Trading off Quality for Throughput Using Content Adaptation in Web Servers

Michael Gopshtein

Dror Feitelson

Hebrew University, Jerusalem

CNN on 9/11

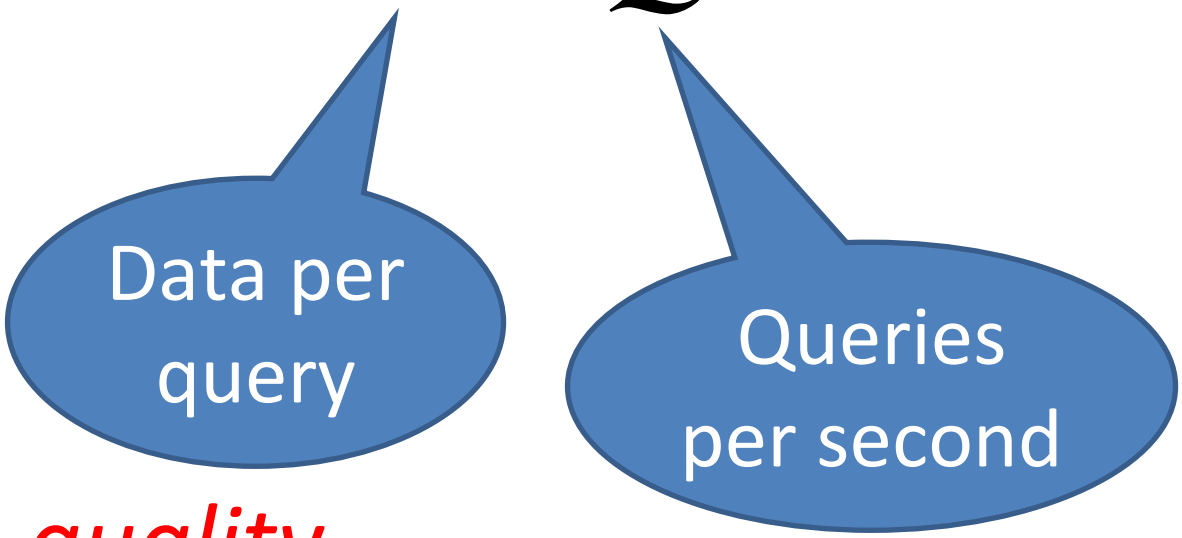
- Load grew from 85,000 hits/sec to 229,000 hits/sec in 15 minutes
- Hosting service diverted additional servers: grew from 10 to 52
- Shut down monitoring service
- Reduce content to image, logo, and 1247 bytes of HTML

CNN on 9/11

- Load grew from 85,000 hits/sec to 229,000 hits/sec in 15 minutes
- Hosting service diverted additional servers: grew from 10 to 52
- Shut down monitoring service
- Reduce content to image, logo, and 1247 bytes of HTML

Brewer's DQ Principle

$$D \times Q = const.$$



Previous Work

- Switching between versions of a site
  - Create a parallel degraded content tree
    - Remove decoration images
    - Compress remaining images
    - Integrate style files and JavaScript
    - Question of doing this automatically
  - Monitor load conditions
    - E.g. response time or server utilization
  - Switch between versions based on load
    - Simply switch root directory of server

Previous Work

- Reported improvements of
  - Up to factor of 2

    [Pradhan and Claypool, “Adaptive content delivery for scalable web servers ”. In *Intl. Network Conf., Jul 2002*]
  - Up to factor of 7 (for very large files)

    Up to factor of 2 (for < 64 KB)

    [Abdelzaher, Shin, and Bhatti, “Performance guarantees for web server end-systems: A control-theoretical approach ”. *IEEE Trans. Parallel & Distributed Syst.* ***13(1)***, pp. 80–96, Jan 2002]

Previous Work

- Automatic content adaptation to support mobile devices
  - Device-specific CSS or layout
- Becoming increasingly important in recent years
- Different emphasis than in our context

Our Contributions

- Describe full implementation on Apache
  - Normal or optimized mode
  - Triggering mode switch
  - Simple integration using existing facilities
- Detailed experimental evaluation
  - Achieved throughput improvement of 2-4 (and up to 10 with more extreme adaptation)
  - Speed of mode switch

