

Exploiting Object Relationships for Deterministic Web Object Management

Mikhail Mikhailov and Craig E. Wills

Computer Science Department

Worcester Polytechnic Institute

`{mikhail,cew}@cs.wpi.edu`

Presented by Mikhail Mikhailov

International Workshop

on

Web Content Caching and Distribution

Boulder, Colorado

August 15, 2002

This Work Is About

MONARCH: A novel Web object management mechanism

- primary goal is to provide **strong** cache consistency
- lower overhead compared to heuristic consistency mechanism
- maintain no per-client state at the server

Outline of Talk

- Motivation
- Our Approach
- Prototype Implementation
- Performance Evaluation
- Conclusions
- On-going Work

Motivation

Today, servers can tell caches:

- nothing at all
- not to cache an object
- when an object Expires
- when an object was Last-Modified
(best case scenario for objects
that change)

Motivation

If given only Last-Modified, caches use %-age of object age as a reasonable freshness estimate

- this approach is heuristic in nature (caches may use different %-ages)
- results in many unnecessary validations (15–37% of all requests)
- clients still receive *stale* objects

Motivation

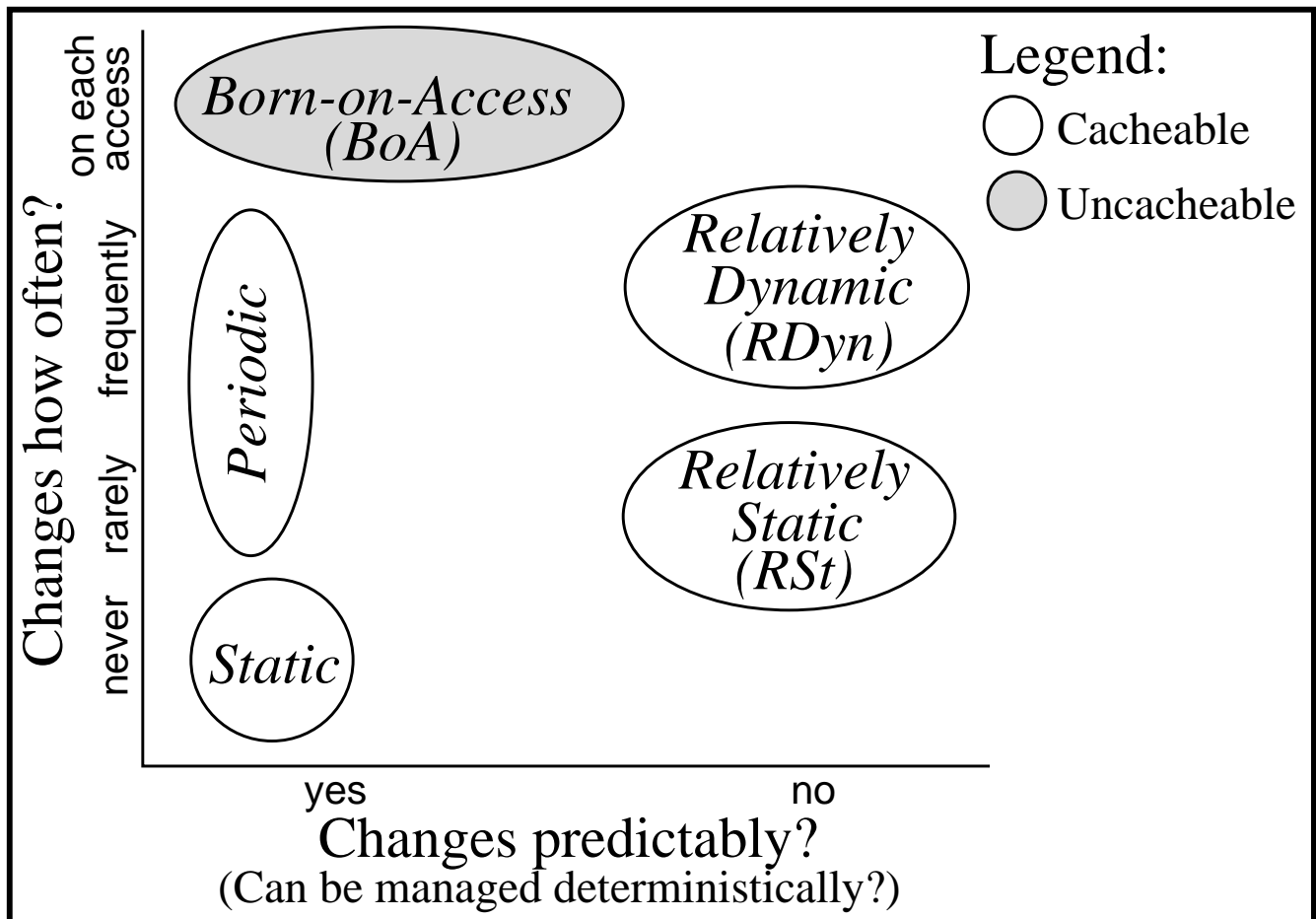
It would be nice to...

- know exactly how intermediaries handle content
- guarantee strong consistency
- eliminate unnecessary validations

MONARCH

- objects on a Web page have different *change characteristics*
- identify the most frequently changing object on the page—the *manager*
- validate the manager on every access
- server invalidates other objects on the same page (same *volume*)
- server relates explicit per-object instructions to caches by assigning each object an appropriate *Content Control Command (CCC)*

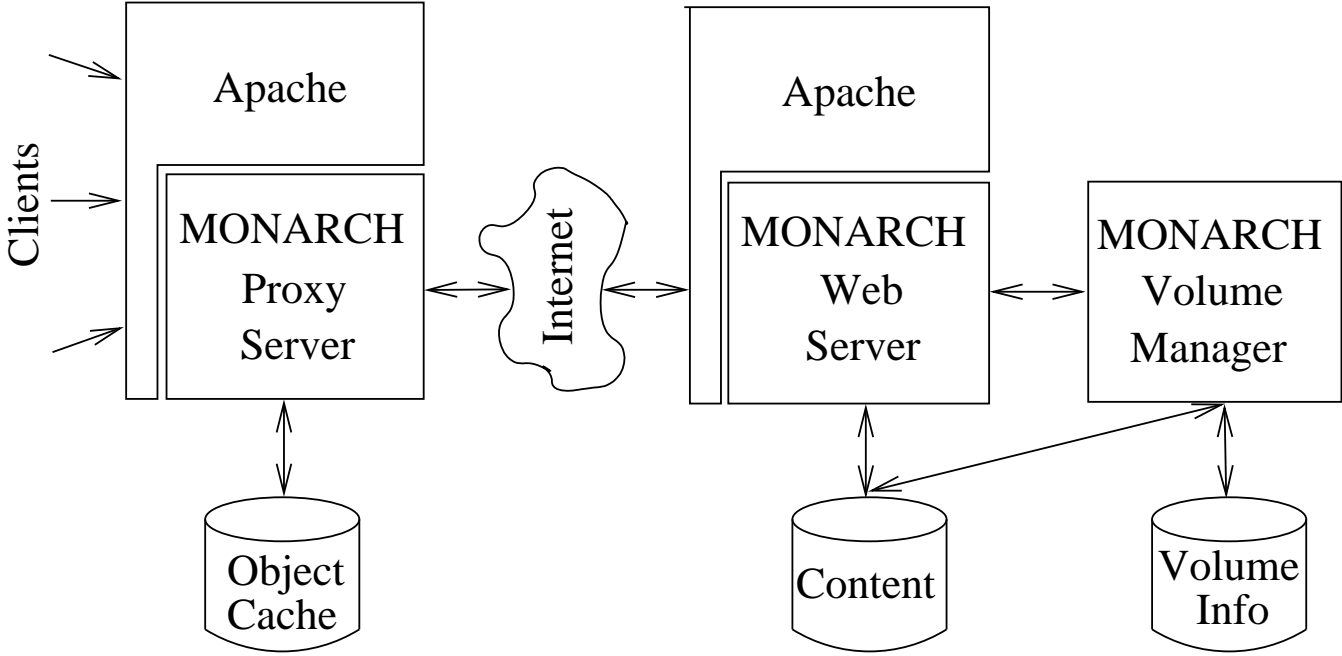
Object Change Characteristics



Feasibility

- servers already generate objects automatically based on measurable events or at regular intervals
- object type or its location within the file system may determine its change characteristic
- only the most popular objects require classification
- only frequently changing objects require classification
- default classification is Relatively Static (RSt)

