Achieving Load Balance and Effective Caching in Clustered Web Servers

Richard B. Bunt

Derek L. Eager

Gregory M. Oster

Carey L. Williamson

April 2, 1999

1. Introduction

- Internet traffic continues to increase, with Web usage being the dominant traffic generator
  - Performance problems: slow response times, network congestion, etc.

  - Approaches:
    - improve Web caching architectures to save network bandwidth

    - higher-performance servers
      - clustered servers

2. Project Goals

- Evaluate clustered server architectures and factors that affect their performance

  - Request/load distribution policies
    - load balancing
    - cache performance

  - Configuration questions
    - intra-cluster bandwidth
    - number of server nodes (scalability)
    - distribution of URL namespace

- Explore tradeoffs in server design using trace-driven simulation

3. Related Work

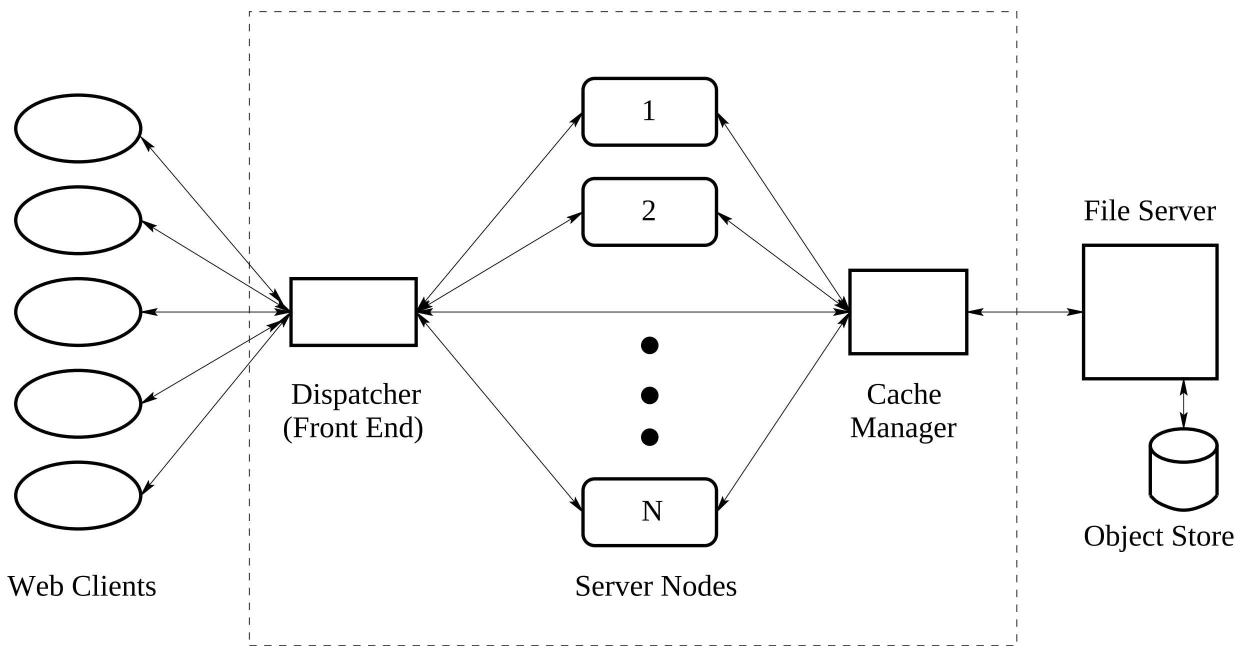
- Clustered architectures
  - LARD (Pai, Banga, Svendsen, Druschel, Zwaenepoel, and Nahum)
  - Shock Absorber (Goldszmidt and Hunt)
  - Womplex (IBM)

- Client and server workload characterization
  - Crovella and Bestavros
  - Arlitt and Williamson