Our Contributions

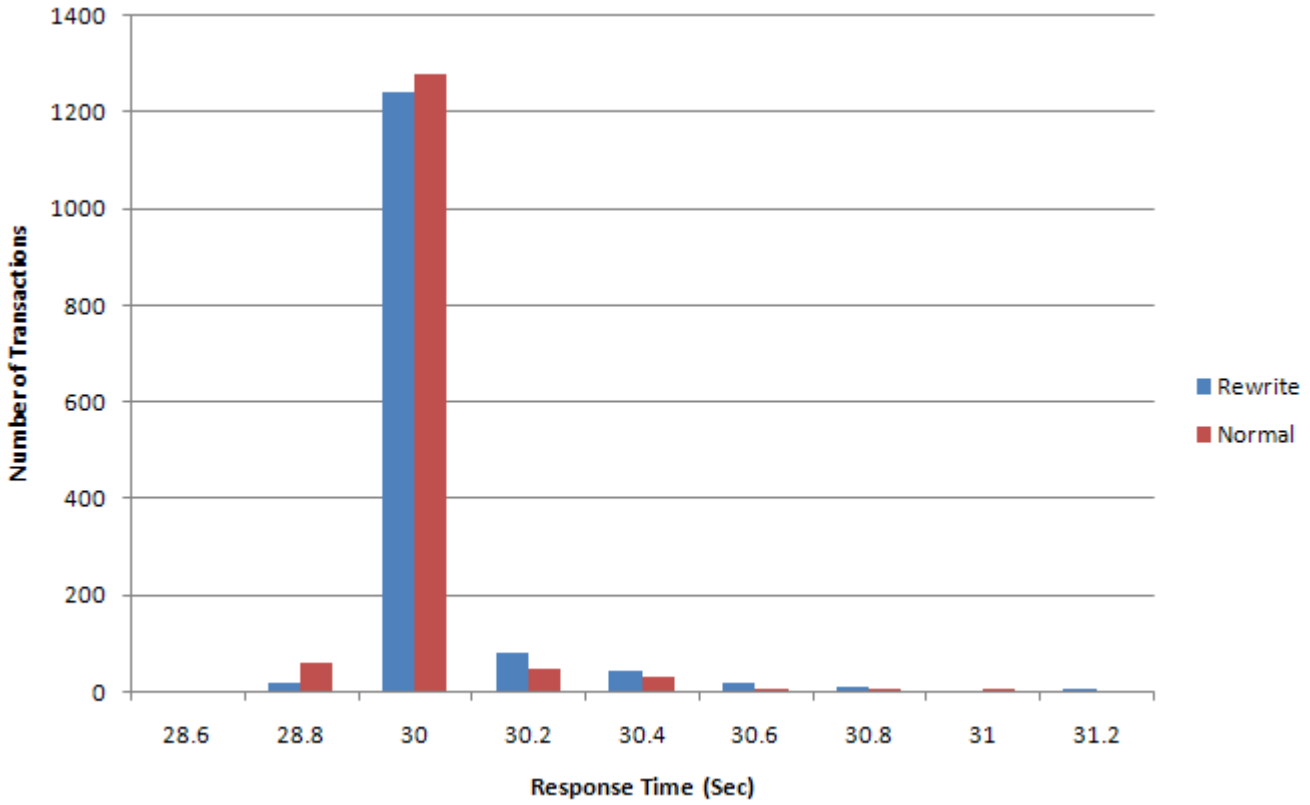
- Describe full implementation on Apache
  - Normal or optimized mode
  - Triggering mode switch
  - Simple integration using existing facilities
- Detailed experimental evaluation
  - Achieved throughput improvement of 2-4 (and up to 10 with more extreme adaptation)
  - Speed of mode switch

Serving Optimized Content

- Optimized content tree mirrored under /opt
- Use Apache mod_rewrite to re-write URL and select appropriate version
- selection rules:
  1. Optimized mode flag is on, implemented as existence of a special file called opt.do
  2. Optimized version of requested file exists and is up to date

Serving Optimized Content

- Mod_rewrite overhead:



Performance Indicators

- CPU utilization
  - 5-second average using sar utility
- Number of incoming TCP connections
  - Available from /proc/net/snmp
- Number of idle Apache processes
  - Available using Apache mod_status

Switching Algorithm

- In normal mode:
  - Keep track of “safe” TCP connection rate
  - If CPU utilization > 85%, switch to optimized
- In optimized mode:
  - Switch to normal if 3 conditions are met:
    1. CPU utilization < 85%
    2. There is at least one idle process
    3. Connection rate is lower than recorded “safe” rate
  - Reduce “safe” rate by 30%

Implementation

- External perl script runs every 5 seconds
- Collects the required statistics
- Creates or removes special file opt.do
- Apache configured with mod_rewrite to look for this file

Our Contributions

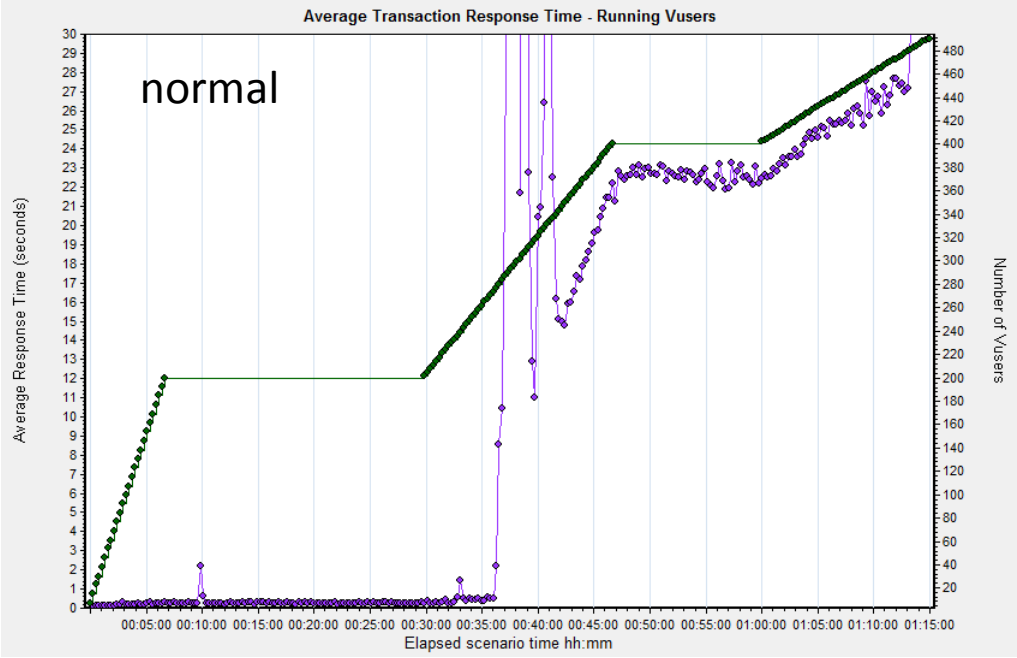
- Describe full implementation on Apache
  - Normal or optimized mode
  - Triggering mode switch
  - Simple integration using existing facilities
- Detailed experimental evaluation
  - Achieved throughput improvement of 2-4 (and up to 10 with more extreme adaptation)
  - Speed of mode switch

