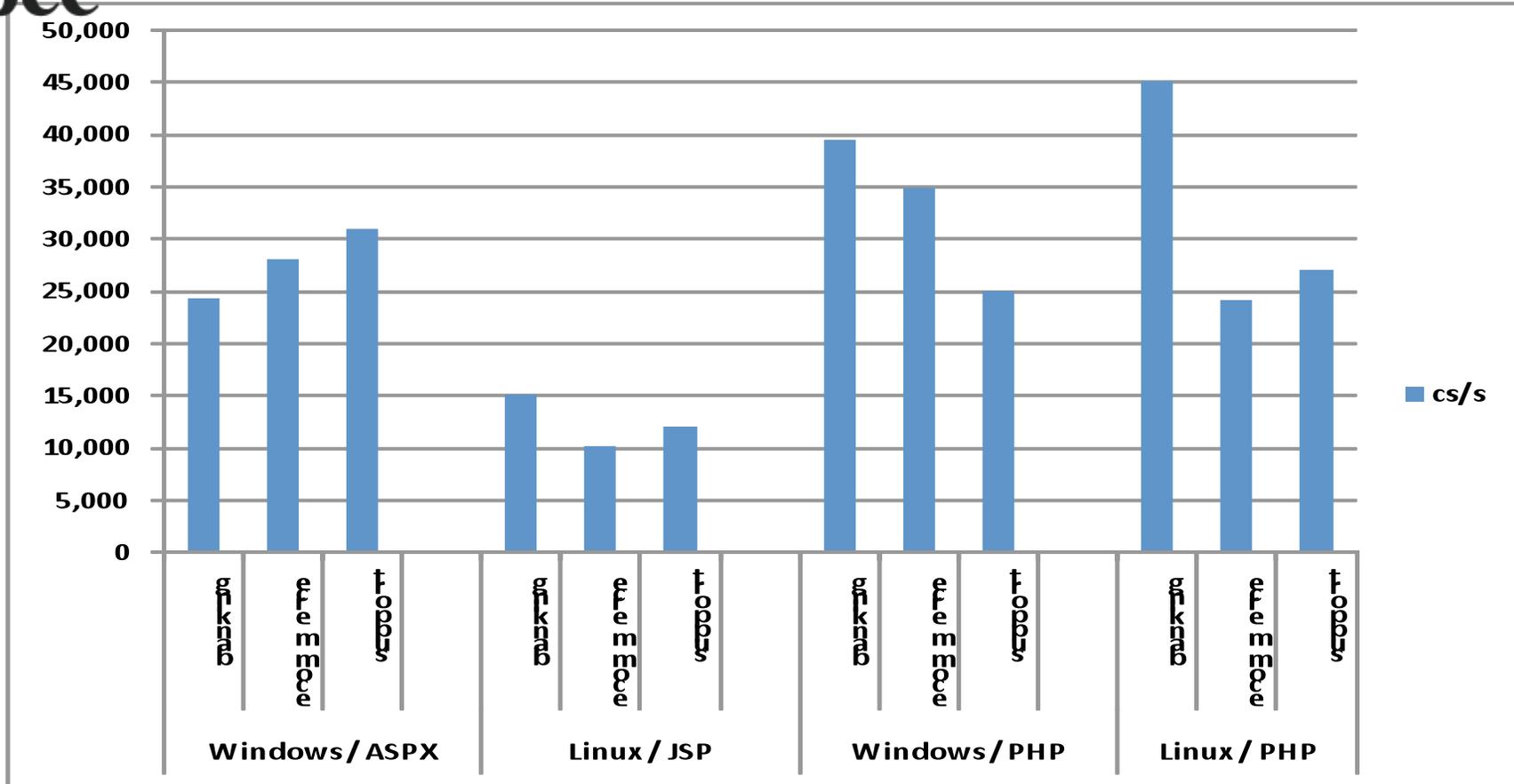


- Banking workload creates highest CPU utilization due to SSL + encryption/decryption
- JSP lowest CPU utilization due to best pre-compiled performance
- PHP highest CPU utilization due to requirement to compile each request

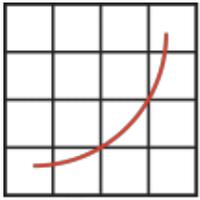


Thread Context Switches per Second

spec[®]