- Proxy workload characterization
  - Mogul
  - Cao, *et al.*

- Proxy caching hierarchies
  - Caceres, Douglis, *et al.*

4. Web Server Model



5. Utilizing Current Load Information

- Policies for utilization of load information (ignoring caching considerations)
  - Random
    - no use of load information
    - request distribution is random
  - Round-Robin
    - no use of load information
    - request distribution is round-robin
  - Load-Based
    - uses current server load
    - requests sent to the most lightly loaded server

6. Utilizing Cache-content Information

- Approaches to utilize cache-content information:
  - No use of cache information
    - request distribution is done only according to load-balance considerations

  - Cache affinities (estimated)
    - hashed assignment of the URL space

  - Cache affinities (content-based)
    - using current cache contents

7. Performance Metrics

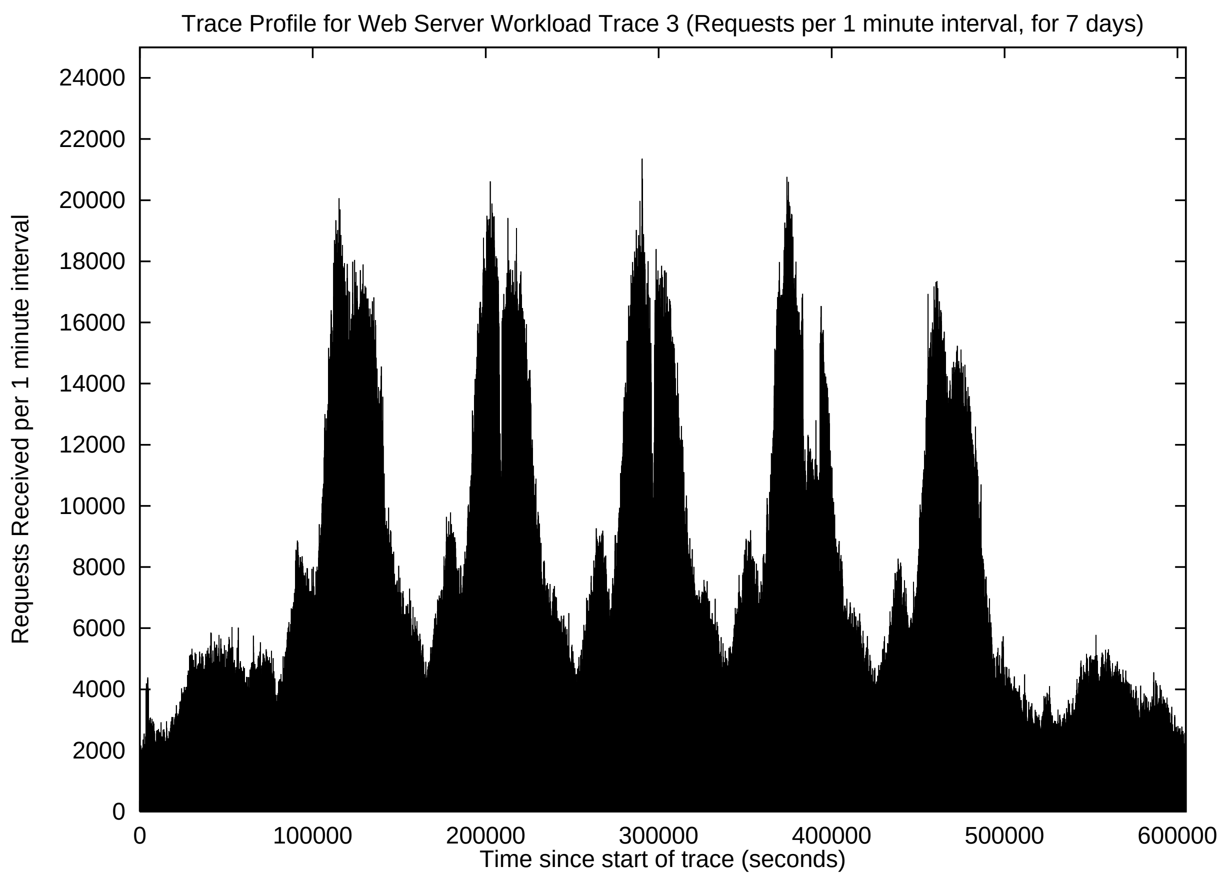
- Load Balance (LBM) - weighted average of sampled peak-to-mean ratios.
  1.0 $\leq LBM \leq N$