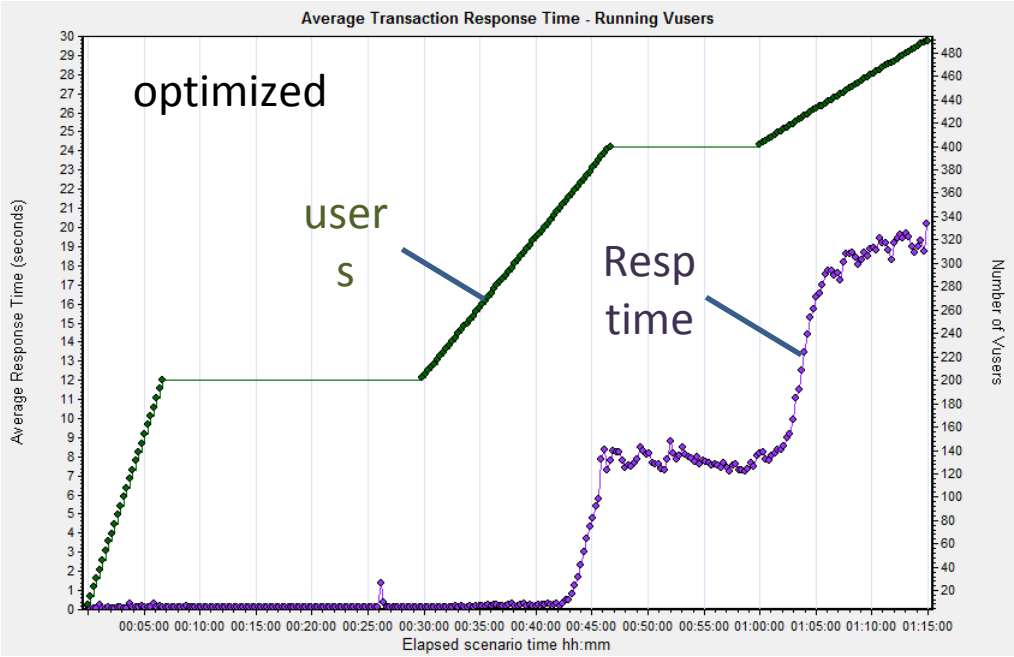
Experimental System

- Apache web server with copy of top500 site
  - Configure with important non-default settings
    - Enable keep_alive connections (1000, 10 sec timeout)
    - Enable in-memory caching (4MB)
    - Enable large number of active processes (1500)
- Several client machines with HP LoadRunner
  - Play scripts of HTTP requests simulating users
  - Loop to simulate requests from additional users
  - List of requests can be manipulated to simulate requests in optimized mode

Response Tim

- Closed-system scenraio
- Optimized version reduces 43% of requests and 22% of bytes
- Normal: 270 ms
- Optimized: 138 ms
- Demonstrates need for enhanced Apache settings