Prototype Implementation



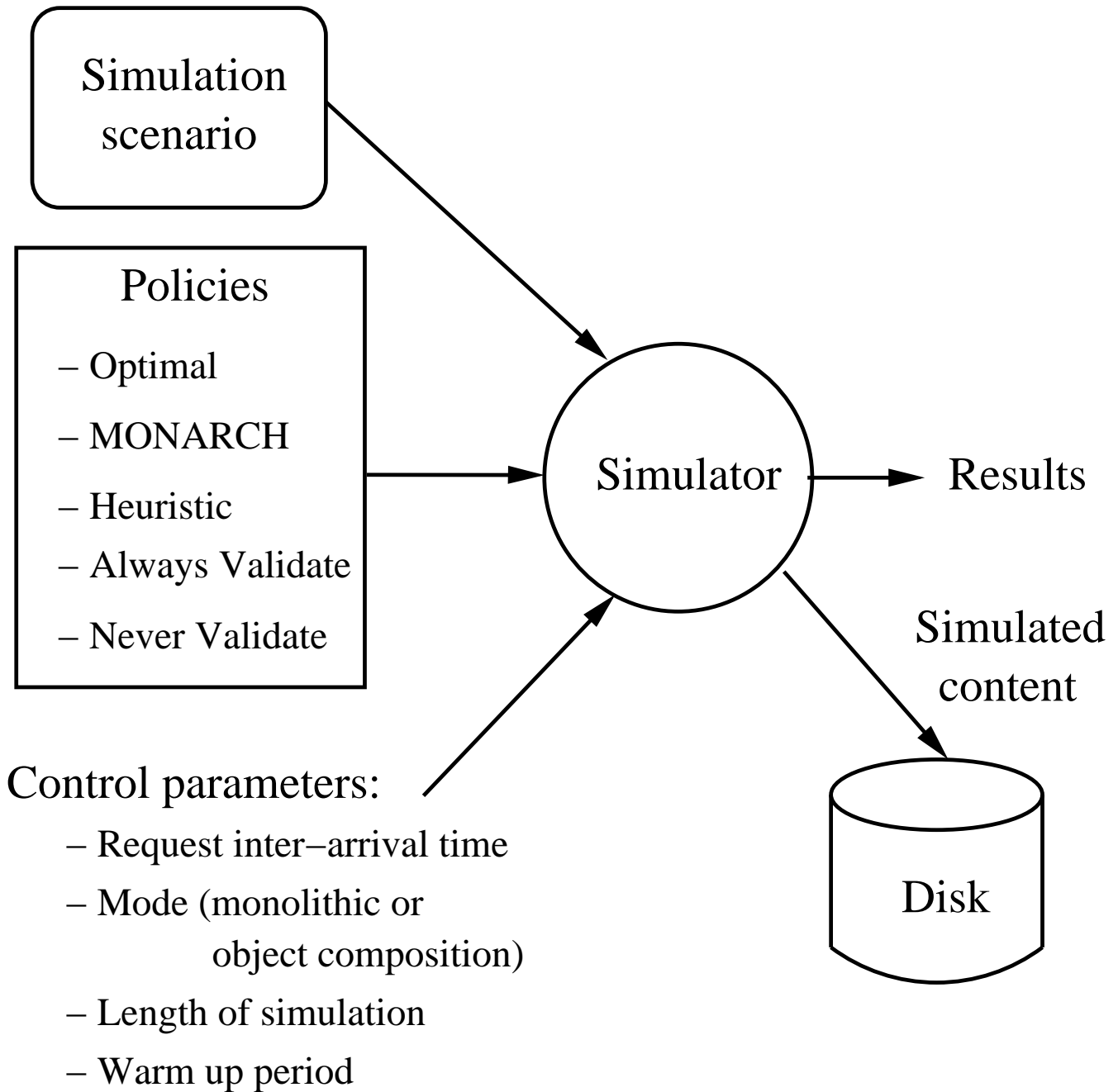
Performance Evaluation

Goal: compare MONARCH to policies that maintain no per-client server state

Metrics (averages per page retrieval):

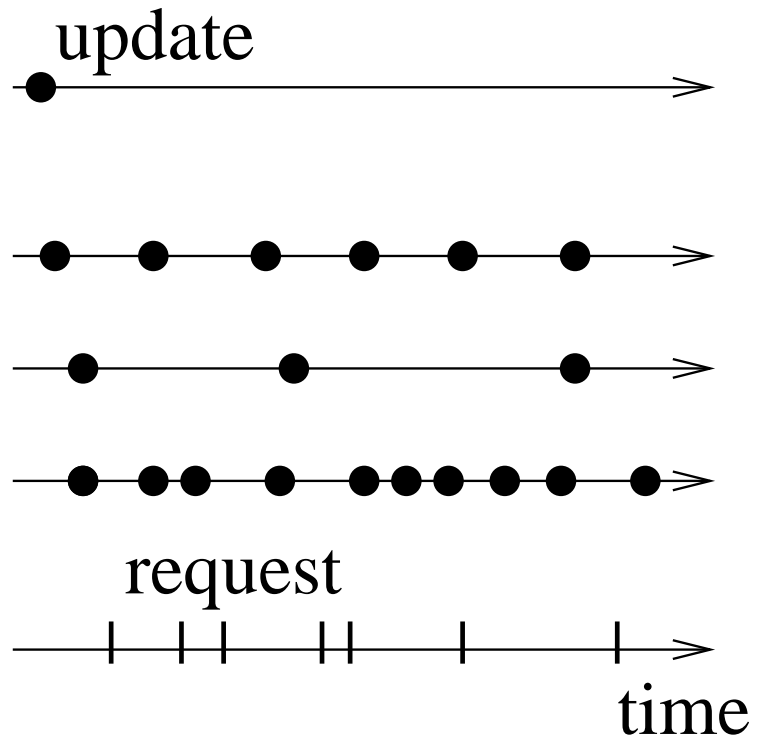
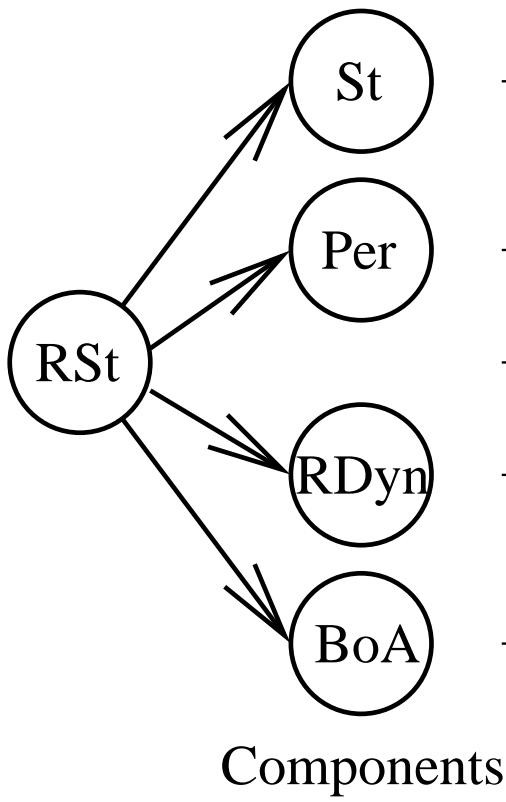
- stale objects served by the proxy
- requests served by the server
- bytes served by the server
- unnecessary validations processed by the server

Simulator

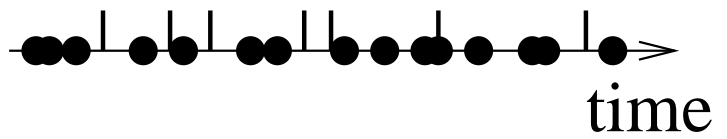
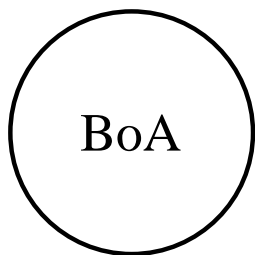