- Cache Hit Ratios
  - object hit ratio
  - byte hit ratio

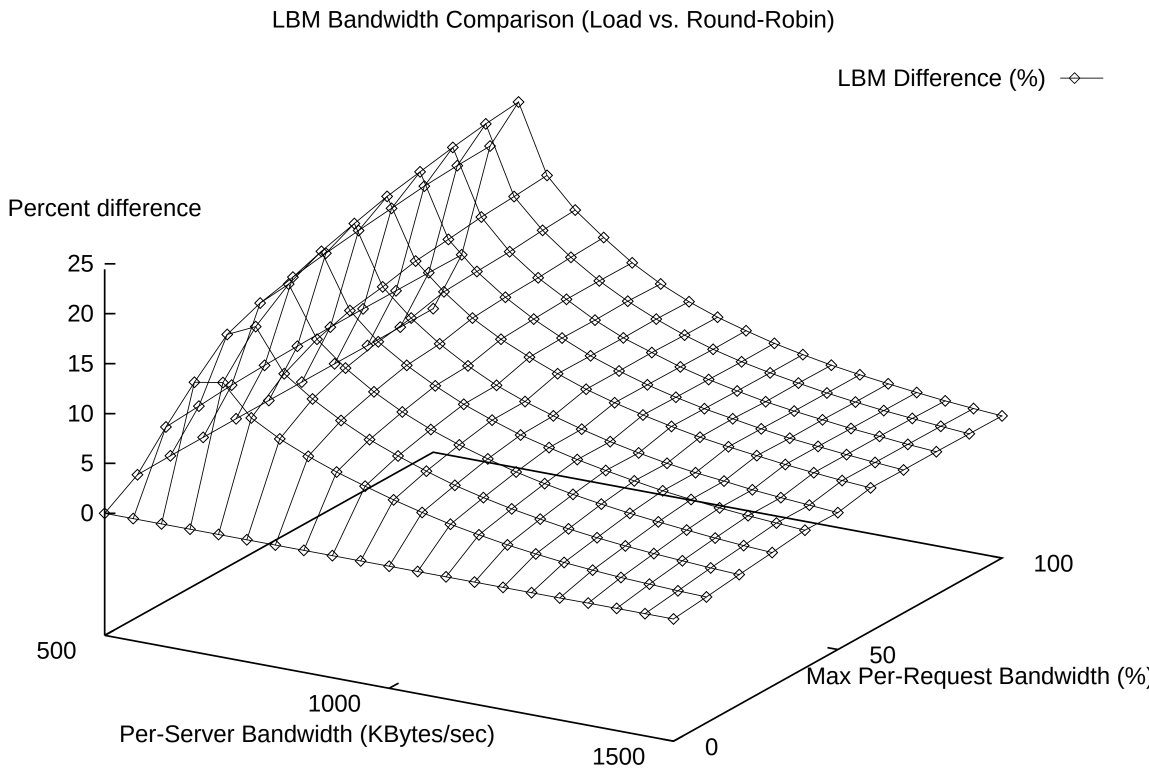
- Response Time Inflation Factors

8. Summary of Trace Characteristics

Duration	7 days
Start Date	June 14, 1998
Total Requests	56,458,479
Average Requests/Day	8,065,497
Average Object Size (bytes)	5,716
Total MBytes Transferred	307,785
Average MBytes/Day	43,969