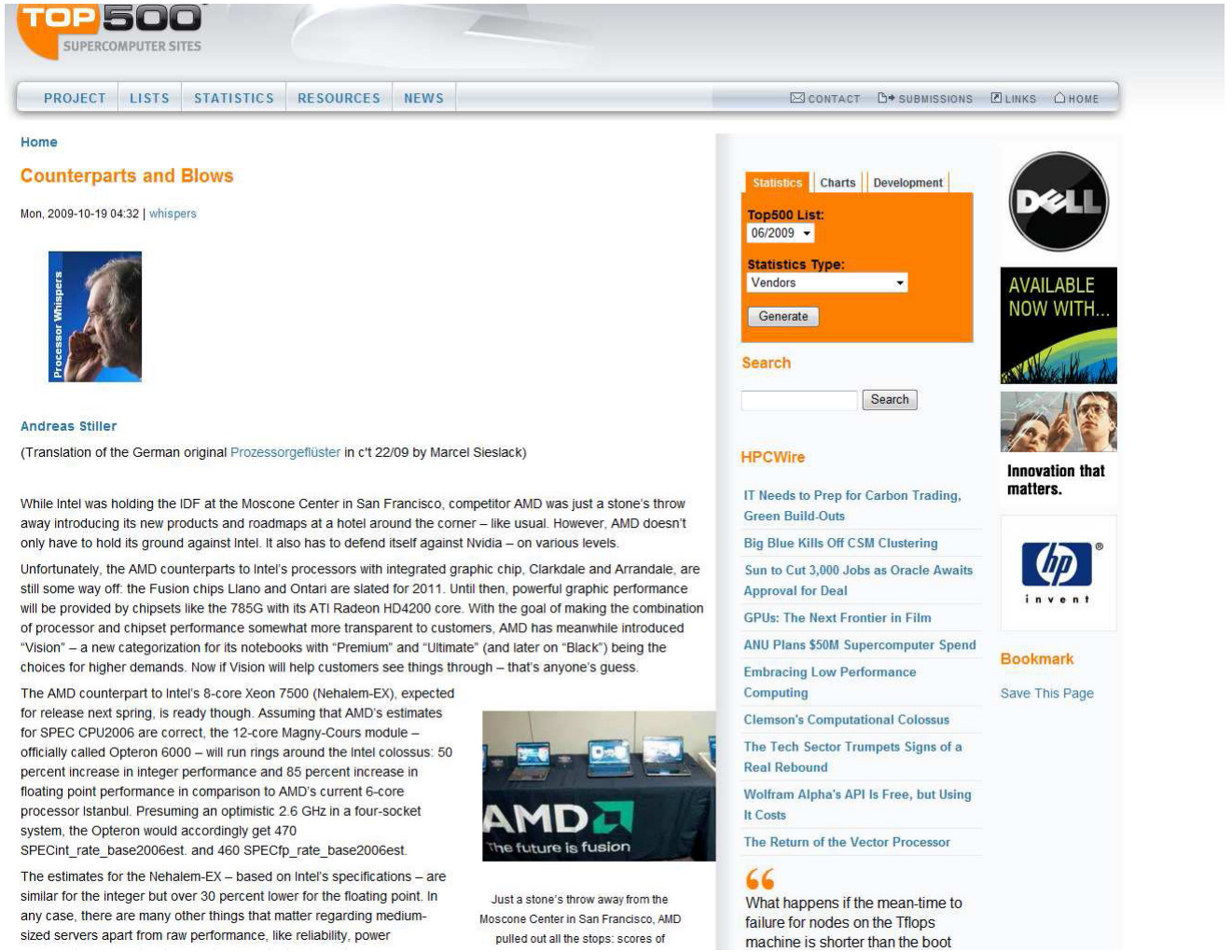
Slashdot Scenario

- All users request the same page
- Open-system with no feedback effects

Normal	Moderate opt	Extreme opt
59 HTTP reqs	29 HTTP reqs	3 HTTP reqs
322 KB	188 KB	65 KB
2 connections	2 connections	1 connection

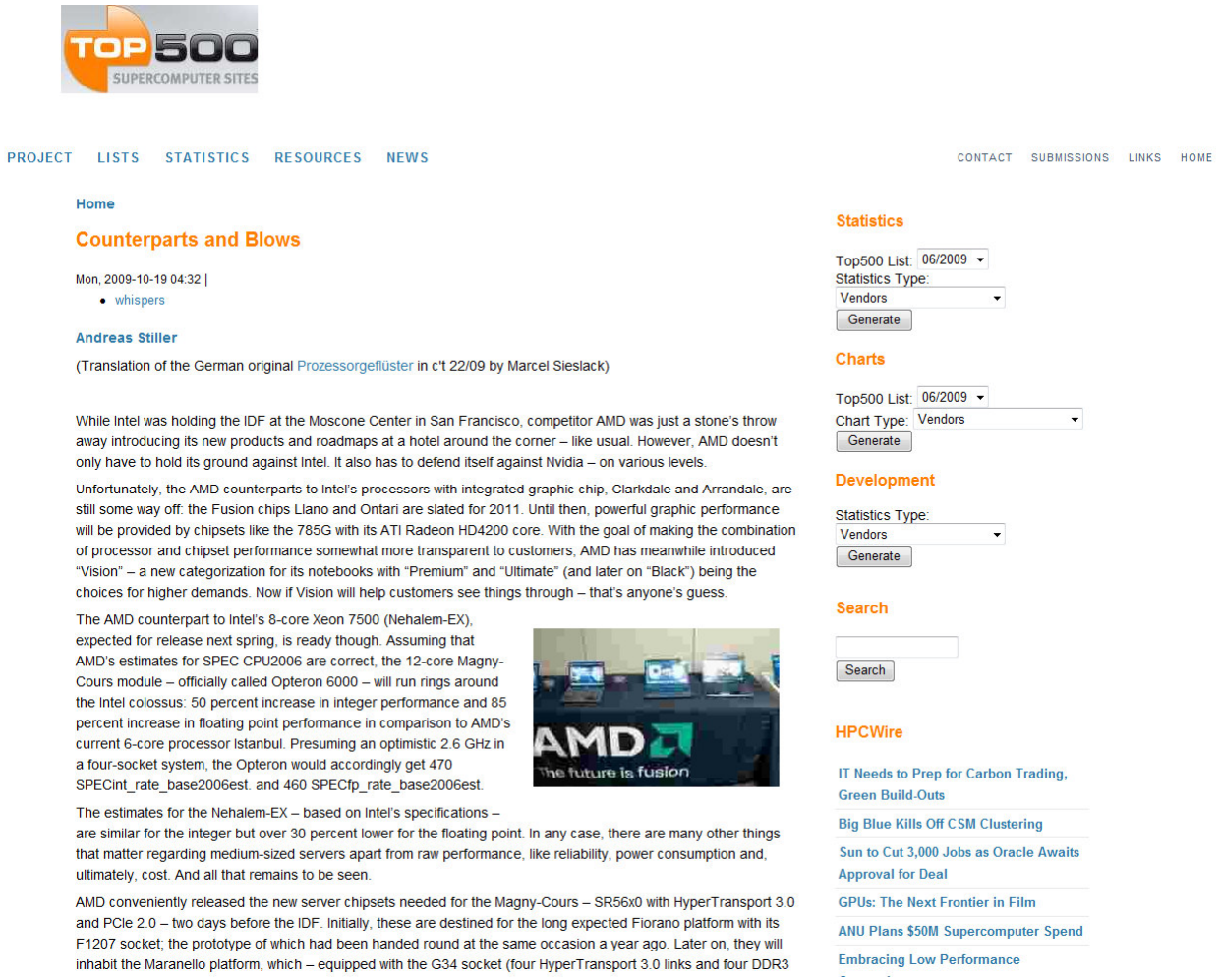
Extreme Optimization

- Leave only logo and story image
- Compress them
- Remove CSS and JS, integrate minimal needed parts