Simulator Modes

Object Composition

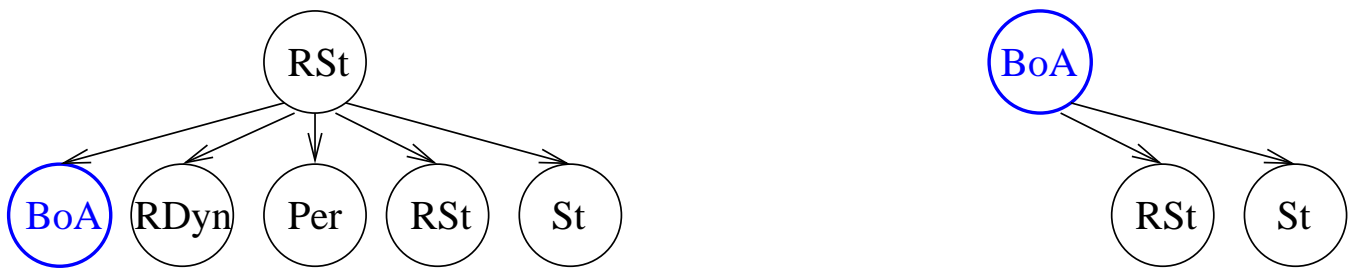


Monolithic

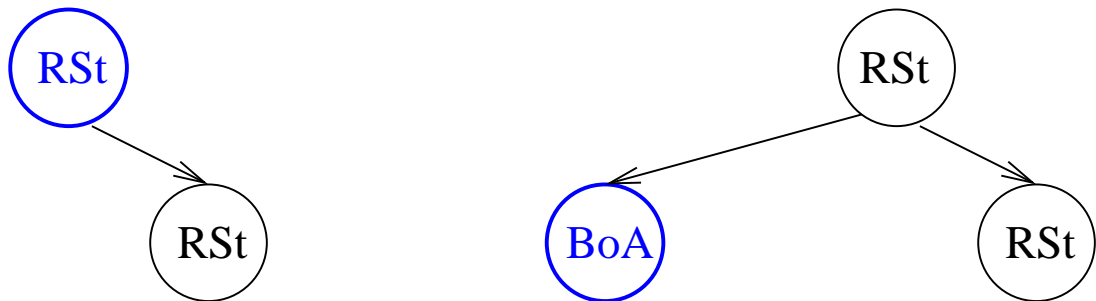


Simulation Scenarios

News Portal



User Home Pages

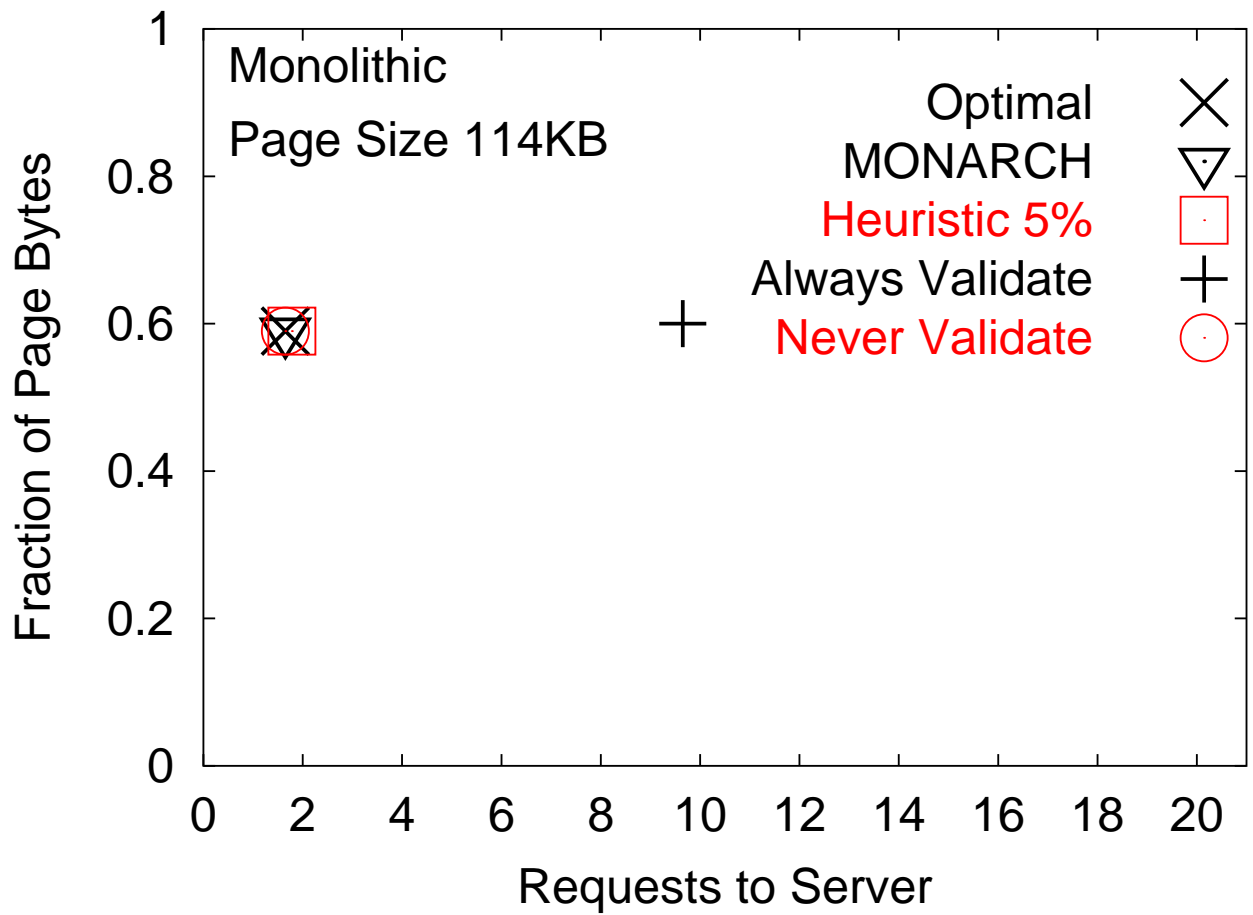
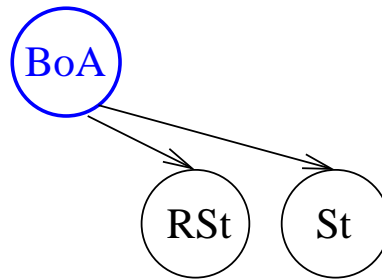


with Access Counter

Manager is shown in blue

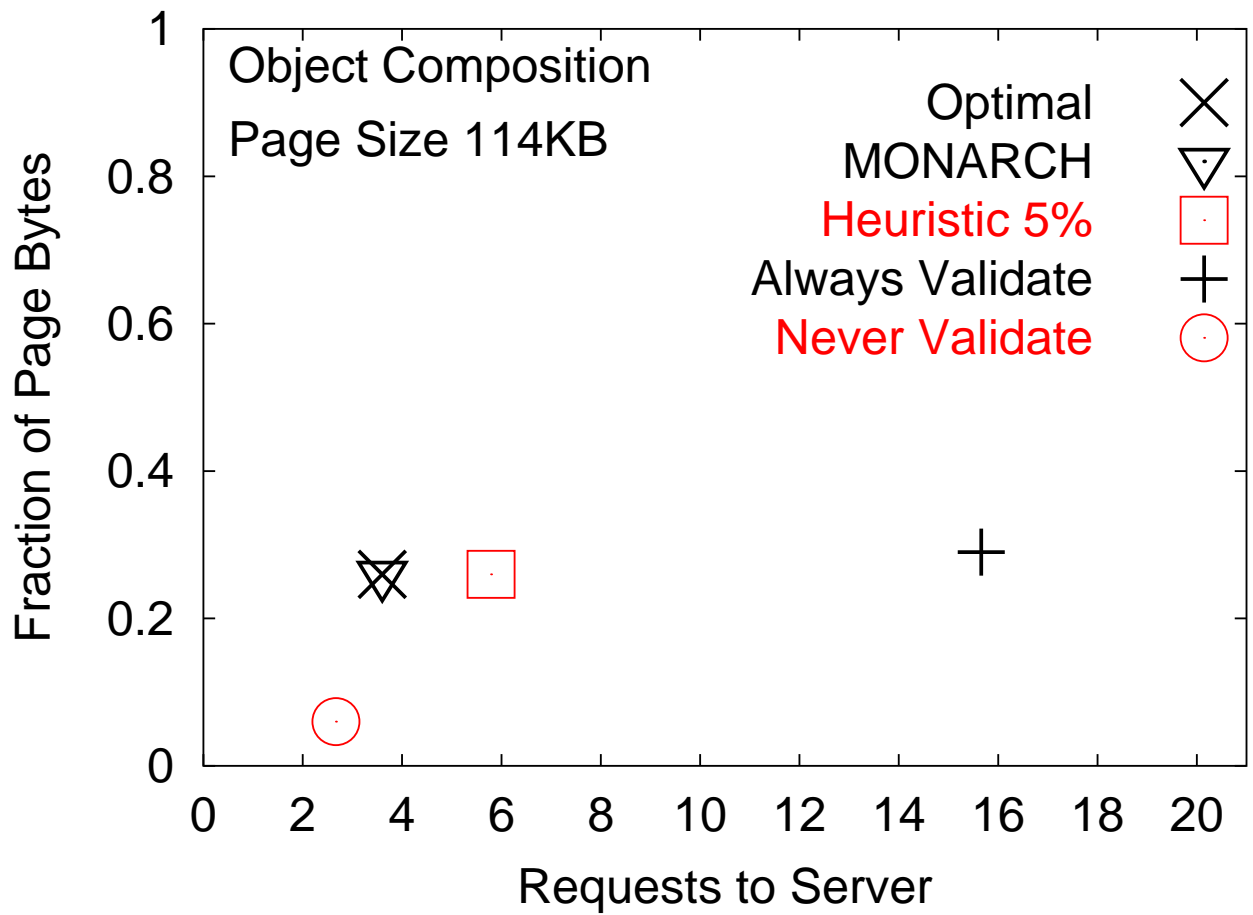
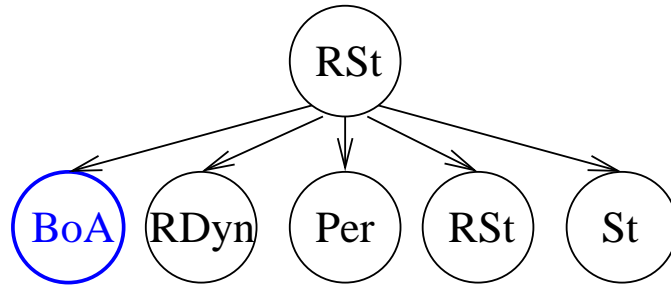
Results