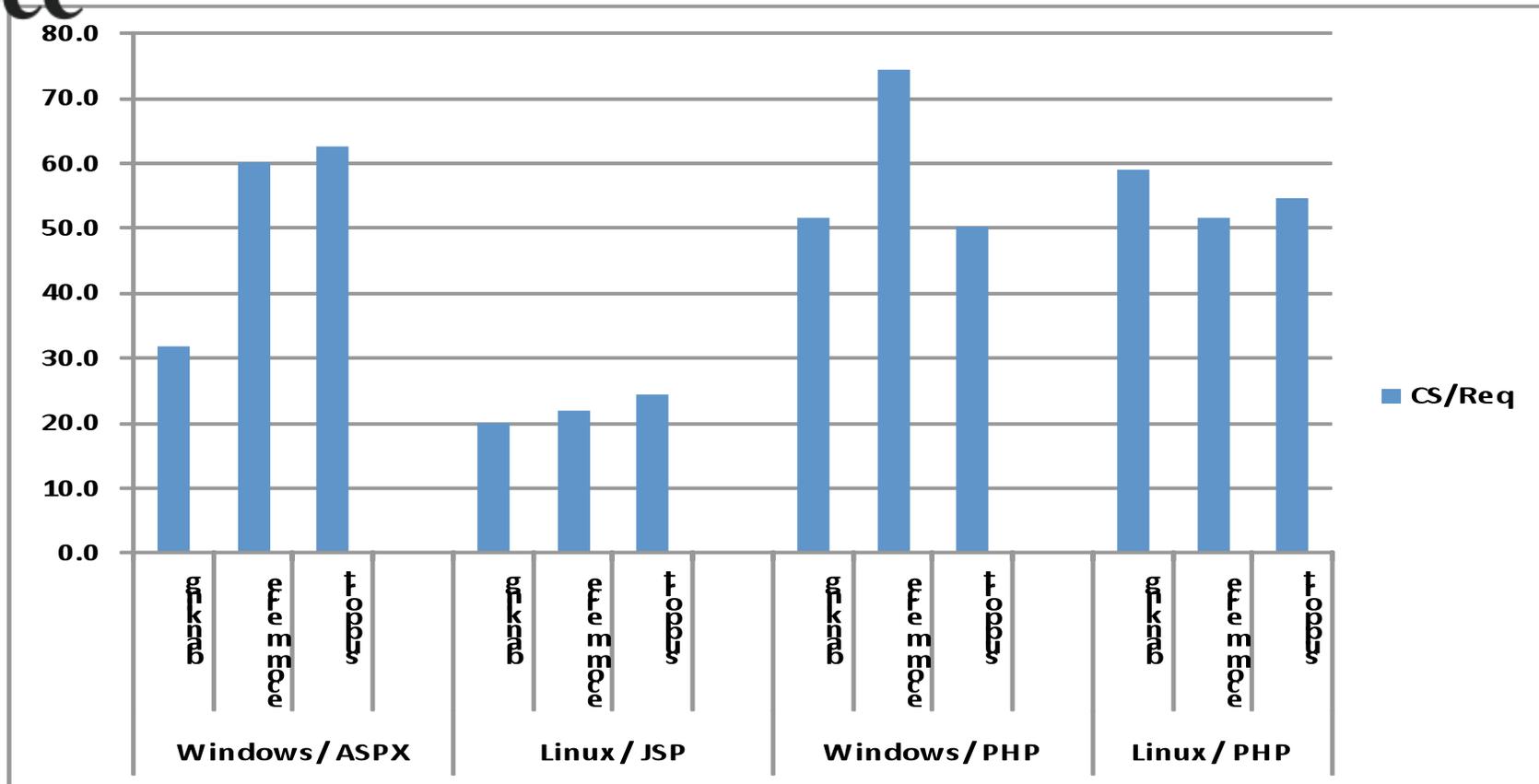


- Banking PHP causes highest switch rate due to secure transactions and script compilations
- Linux lowest due to optimized SSL connections

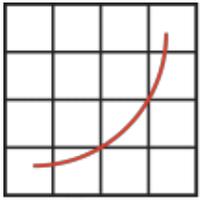


Thread Context Switches per Request

spec[®]