Extreme Optimization

- Leave only logo and story image
- Compress them
- Remove CSS and JS, integrate minimal needed parts



TOP500 SUPERCOMPUTER SITES

PROJECT LISTS STATISTICS RESOURCES NEWS CONTACT SUBMISSIONS LINKS HOME

Home

Counterparts and Blows

Mon, 2009-10-19 04:32 |

- whispers

Andreas Stiller

(Translation of the German original Prozessorgeflüster in c't 22/09 by Marcel Sieslack)

While Intel was holding the IDF at the Moscone Center in San Francisco, competitor AMD was just a stone's throw away introducing its new products and roadmaps at a hotel around the corner – like usual. However, AMD doesn't only have to hold its ground against Intel. It also has to defend itself against Nvidia – on various levels.

Unfortunately, the AMD counterparts to Intel's processors with integrated graphic chip, Clarkdale and Arrandale, are still some way off: the Fusion chips Llano and Ontari are slated for 2011. Until then, powerful graphic performance will be provided by chipsets like the 785G with its ATI Radeon HD4200 core. With the goal of making the combination of processor and chipset performance somewhat more transparent to customers, AMD has meanwhile introduced "Vision" – a new categorization for its notebooks with "Premium" and "Ultimate" (and later on "Black") being the choices for higher demands. Now if Vision will help customers see things through – that's anyone's guess.

The AMD counterpart to Intel's 8-core Xeon 7500 (Nehalem-EX), expected for release next spring, is ready though. Assuming that AMD's estimates for SPEC CPU2006 are correct, the 12-core Magny-Cours module – officially called Opteron 6000 – will run rings around the Intel colossus: 50 percent increase in integer performance and 85 percent increase in floating point performance in comparison to AMD's current 6-core processor Istanbul. Presuming an optimistic 2.6 GHz in a four-socket system, the Opteron would accordingly get 470 SPECint_rate_base2006est. and 460 SPECfp_rate_base2006est.

The estimates for the Nehalem-EX – based on Intel's specifications – are similar for the integer but over 30 percent lower for the floating point. In any case, there are many other things that matter regarding medium-sized servers apart from raw performance, like reliability, power consumption and, ultimately, cost. And all that remains to be seen.

AMD conveniently released the new server chipsets needed for the Magny-Cours – SR56x0 with HyperTransport 3.0 and PCIe 2.0 – two days before the IDF. Initially, these are destined for the long expected Fiorano platform with its F1207 socket; the prototype of which had been handed round at the same occasion a year ago. Later on, they will inhabit the Maranello platform, which – equipped with the G34 socket (four HyperTransport 3.0 links and four DDR3

