Improving the Scalability of a Multi-core Web Server

R. Hashemian¹, D. Krishnamurthy¹, M. Arlitt², N. Carlsson³

1. University of Calgary
2. HP Labs
3. Linköping University

by: Raoufehsadat Hashemian

The 4th ACM/SPEC International Conference on Performance Engineering
ICPE 2013
OUTLINE

• Introduction
• Scalability Evaluation
• Scalability Enhancement Approach
• Validation
• Conclusion
**INTRODUCTION**

**PROBLEM DESCRIPTION**

- **Enterprise applications**
  - Performance: Improving QoS
    - e.g. Lower response times
  - Cost: Less money spent on hardware
    - e.g. Improving effective utilization

- **Goal**: Higher utilization and acceptable response time
- **How to achieve this “Goal” for Web servers running on Multi-core hardware?**
INTRODUCTION

BACKGROUND

• Web servers before multi-core
  • Mature topic, wide-ranging discussions

• Multi-core architecture
  • Most research on batch (non-interactive) workload

• Web servers running on Multi-core
  • BUS problem in UMA system (Veal et al. `07)
  • Multiple Web server instances: 1 instance per processor (Scogland et al. `09, Boyd et al.,10 Gaud et al.,11)
SCALABILITY EVALUATION

SCALABILITY MEASUREMENT

• Measure Web server scalability for two workloads
• Evaluate the effectiveness of multiple Web server approach in the server’s scalability
• Scalability
  • Maximum Achievable Throughput (MAT)
SCALABILITY EVALUATION

EXPERIMENTAL SETUP

- 2 x 4 core Intel Xeon E5620 processors

<table>
<thead>
<tr>
<th>Microarch.</th>
<th>Nehalem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>L1 Cache</td>
<td>32K IC - 32K DC</td>
</tr>
<tr>
<td>L2 Cache</td>
<td>256K</td>
</tr>
<tr>
<td>L3 Cache</td>
<td>12M (Inclusive)</td>
</tr>
<tr>
<td>Inter-conn.</td>
<td>QPI -5.86 GT/s</td>
</tr>
<tr>
<td>Memory</td>
<td>16GB - DDR3-1333</td>
</tr>
</tbody>
</table>

OS: Linux, kernel 3, Ubuntu

Webserver: Lighttpd

Application Server: php (FastCGI module)
SCALABILITY EVALUATION
WORKLOADS

• TCP/IP Intensive workload
  • High TCP connection rate
  • Processing: low user level & high kernel level
  • 1 KB static file, up to 155,000 requests/second

• SPECweb Support workload
  • Both static requests and php requests
  • Wider range of request types
  • Processing: high user level & moderate kernel level
SCALABILITY EVALUATION

CONFIGURATION TUNING

• Change default lighttpd recommendation (1 Lighttpd worker process per core)

• Disable default Linux scheduling (use affinity)

• Distribute interrupt handling load

• Improved MAT up to 69%
• Balanced utilization levels for the eight cores
• Fully utilized the server
SCALABILITY EVALUATION

RESULTS

- TCP/IP Intensive workload
  - Sub-linear
    Maximum Achievable Throughput
    \(146,000\) req/sec

- SPECweb Support workload
  - Almost linear
    Maximum Achievable Throughput
    \(23,000\) req/sec
Response time vs. Core Count

- “Low response time” requests
  - Static requests
  - Performance *degrades*
- “High response time” requests
  - Dynamic requests
  - Performance *improves*

Knowing this behavior, how can we improve the scalability?

**CDF of Response times**
- 80% CPU Utilization
- SPECweb Support Workload
SCALABILITY ENHANCEMENT

MULTIPLE WEBSITE REPLICAS

- Approach: Use 1 Web server instance per processor
- Goal: Reduce inter-processor data migration

Original Configuration with one replica

Alternative Configuration with two replicas
SCALABILITY ENHANCEMENT

EVALUATING NEW CONFIGURATION

TCP/IP Intensive Workload
- Scalability Improvement
- MAT increment: 12.3%

SPECweb Support Workload
- Scalability Degradation
- MAT decrement: 10%
SCALABILITY ENHANCEMENT
EVALUATING NEW CONFIGURATION

- The response time inflation for *Dynamic* requests dominates the improvement achieved for *Static* requests.
- Mean and 99.9th percentile response times increase with 2-replicas.

**Hypothesis:**
- Cache contention with 2-replicas due to the larger working set size of dynamic requests.

CDF of Response times
80% CPU Utilization
22,000 req/sec
SPECweb Support workload
VALIDATION
INTER-CONNECT TRAFFIC

- Inter-connect traffic decreased significantly
- Improved performance

- No significant decrement
- Improved performance for Static requests

TCP/IP Intensive Workload

SPECweb Support Workload
Last Level cache (LLC) HIT ratio degrades with 2-replica configuration

Confirms the cache contention hypothesis

SPECweb Support Workload
CONCLUSIONS

- Multi-core Web server: scalable after tuning
  - 80% utilization with acceptable response time

- Multiple Website Replicas
  - The effect on the scalability is workload dependent
  - Dynamic requests trigger LLC contention
    - Contention may be architecture and application dependent

- Future plan:
  - Design and develop an automatic, workload adaptive technique which decides about best configuration
Questions?
Thank you!

Raoufeh Hashemian
University of Calgary, Canada

rhashem@ucalgary.ca

This work is financially supported by:
REFERENCES

- Cherkasova et al. `00: Characterizing Temporal Locality and its Impact on Web Server Performance, International Conference on Computer Communications and Networks’00, Cherkasova; Ciardo; HP Labs
- Elnozahy et al. `03: Energy Conservation Policies for Web Servers, USITS ’03, Elnozahy; Kistler; Ramakrishnan; IBM
- Majo et al. `12: Matching Memory Access Patterns and Data Placement for NUMA Systems, GC’12, Majo; Gross; ETH
- Blagodurov et al. `11: A case for NUMA-aware contention management on multicore systems, USENIX ATC’11, Blagodurov; Zhuravlev; Dashti; Fedorova; SFU
- Veal et al. `07: Performance scalability of a multi-core web server. ACM/IEEE ANCS’07, Veal; Foong; Intel
- Scogland et al. `09: Asymmetric interactions in symmetric multi-core systems: Analysis, enhancements and evaluation. ACM/IEEE SC’08, Scogland; Balaji; Feng; Narayanaswamy,
- Boyd et al. `10: An analysis of linux scalability to many cores, USENIX OSDI’10, Boyd-Wickizer; Clements; Mao; Pesterev; Kaashoek; Morris; Zeldovich, MIT
- Gaud et al. `11: Application-level optimizations on numa multicore architectures: the apache case study, RR-LIG-011, Gaud; Lachaize; Lepers; Muller; Quema.
SCALABILITY EVALUATION

CONFIGURATION TUNING

- Network interrupt handling
  - 4 RSS queue per NIC port
  - Each queue bind to one core

![Graph showing response time vs rate before and after distributing internal load.](image-url)
SCALABILITY EVALUATION
CONFIGURATION TUNING

• OS scheduling
  • Binding each lighttpd process to 1 core
SCALABILITY EVALUATION
WEB TIER VS. APPLICATION TIER

- **Static**: Requests with lower response time
  - Processed only in Web tier (lighttpd)
- **Dynamic**: Requests with higher response time
  - Processed only in Web and application tiers (lighttpd and php)
SCALABILITY EVALUATION

EXPERIMENTAL SETUP

Improving the Scalability of a Multi-core Web Server