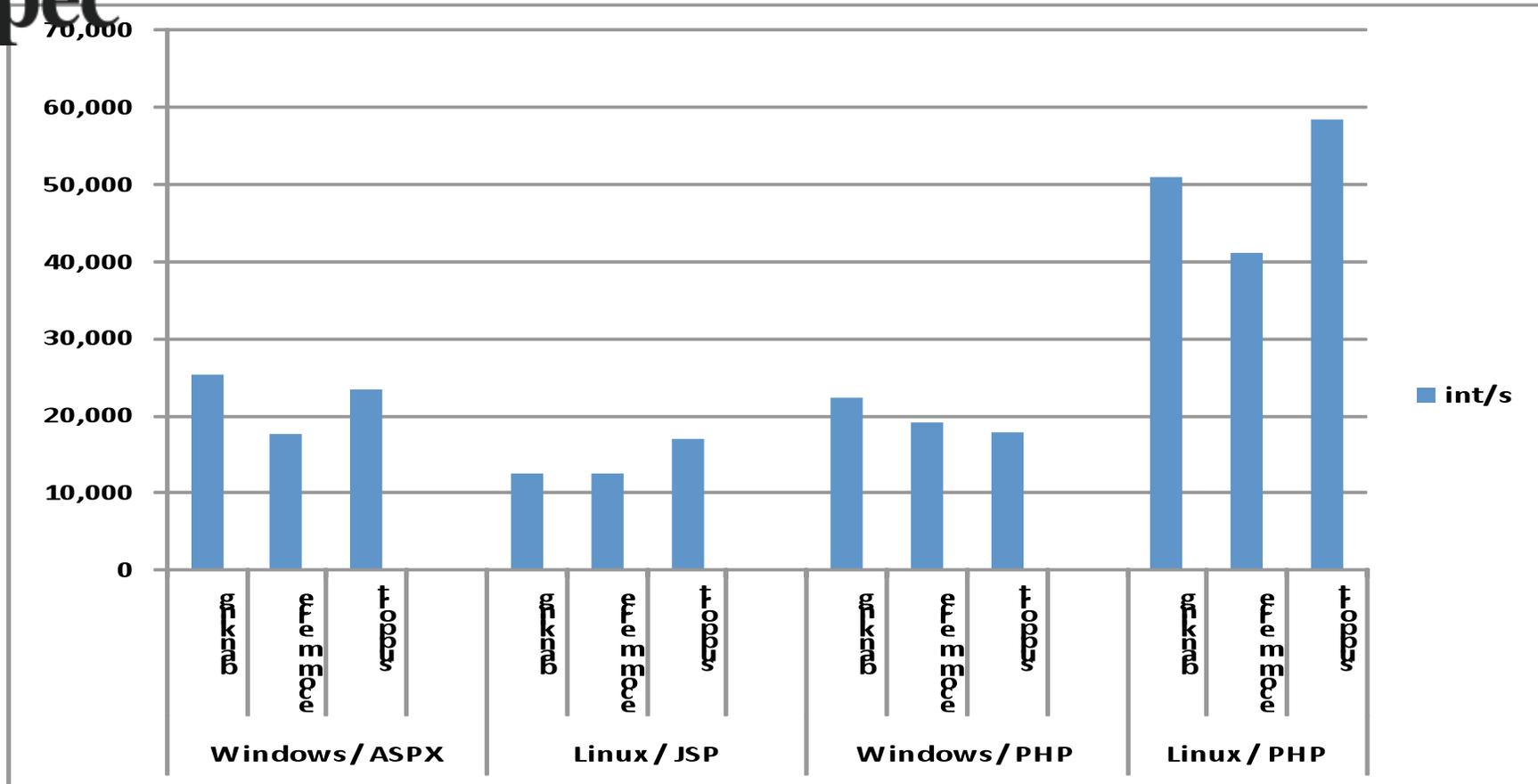


- Ecommerce highest CS/req due to large amount of backend processing
- Banking CS/req is lower due to high number of requests and lowest network bytes per request



CPU Interrupts per Second

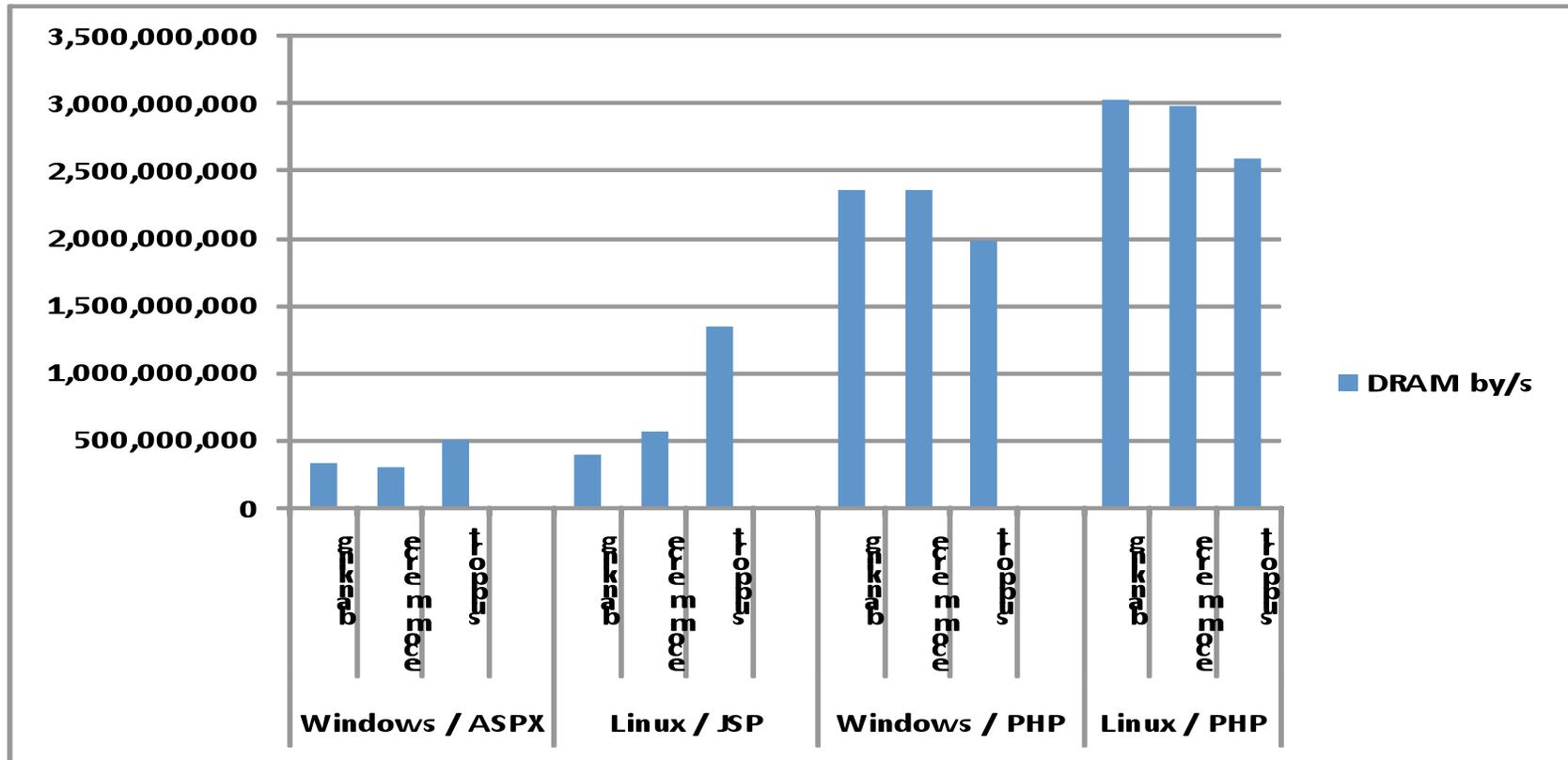
spec[®]