News Portal



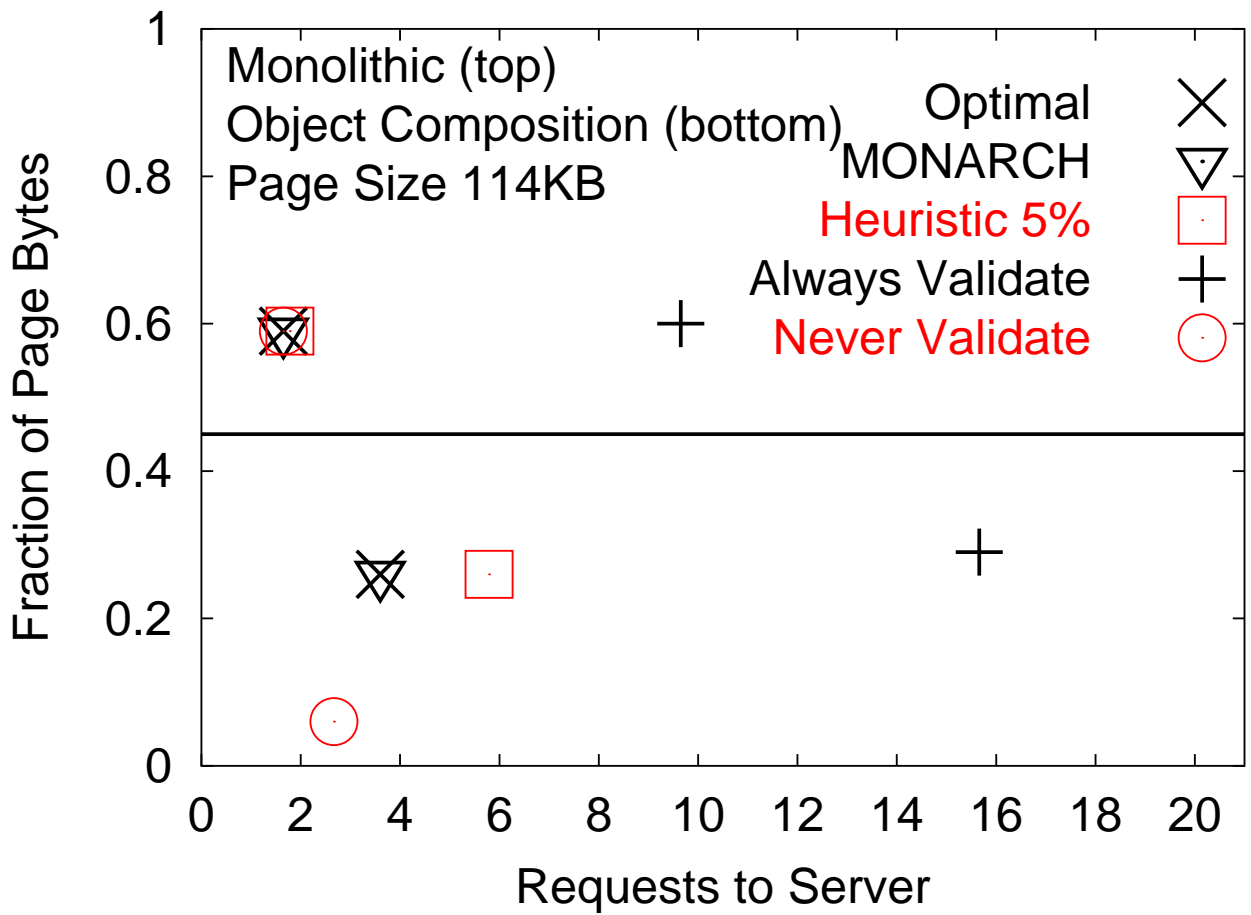
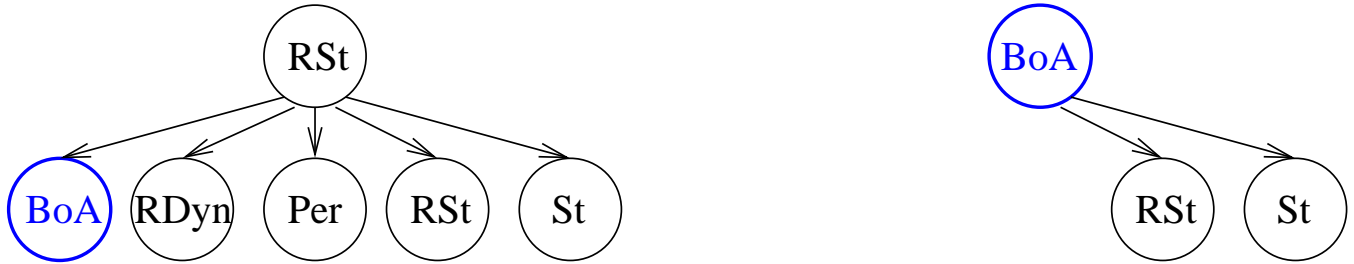
Results

News Portal



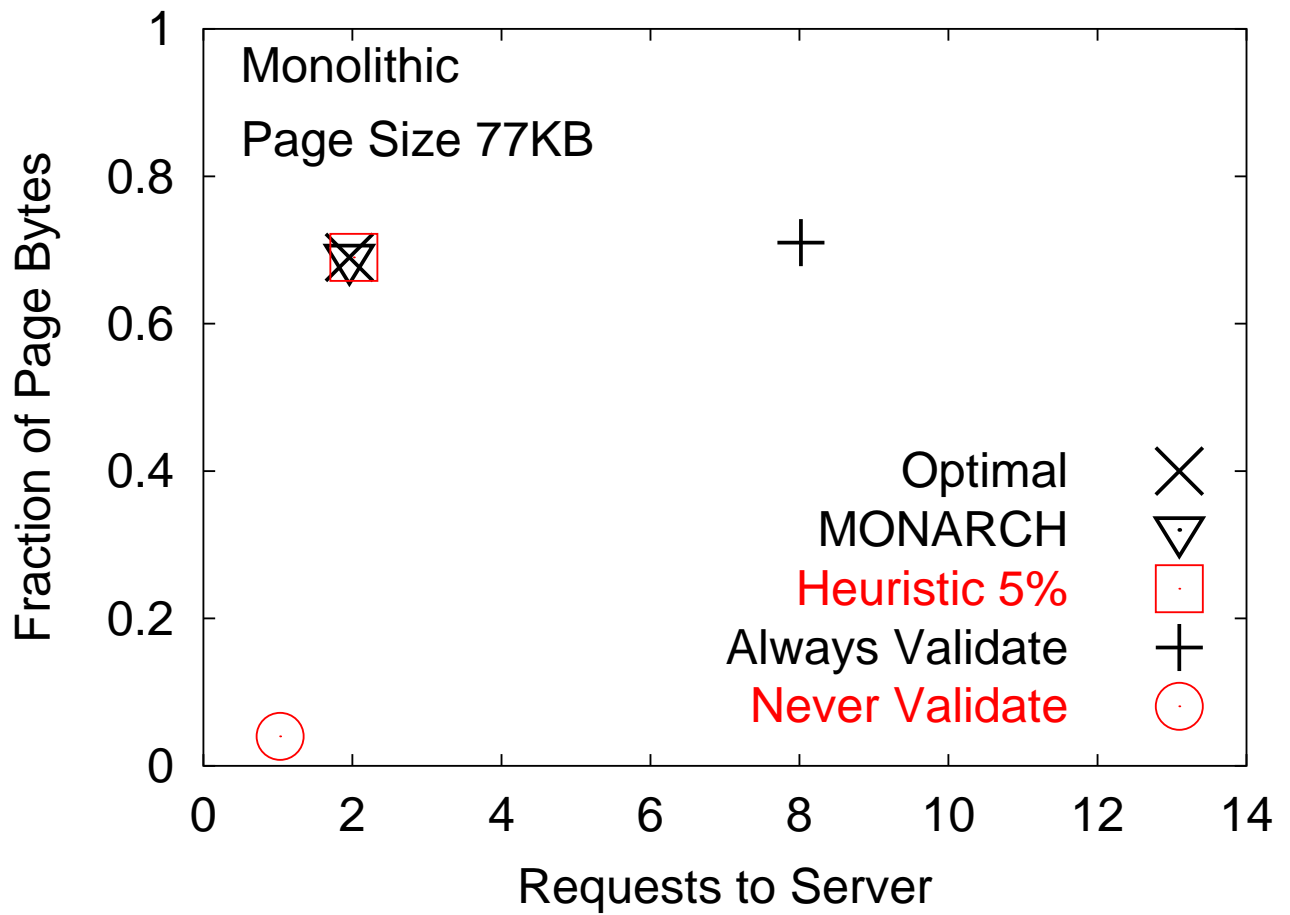
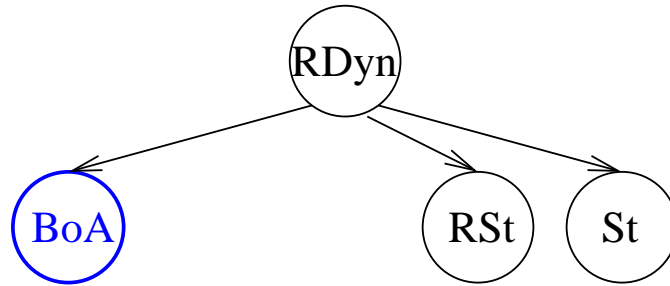
Results

