Exploiting Regularities in Web Traffic Patterns for Cache Replacement

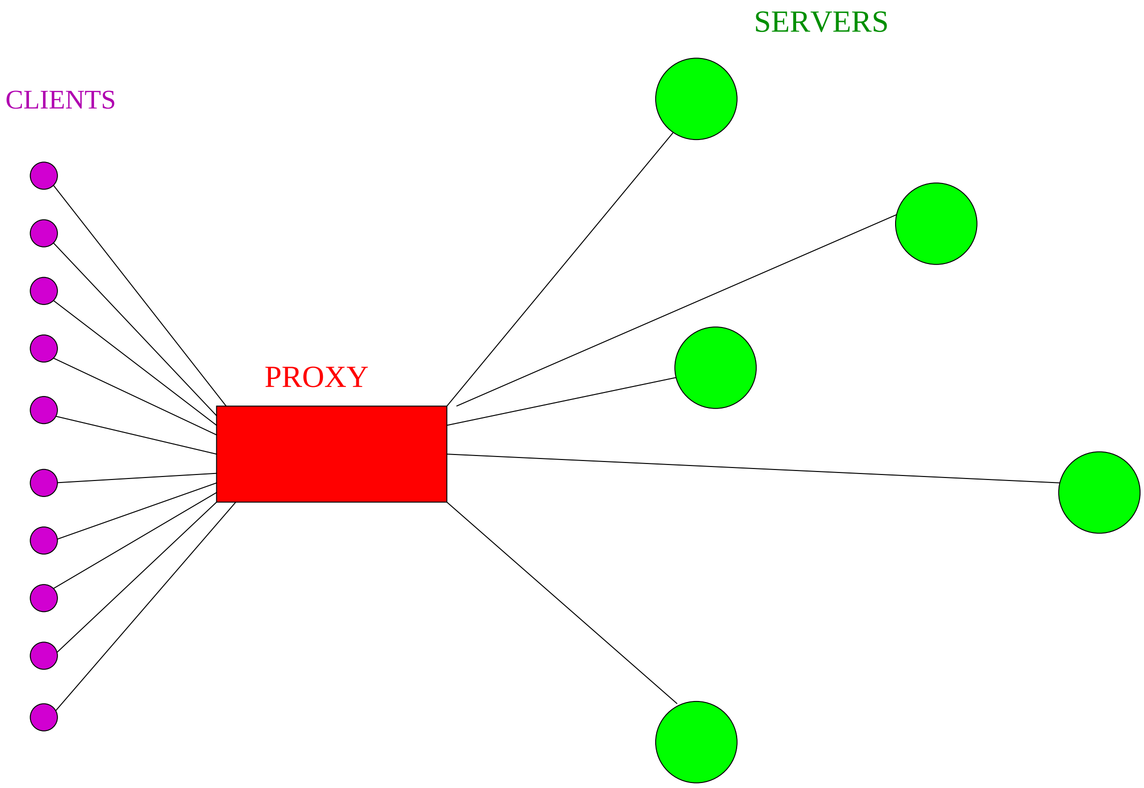
Haim Kaplan
AT&T Labs

Joint work with:

Edith Cohen
AT&T Labs

Motivation

- Improved algorithms for caching documents in Web proxies and servers.



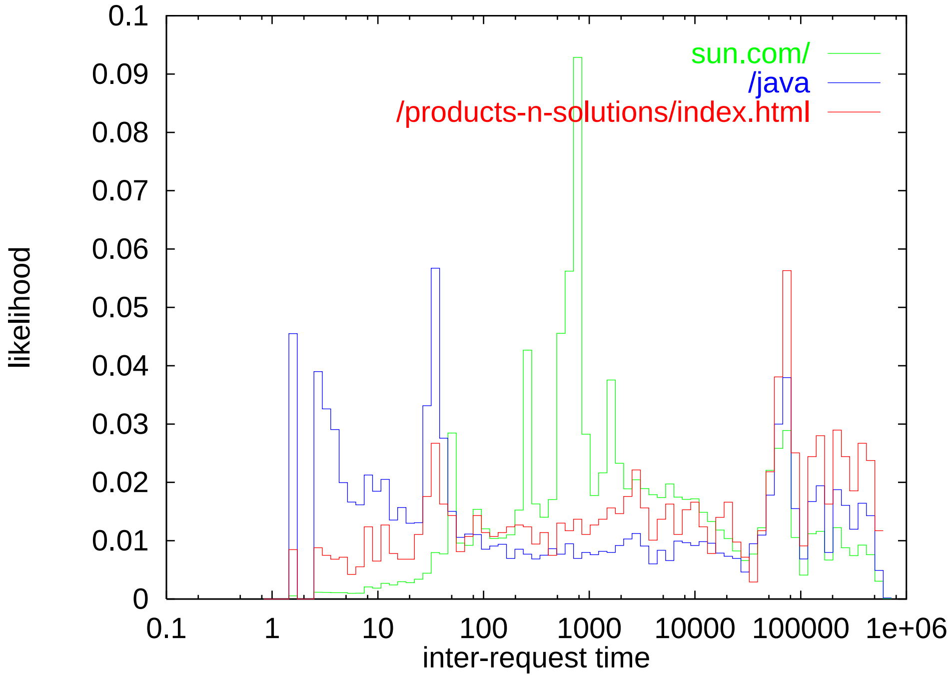
- Theoretical support for some experimental results.
(e.g. Cohen, Krishnamurthy, Rexford ESA'98)

What is different in caching Web data ?

