Eliminating the I/O Bottleneck in Large Web Caches

Alex Rousskov and Valery Soloviev

North Dakota State University

{rousskov,soloviev}plains.NoDak.edu

The Problem

- Web growth
- Expensive network bandwidth and poor response time
- Large Web caches (**30-50 GB** and larger)
- At least **80%** of traffic goes through disk storage
- Disk storage subsystem is a bottleneck for *hits*

Traditional Approach

- Caching policies adopted from DBs and FSs

- Optimize Hit Ratios while ignoring disk I/O overhead

- Algorithms give very close Hit Ratios on *large* caches

- Every incoming cachable document is swapped to disk

- Disk traffic proportional to Web traffic

- Peak load: severe performance degradation; “Night”: idle

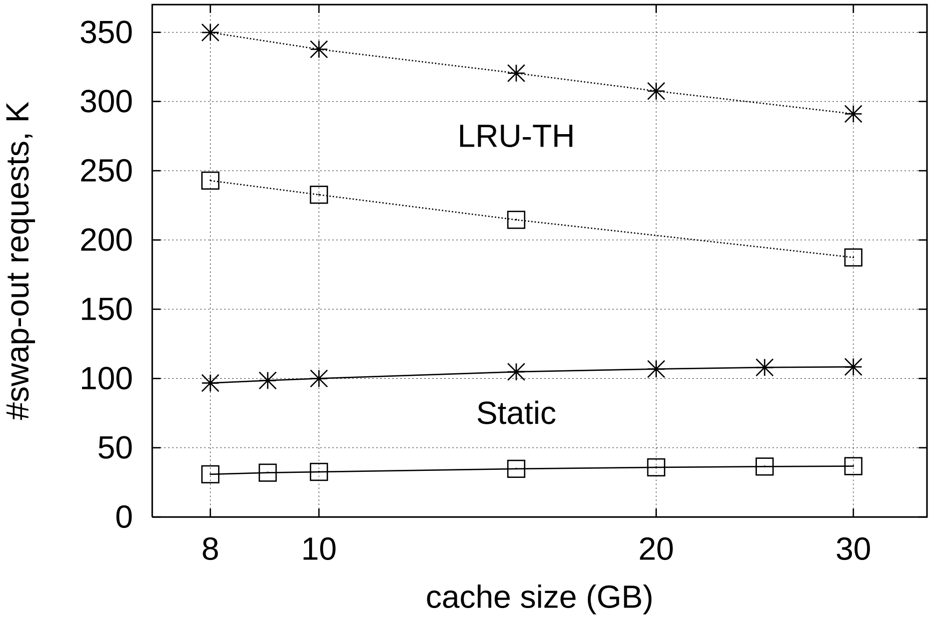
Proposed Scheme

- Once per day, scan the logs, find most *valuable* documents
- Form an *active* set of most valuable documents
- Preserve the active set during peak load;
  updates are OK but no new documents are cached
- Peak load disk traffic consists mostly of read requests
- More than **50%** of disk requests are eliminated

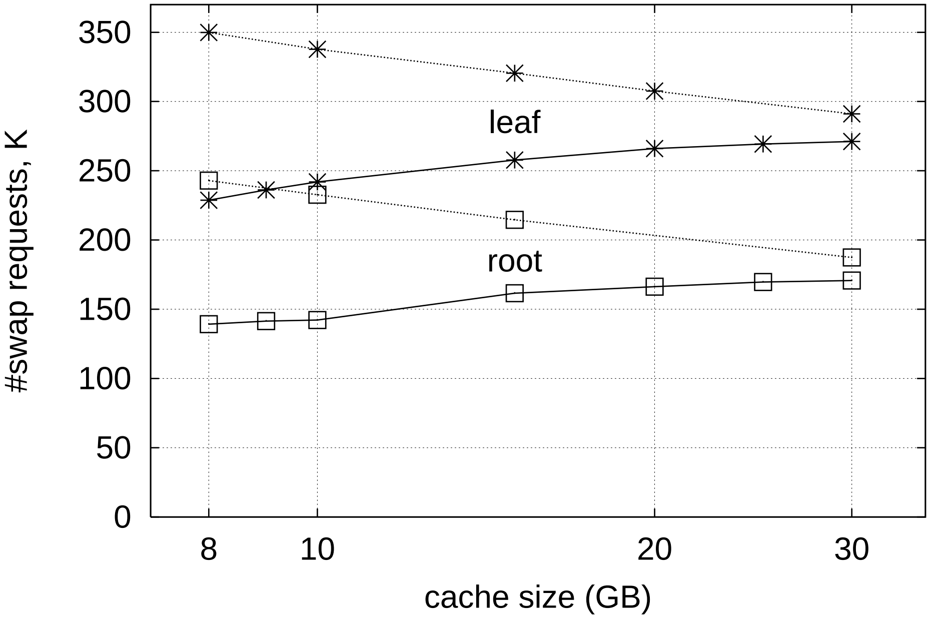
Performance

Disk (swap) requests saved

On-Line Swap-Out Activity



Total Disk Activity



Conclusions

- Traditional algorithms do not scale well with Web growth
- We maintain Hit Ratios at the level of traditional algorithms,
- substantially decrease disk activity during peak load
- Our approach improves hit response time and
- reduces overhead from maintaining dynamic cache contents

Future Work

- More diverse group of caches
- Symbiosis of *static* and *dynamic* algorithms
- Tuning and simplifying the heuristics used for off-line valuable document selection
- Implementation in Squid caching proxy