- Linux PHP workload has highest interrupt rate due to network I/O issues
- Linux JSP handles best for I/O and SSL handshake due optimized SSL stack



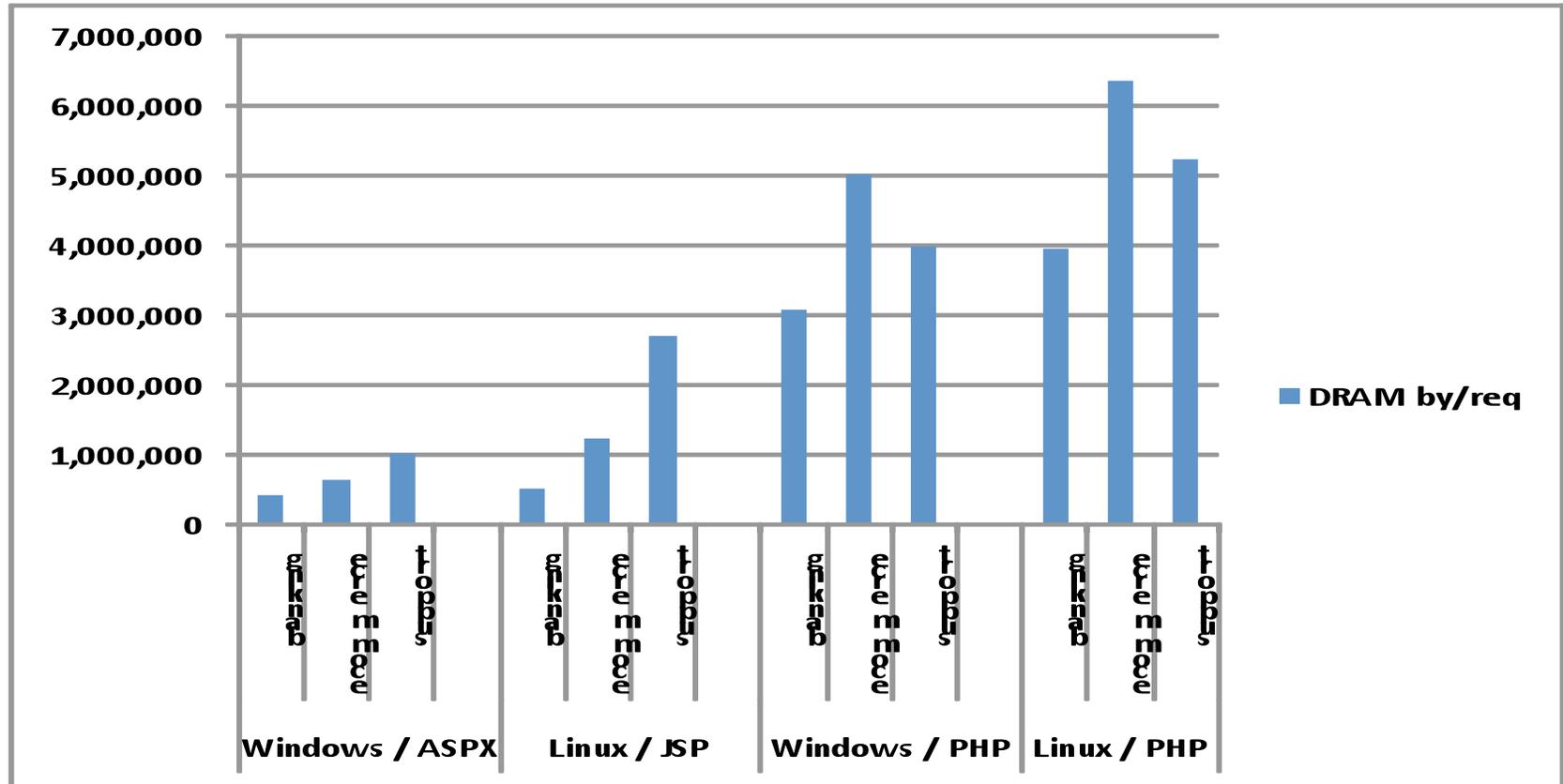
DRAM Bytes per Second



- Windows ASPX best memory usage better alignment to page size
- PHP scripts require higher memory bandwidth due to script compilation