9. Load Balancing Metric Comparison: Load vs. Round-Robin

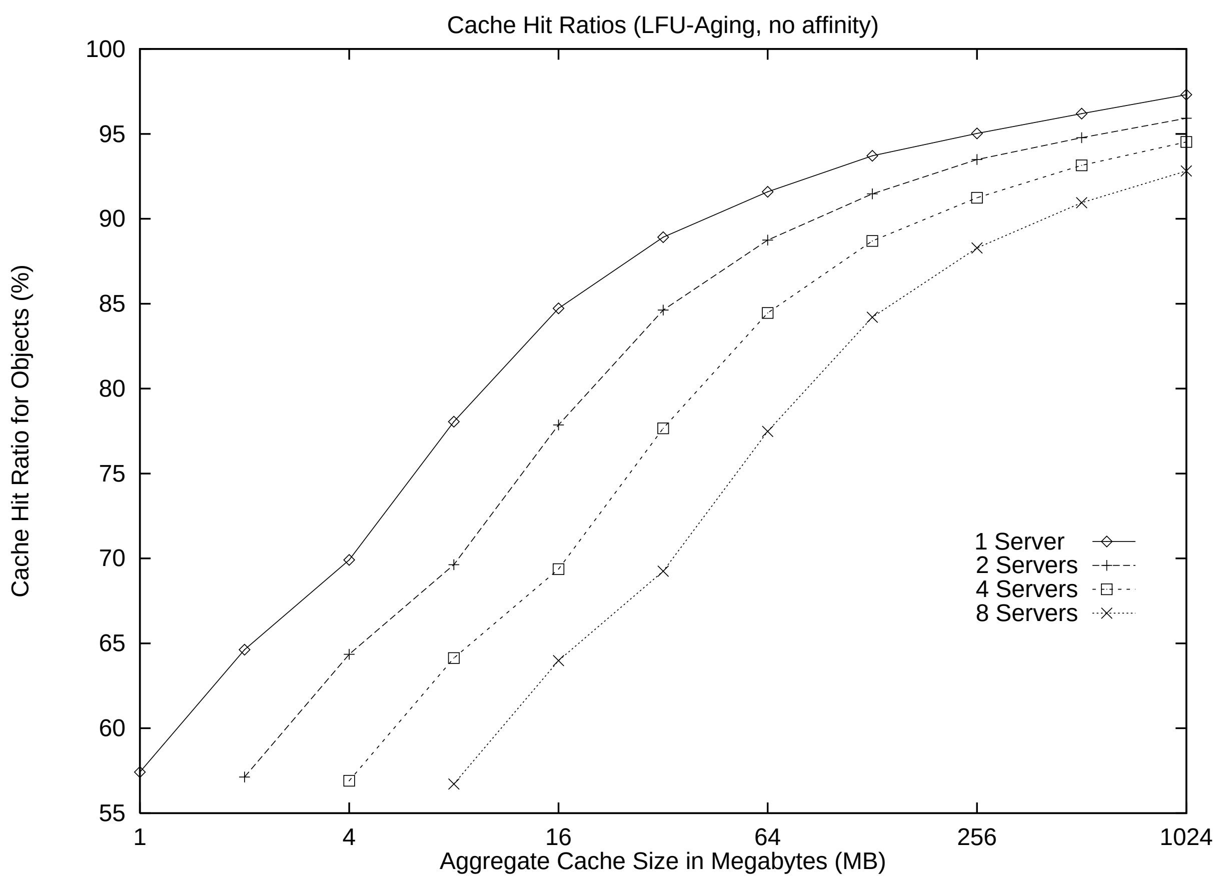


10. Scalability with Number of Servers

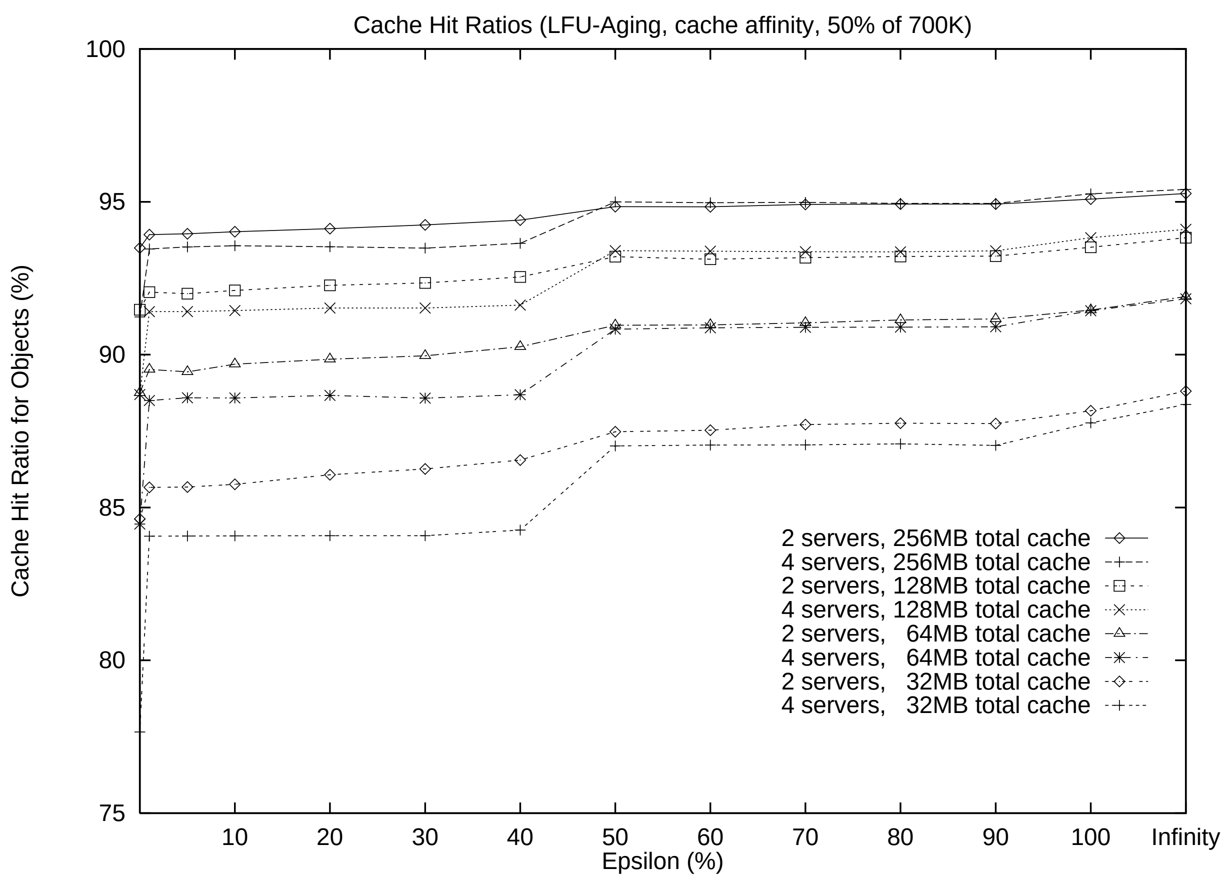
(Fixed aggregate server bandwidth of 1400KB/sec)

Servers	Load LBM		Round-Robin LBM	
	Bandwidth	Requests	Bandwidth	Requests
2	1.059	1.142	1.248	1.406
4	1.177	1.688	1.706	2.217
8	1.294	2.200	2.236	3.087
16	1.406	2.608	2.738	3.897
32	1.508	2.881	3.116	4.705

11. Caching Performance



12. Effect of Cache Affinity



13. Conclusions

- Use of current load information is needed for load balancing, but only if the achievable per-request bandwidth is not strongly network or client limited

- Use of cache content information is not required for good cache behaviour

- Load balancing and caching policies can negatively affect each other

- Good cache performance and good load balance are both achievable if both objectives are taken into consideration

14. Future Work

- Geographically distributed server architectures
- Distributed web proxies
- Proxy caching hierarchies