News Portal



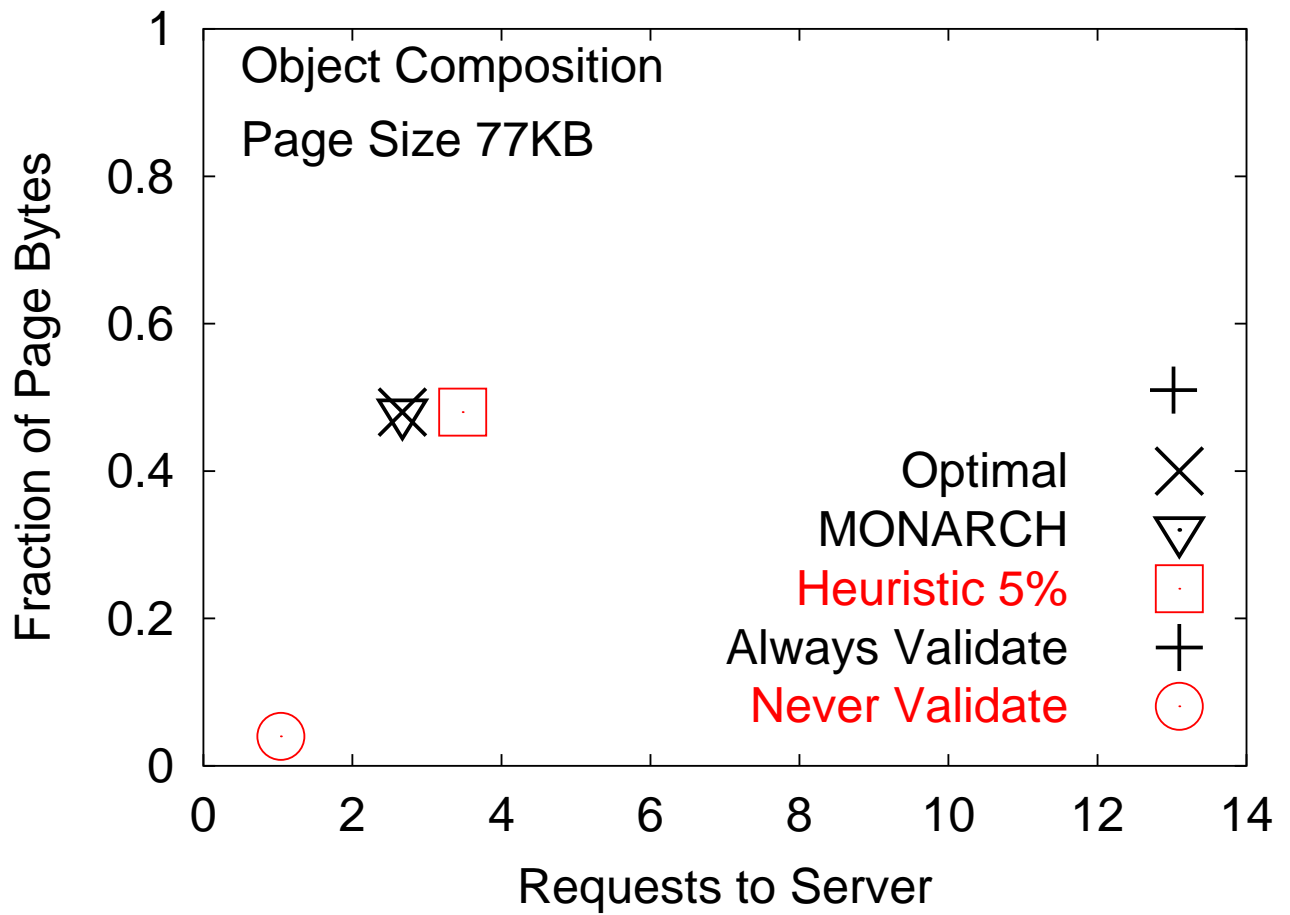
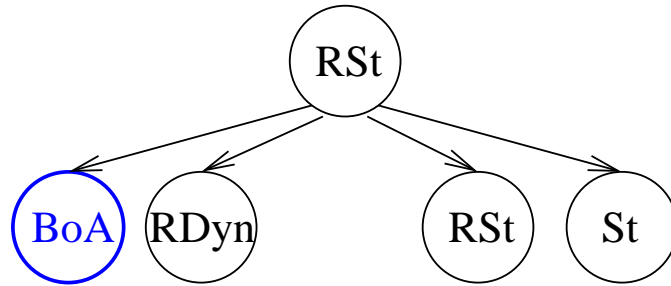
Results

Discussion Site



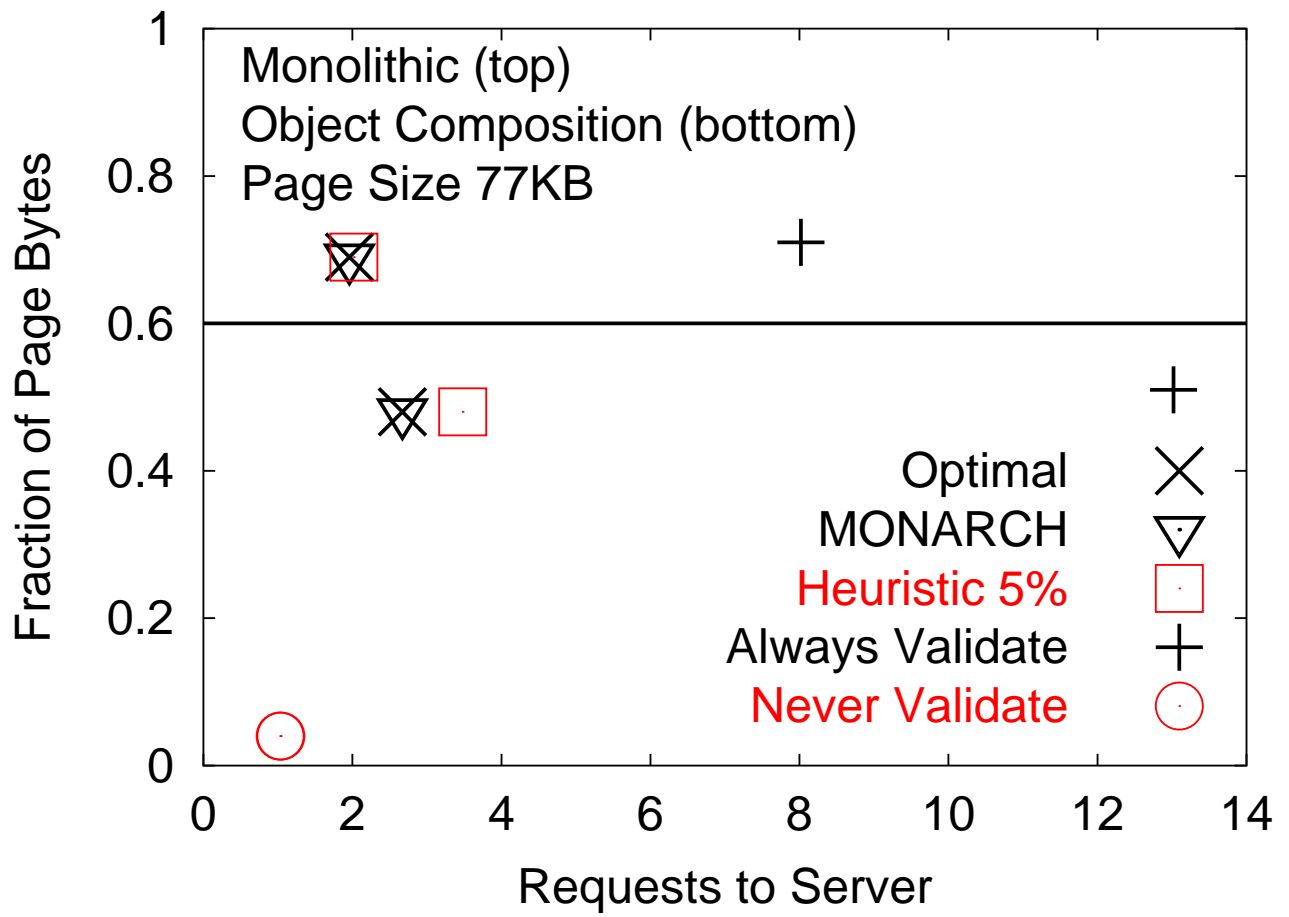
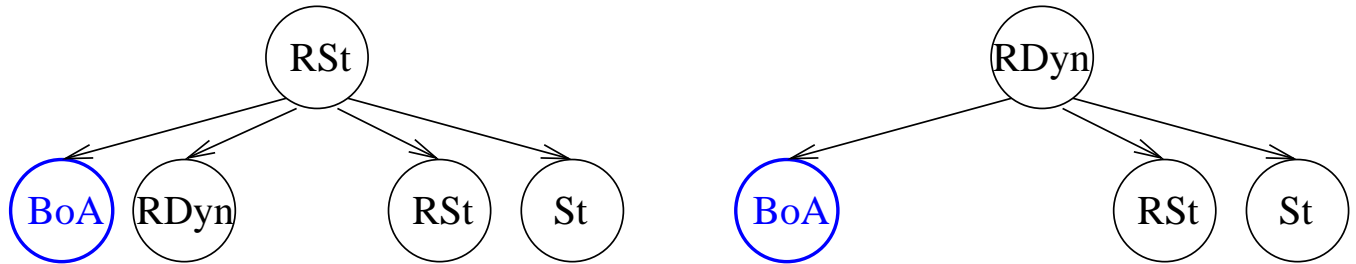
Results