DRAM Bytes per Request



- Windows ASPX best memory usage better alignment to page size

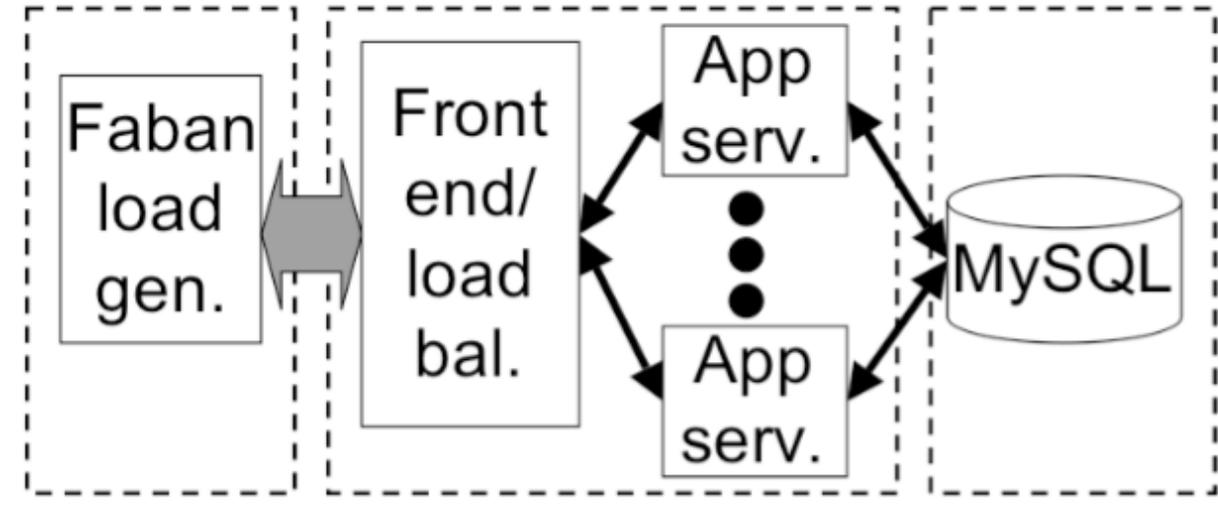


Olio

- This is a web2.0/cloud benchmark created by Sun/Oracle and UC Berkeley.
- Based on social event calendar application.
- Uses memcached, backend dB (mysql) and PHP scripts.