Statistics

Top500 List: 06/2009
Statistics Type:
Vendors
Generate

Charts

Top500 List: 06/2009
Chart Type: Vendors
Generate

Development

Statistics Type:
Vendors
Generate

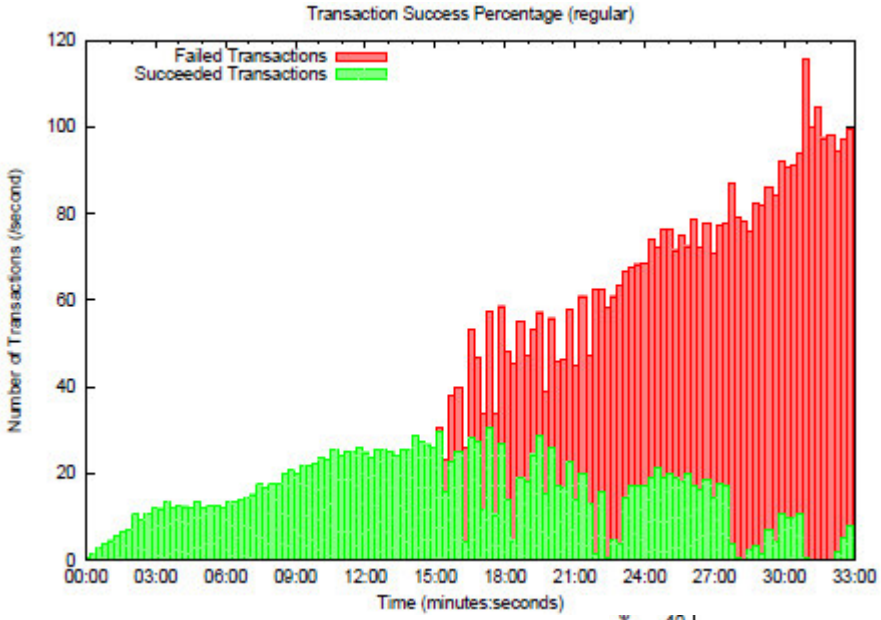
Search

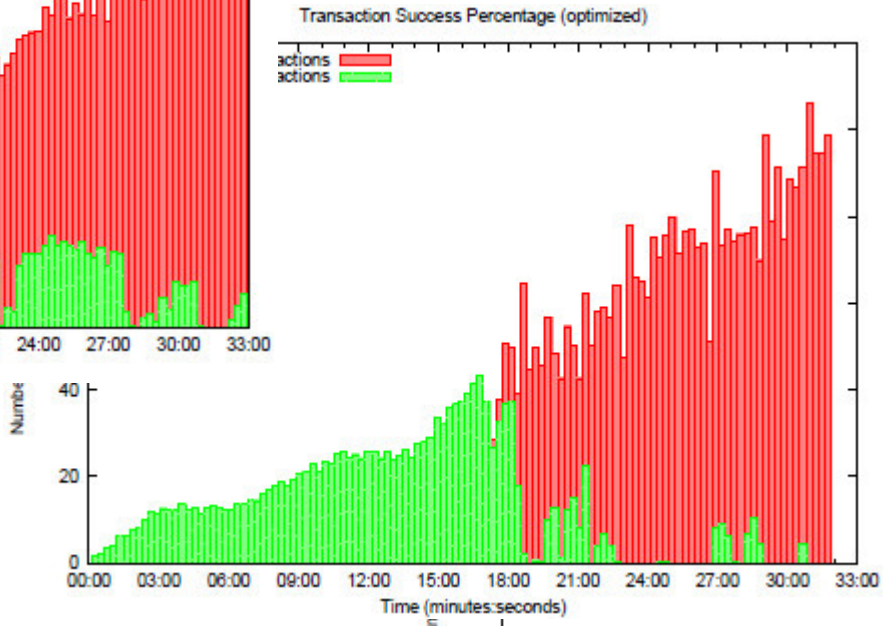
Search

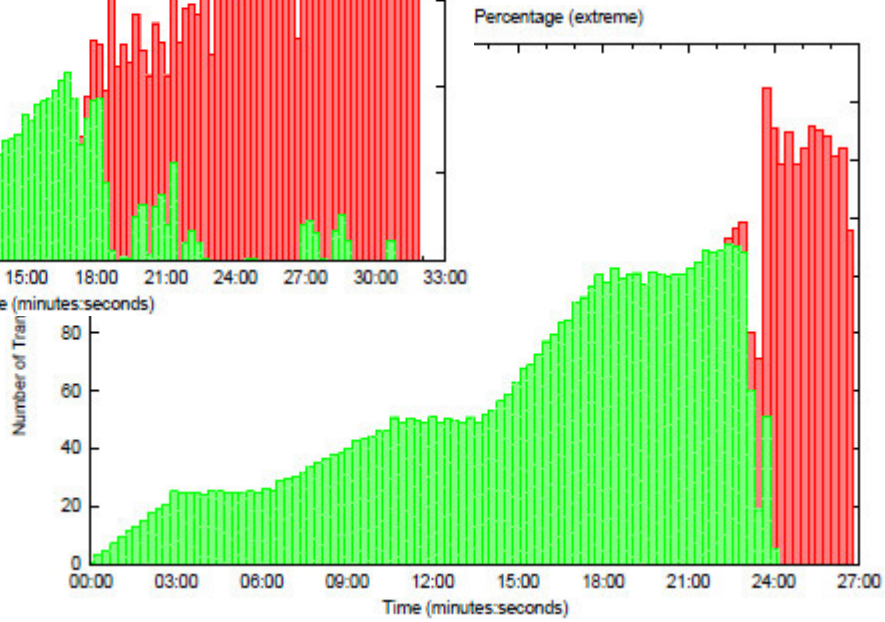
HPCWire

IT Needs to Prep for Carbon Trading, Green Build-Outs
Big Blue Kills Off CSM Clustering
Sun to Cut 3,000 Jobs as Oracle Awaits Approval for Deal
GPUs: The Next Frontier in Film
ANU Plans $50M Supercomputer Spend
Embracing Low Performance