Discussion Site



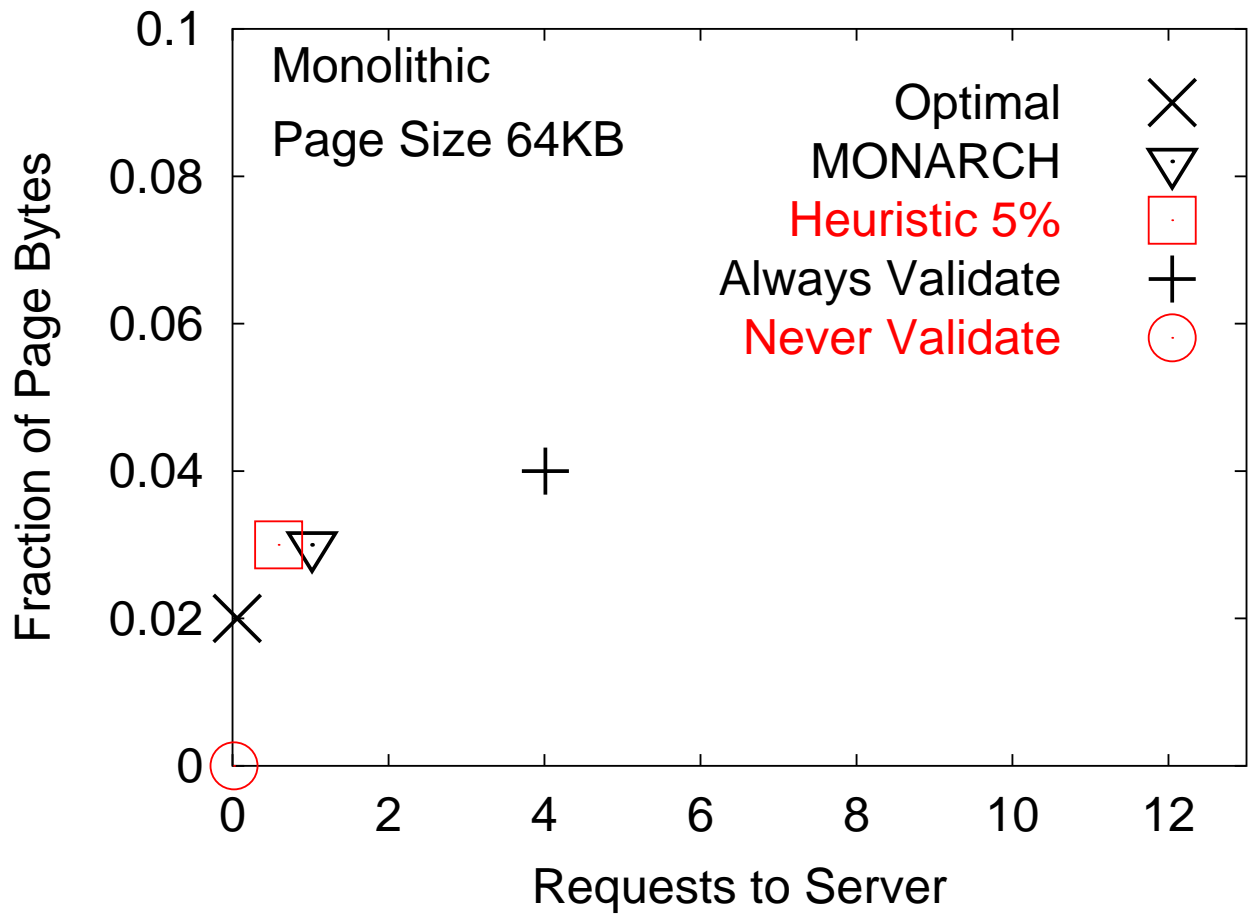
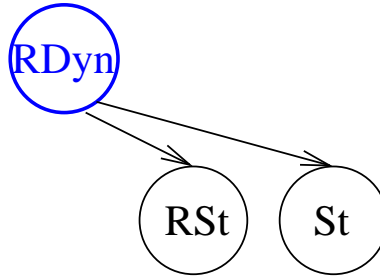
Results

Discussion Site



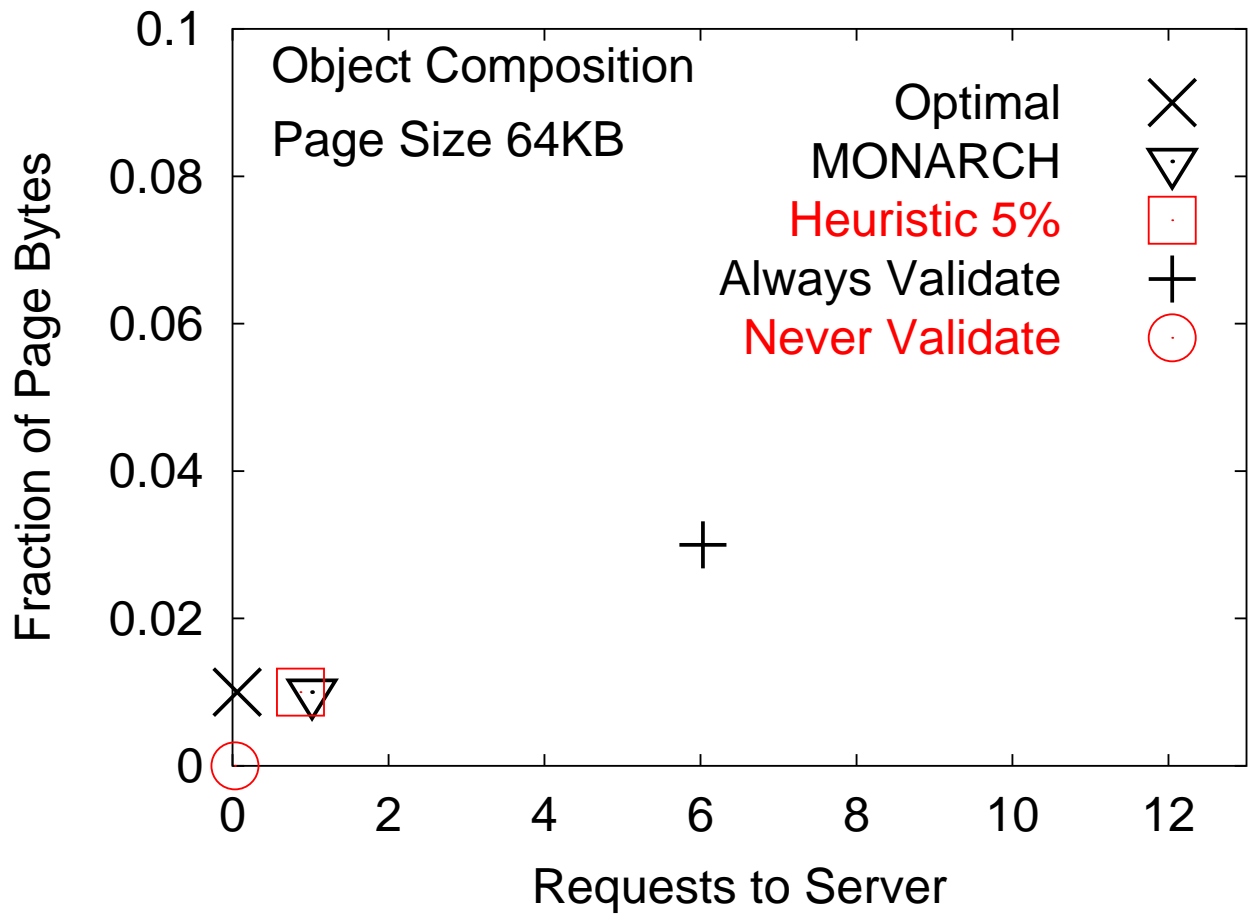
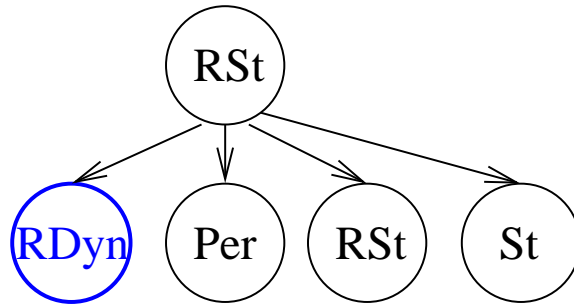
Results