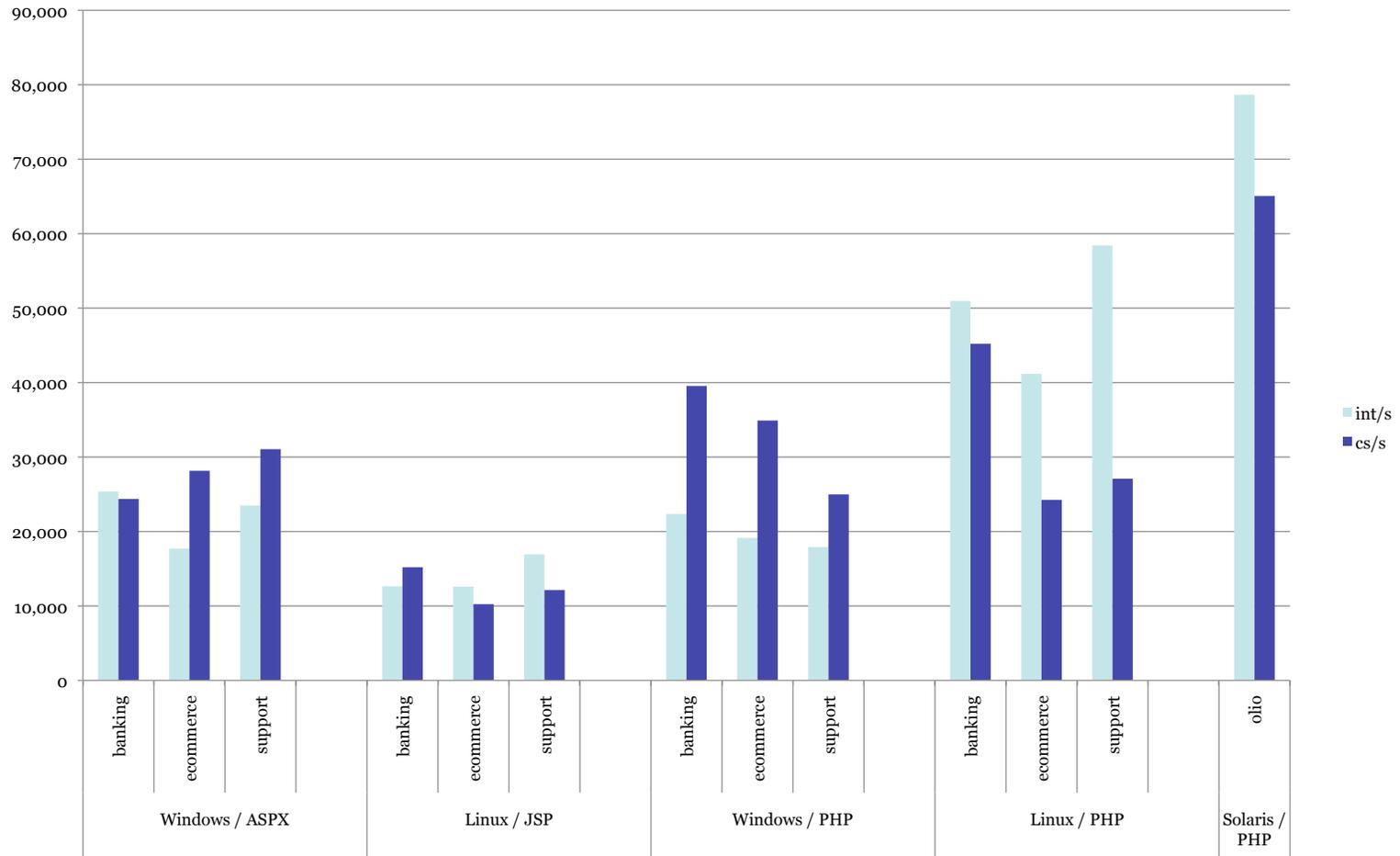


Olio Layout



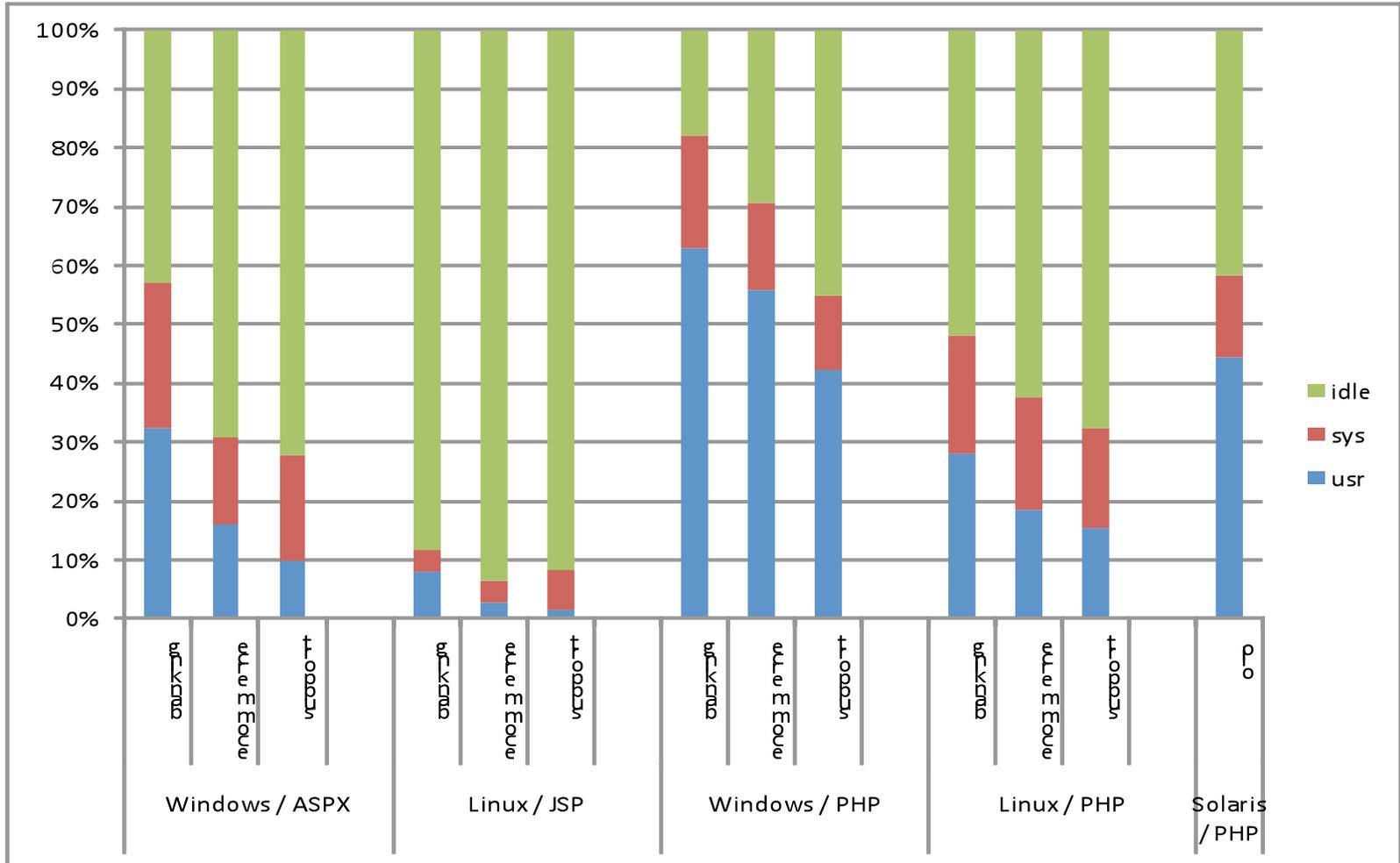


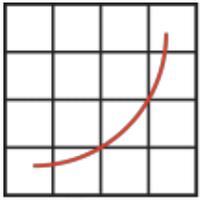