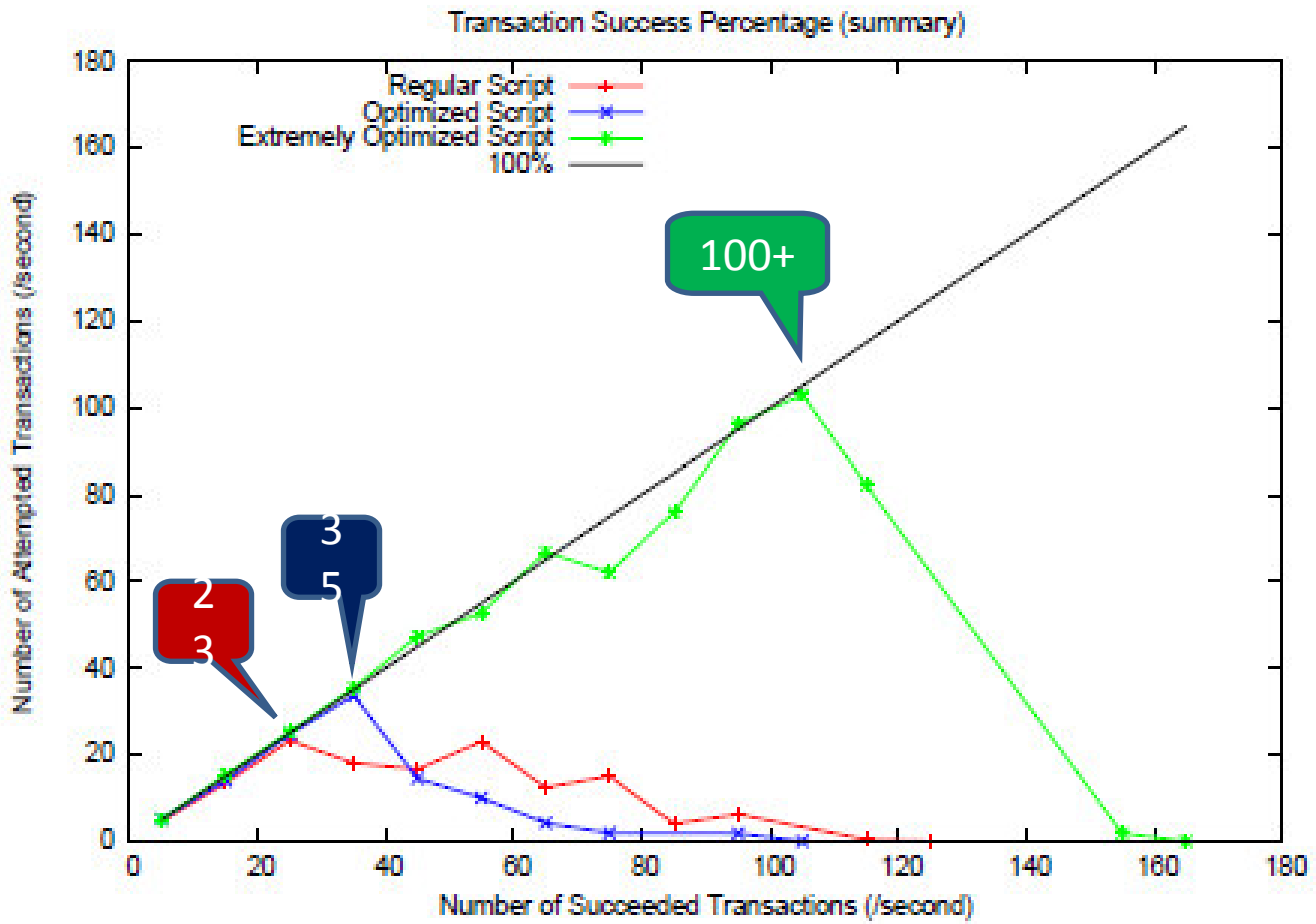
Transaction Success Rate



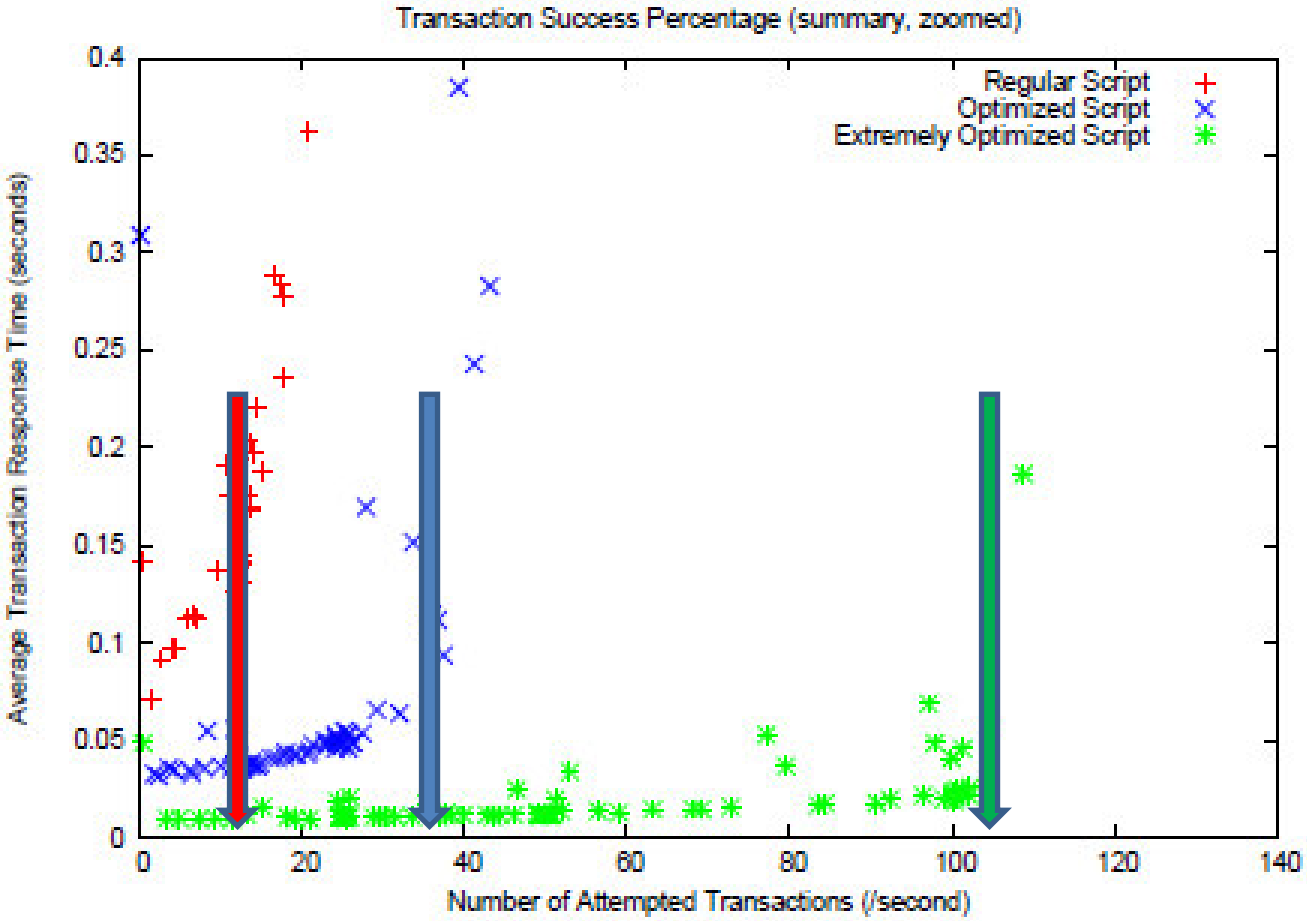




Throughput

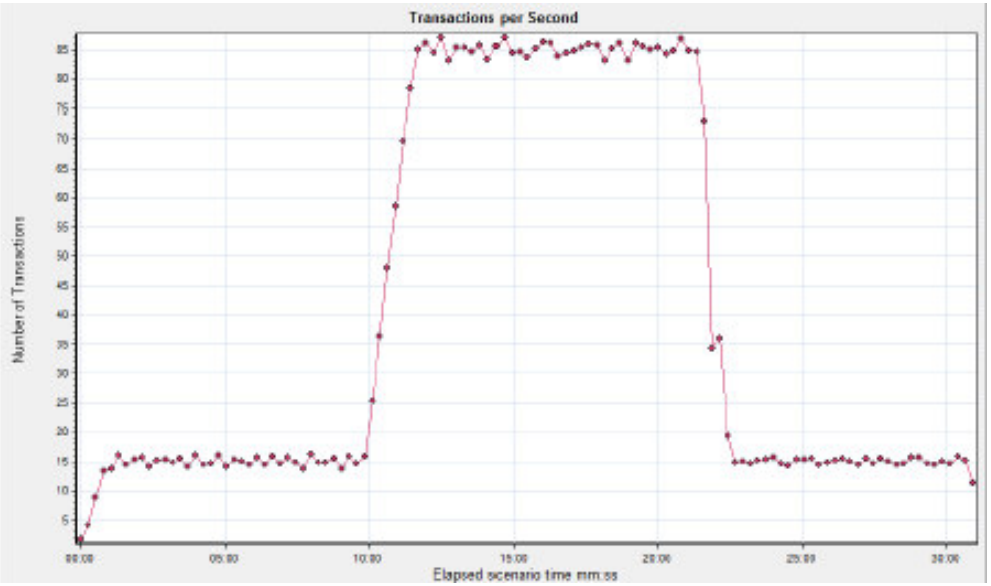

Transaction Success Percentage (summary)
Regular Script
Optimized Script
Extremely Optimized Script
100%
100+
3
5
2
3
Number of Attempted Transactions (/second)
Number of Succeeded Transactions (/second)
0
20
40
60
80
100
120
140
160
180

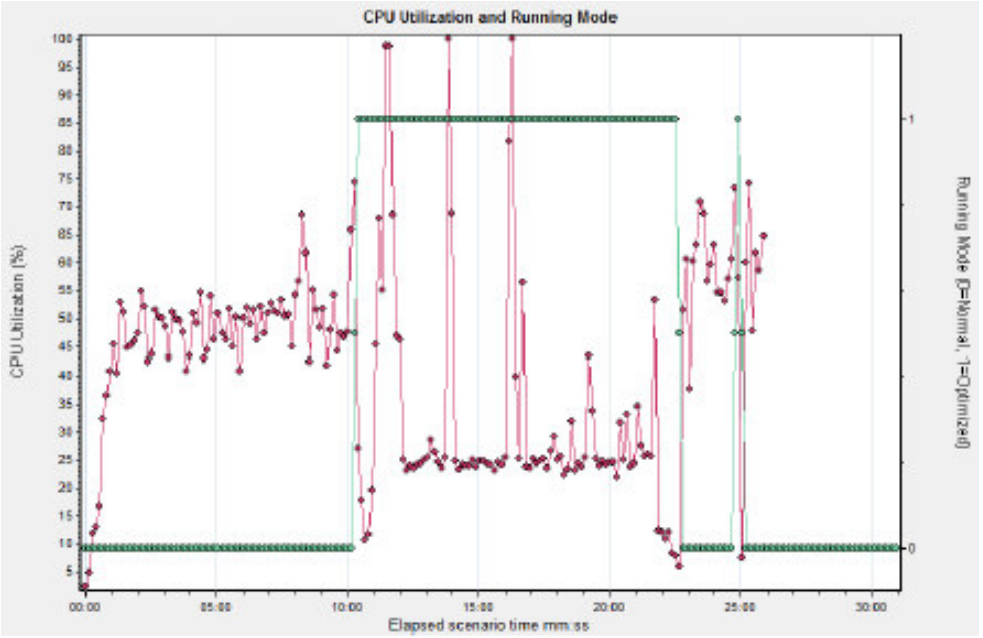
Throughput


Transaction Success Percentage (summary, zoomed)
Regular Script
Optimized Script
Extremely Optimized Script
Average Transaction Response Time (seconds)
Number of Attempted Transactions (/second)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0
20
40
60
80
100
120
140

Mode Switch

- Load: create step in number of requests
  - From 15 to 85 requests per sec
- Causes near-immediate switch to optimized mode
- And actual drop in CPU utilization and response time





Conclusions

- Technical aspects of content adaptation are easy
  - Monitoring using available hooks
  - Switching with mod_rewrite
- Increase throughput by 2-4 with little degradation
- Increase by up to 10 possible
- Larger increase probably requires other techniques
- Automatic creation of adapted content requires more work
- Need for careful experimental evaluation