Corporate Presence



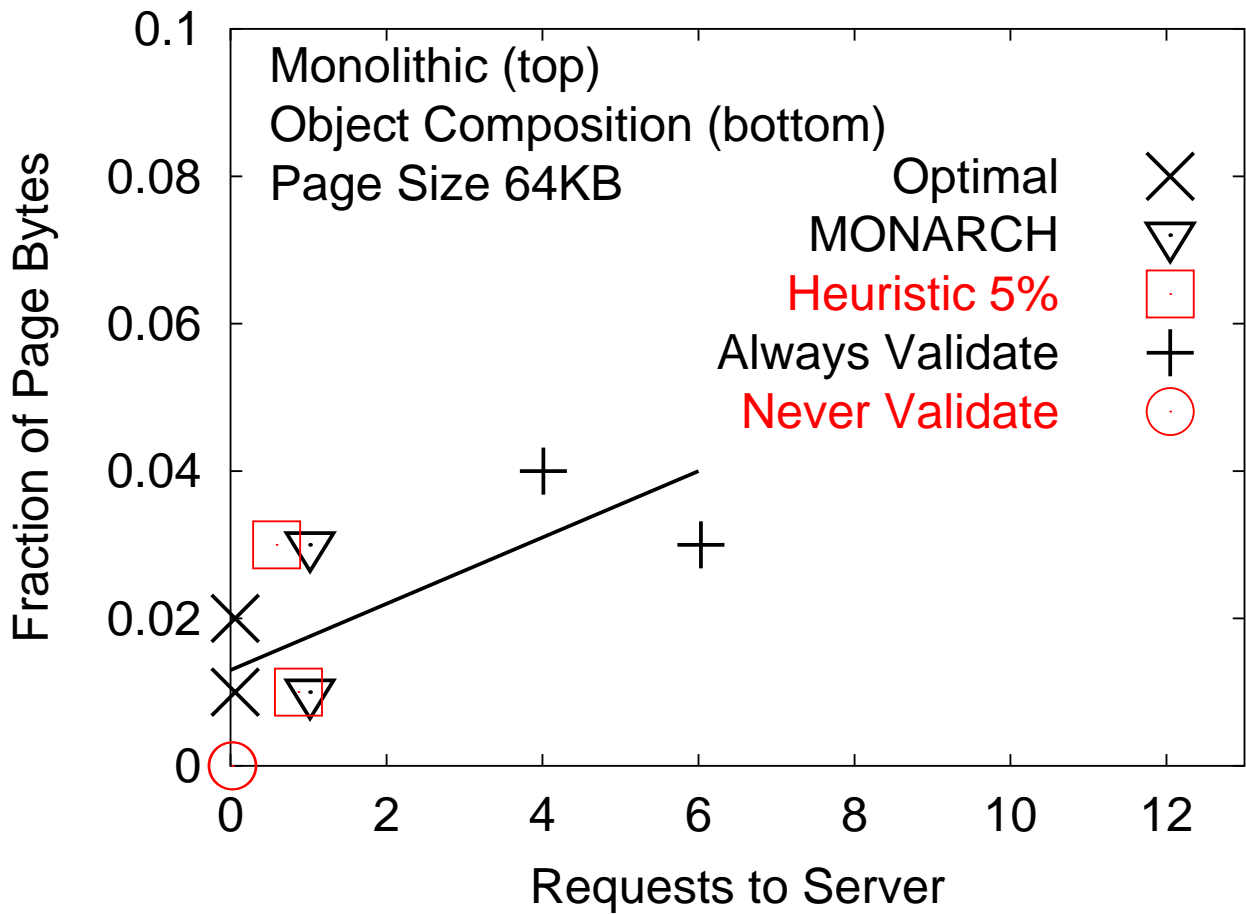
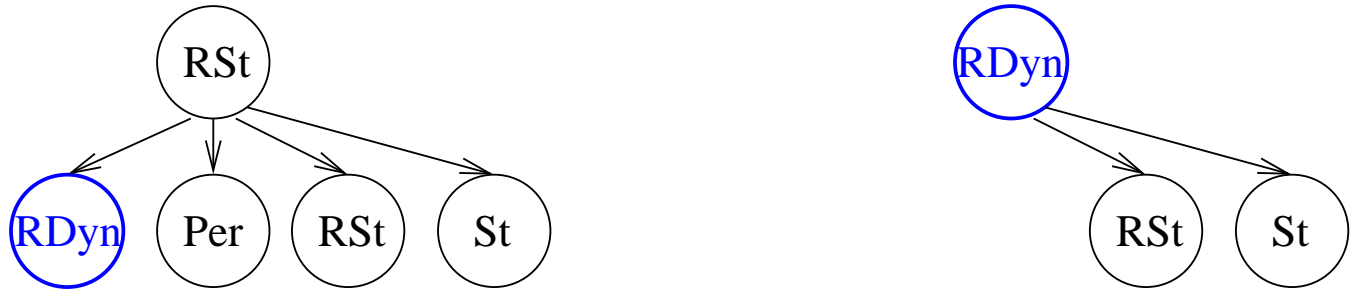
Results

Corporate Presence



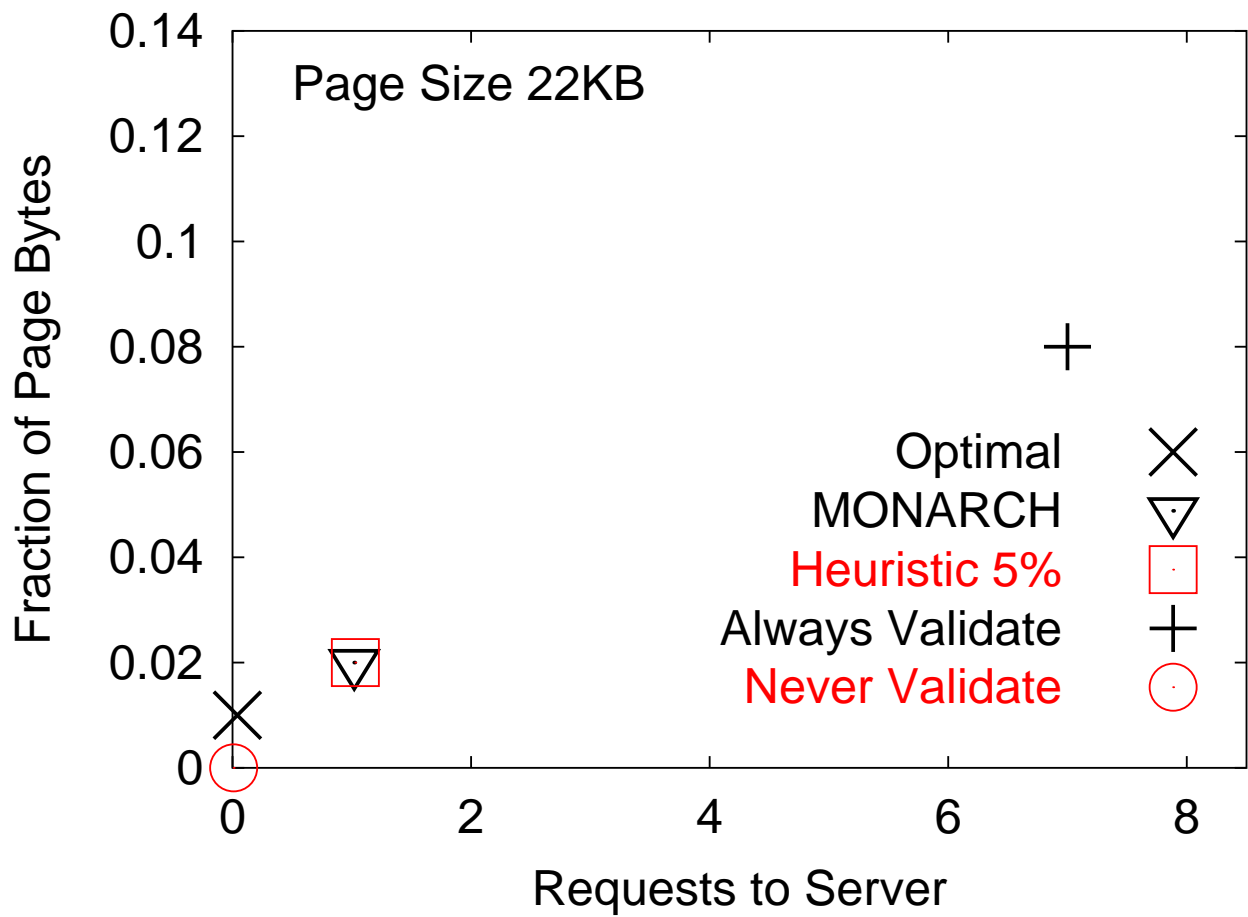
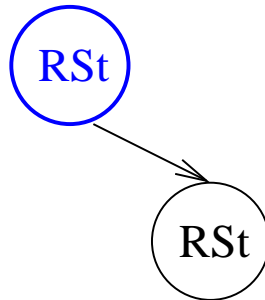
Results