Comparing Interrupts for Olio with SPECweb





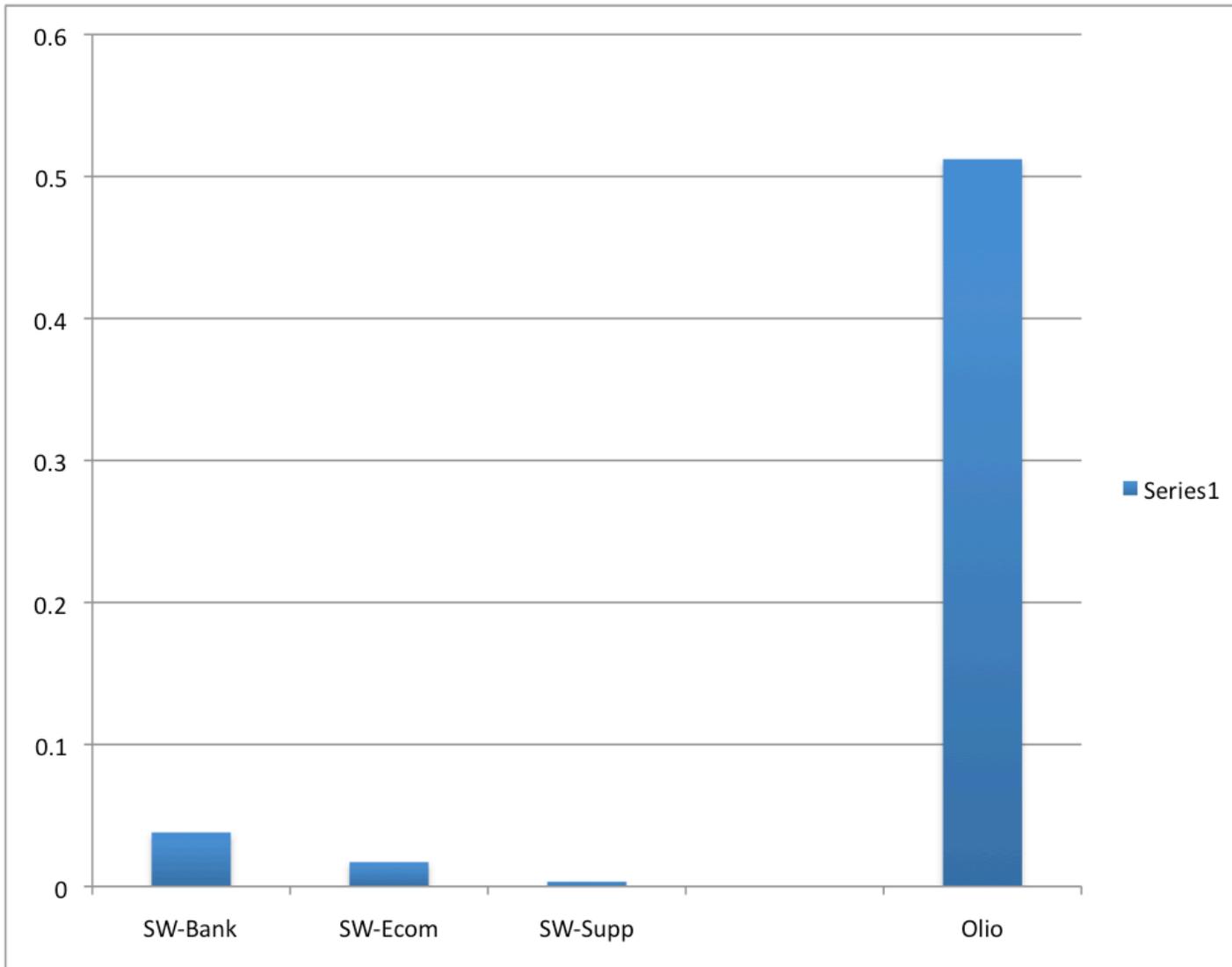
Comparing CPU for Olio with SPECweb

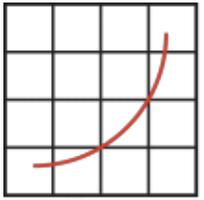




spec[®]

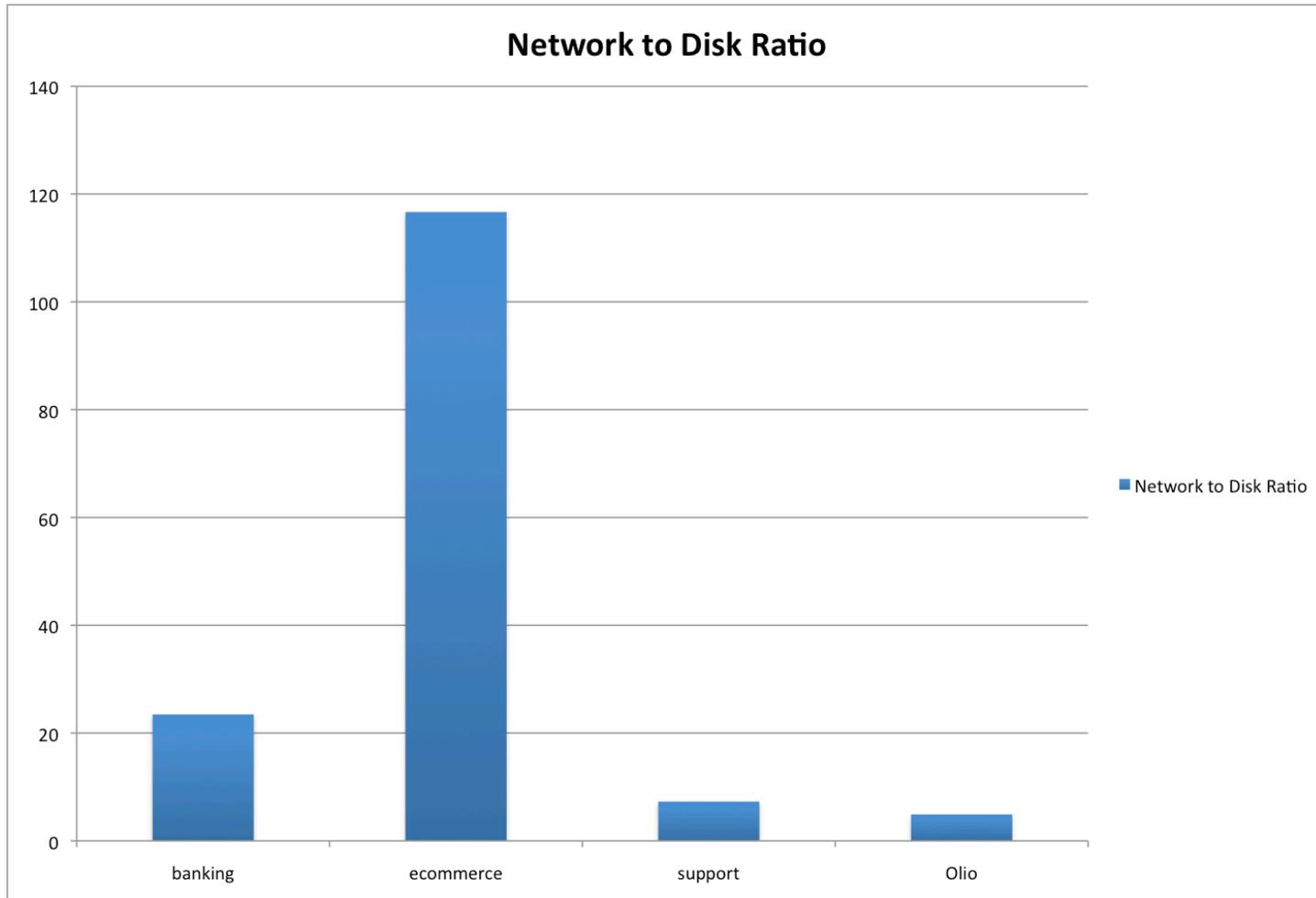
Ratio of Backend RCV to Client TX





spec[®]

Network bytes to Disk Bytes ratio





Highlights of differences between SPECweb workloads

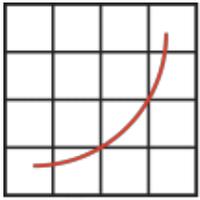
- CPU usr load for Banking > Ecommerce > Support
- PHP cpu usage is lot higher than JSP or ASPX;
- PHP also has higher DRAM bandwidth usage due to script processing/compilation for each request.
- Windows shows lower DRAM bandwidth; but higher cpu utilization. (Reason ??)
- DRAM bandwidth for PHP scripts was lot higher than those for processed scripts like JSP and ASPX



How do the web2.0 workloads differ from what we have?

- Much higher backend traffic
- Much higher Disk traffic; perhaps close to SPECwebSupport.
- Much higher client to SUT traffic; includes a lot of images and data, resulting in higher writes to SUT/Backend.
- CPU usage is very similar to the Windows PHP/Support workload. Reason: it is handling PHP script processing.
- Network usage somewhat similar to SPECwebSupport.

Backup



spec[®]

DRAM bandwidth