Corporate Presence



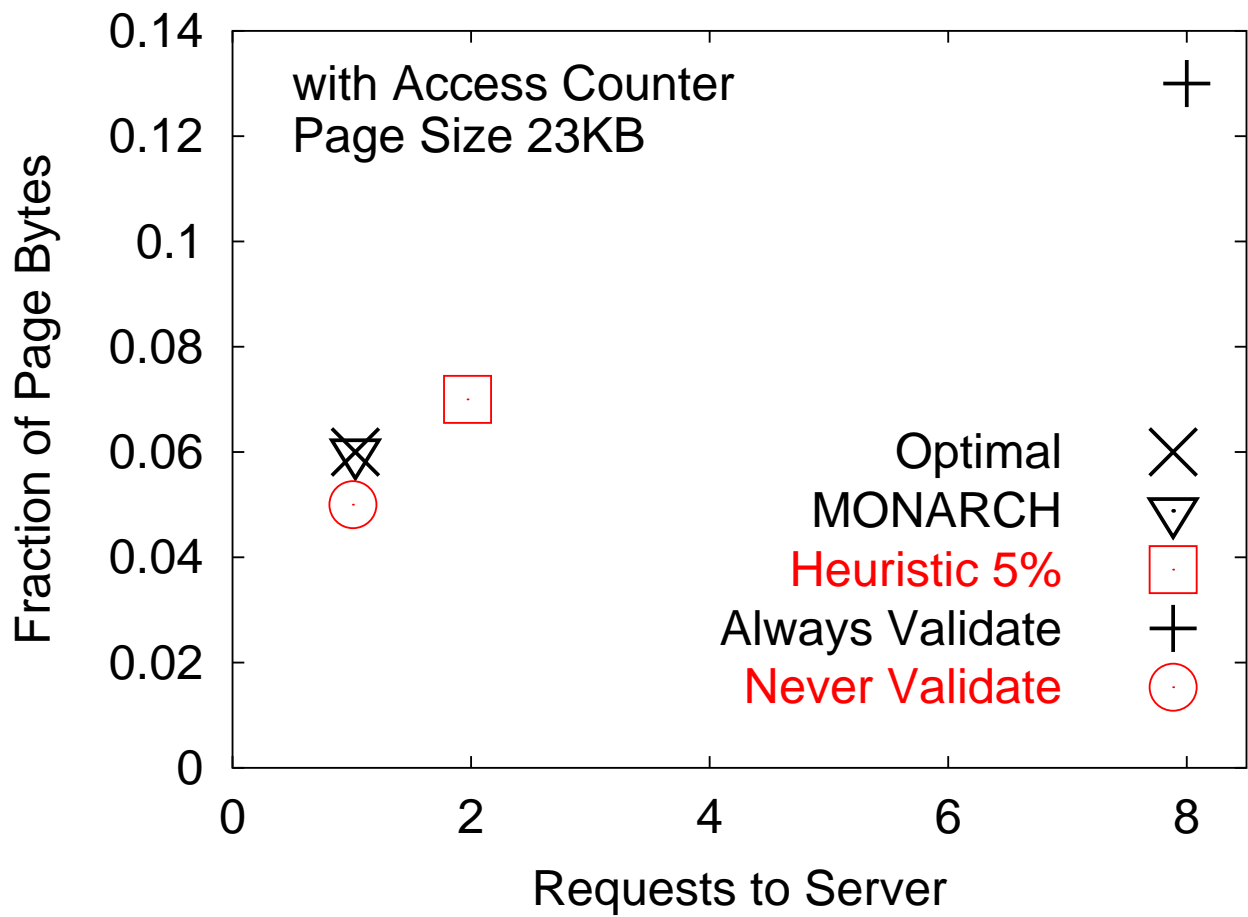
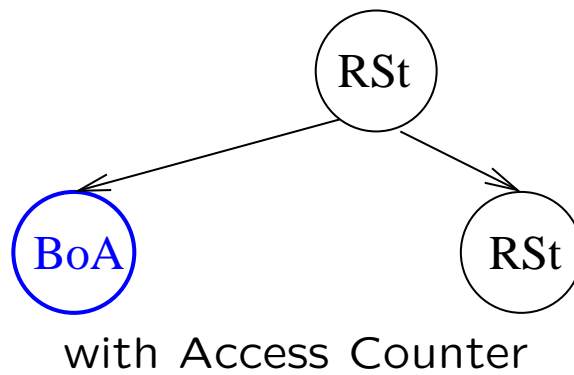
Results

User Home Pages



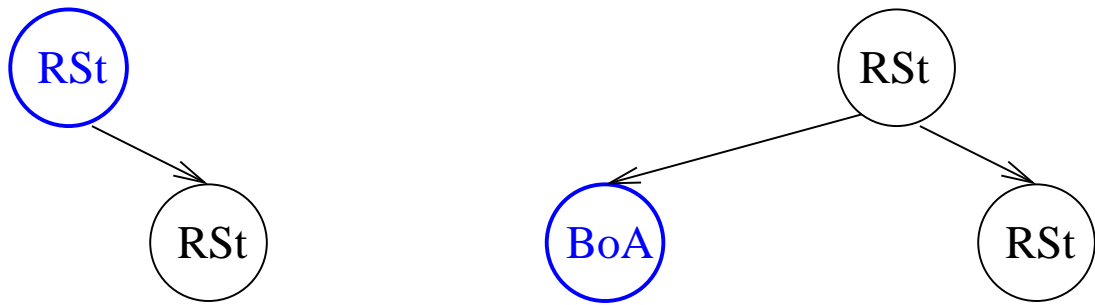
Results

User Home Pages

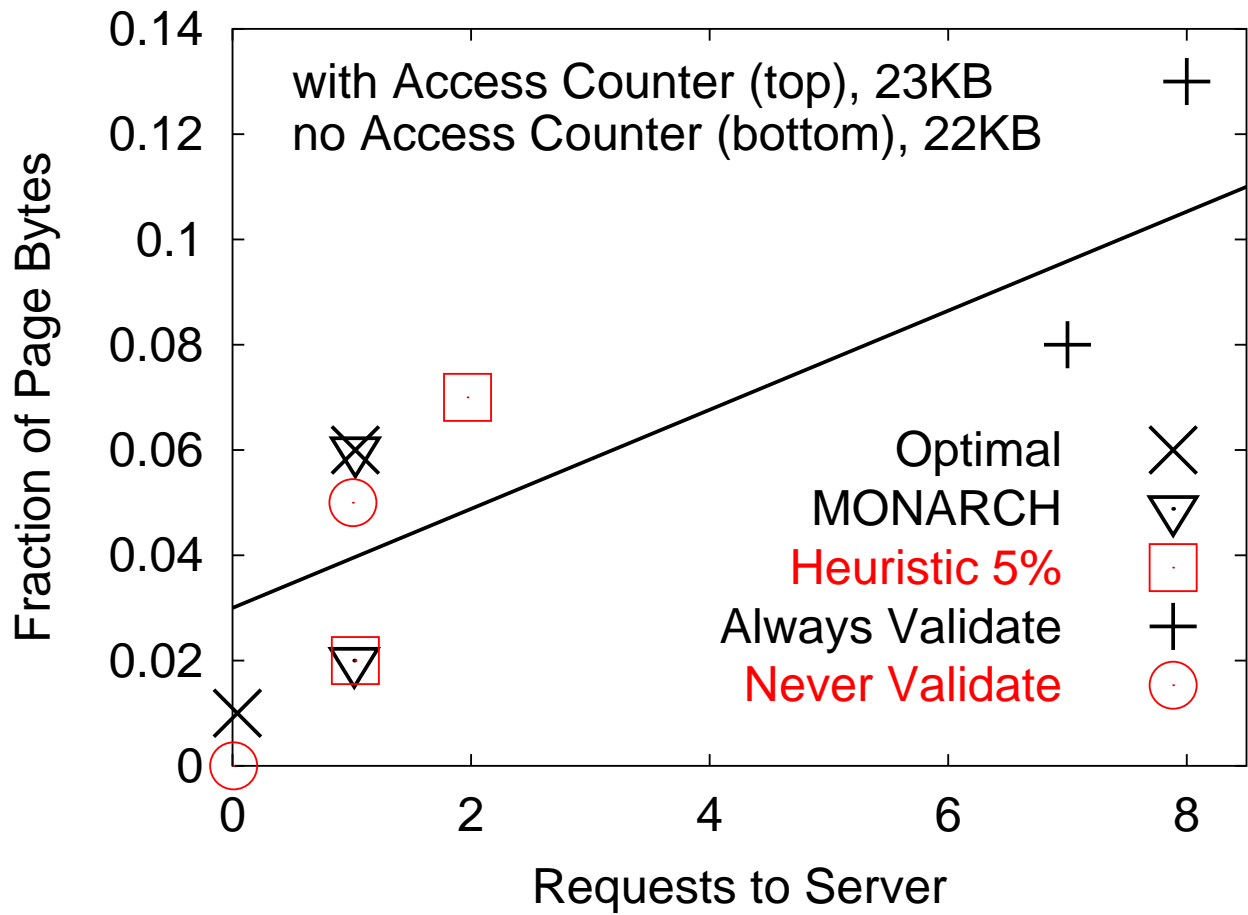


Results

User Home Pages



with Access Counter



Related Work

- Server Invalidation
- Object and Volume Leases
- Data Update Propagation
- Cachuma

Conclusions

- strong cache consistency can be achieved with no per-client server state and with less traffic than produced by heuristic approach
- exposing page structure to clients tends to generate more requests that reach the server, but clients can cache 30%–50% more bytes

On-going Work

- have the simulator drive the prototype system
- group shared objects into *global* volume
- add Server Invalidation, Object and Volume Leases
- use real content from Web sites to drive the simulator

For more information...

- mikhail@cs.wpi.edu
- <http://www.cs.wpi.edu/~mikhail>
- cew@cs.wpi.edu
- <http://www.cs.wpi.edu/~cew>