- Pages have varying sizes and fetching costs.

- Pages are relatively static objects

Different request patterns:
Each page has a distinct distribution of inter-request times.



Model

Cache of size k.

When a page is requested and is not in the cache you fetch it and then may decide to cache it.
In order to cache it you may have to evict other pages.

Assume that with each request for a page p we get the distribution of the time to the next request of p.

This distribution is independent of forthcoming requests to other pages.

What strategy should we use ?

Previous work

• Lund, Reingold, Phillips (FOCS'94): Caching of IP-connections as ATM virtual circuits.

Suggested the median algorithm:
Evict the connections whose conditional distribution has largest median.

Median algorithm is 5-competitive.

The best online algorithm is hard to find. (Lund et al)

• With fetching costs and sizes the offline problem is NP-hard even when next request time is known (Irani).

There are competitive algorithms (Cao, Irani, Young).

Main Result

An online algorithm A such that

$$(1-\delta)(1-\epsilon)\text{OPT}(k) \leq A(k)$$

$$(1-\delta)\text{OPT}((1-\epsilon)k) \leq A(k)$$

if the input sequences satisfy certain conditions .

• Proving experimentally that our conditions are likely to hold for request sequences of Web documents.

• Technique: Link between tradeoffs of policies in two different caching models.

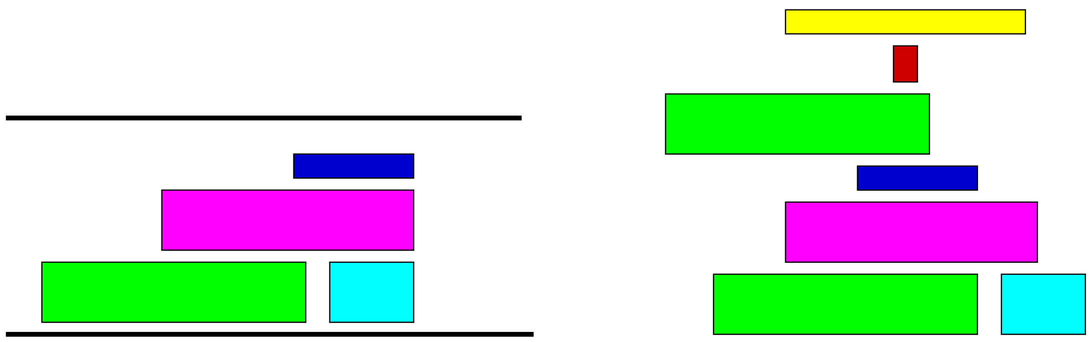
The interval caching model

After request r for page p
keep p in the cache for $T(r)$ seconds.

$cost = \sum_{requests} (cache\ \ time) * size$

if we get a hit on a page p the profit is
the fetching cost of p

Want to maximize the profit/cost ratio.



Optimal interval caching algorithm

To simplify, for the rest of the talk,
assume next request time is known.

Let Δ_r be the time to the next request of the same page.

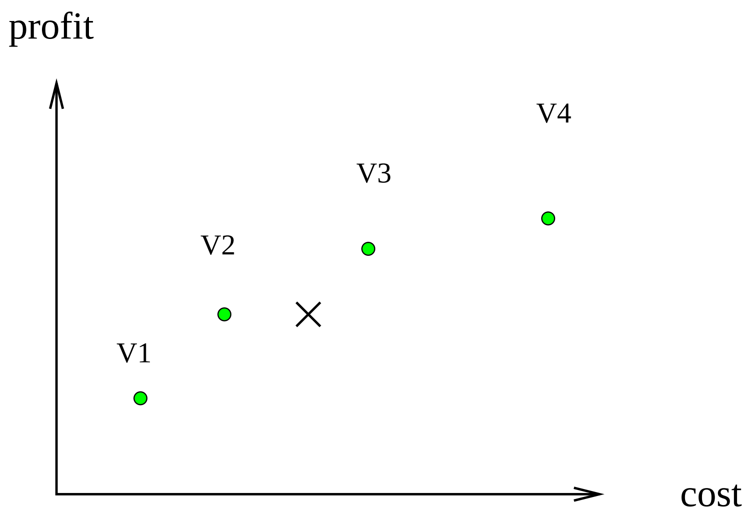
Algorithm $A_I(V)$ caches for Δ_r iff

$$\frac{size * \Delta_r}{profit} \leq V$$

For some fixed V.

Optimal interval caching algorithm *(continued ...)*

Like fractional knapsack.



Back to the regular model

Let A evict the page p with largest

$$f(p) \equiv \frac{size * \Delta_r}{profit}$$

Under which conditions profit/cost tradeoffs in the two models are related ?

• At any one time the sum of the sizes of items with

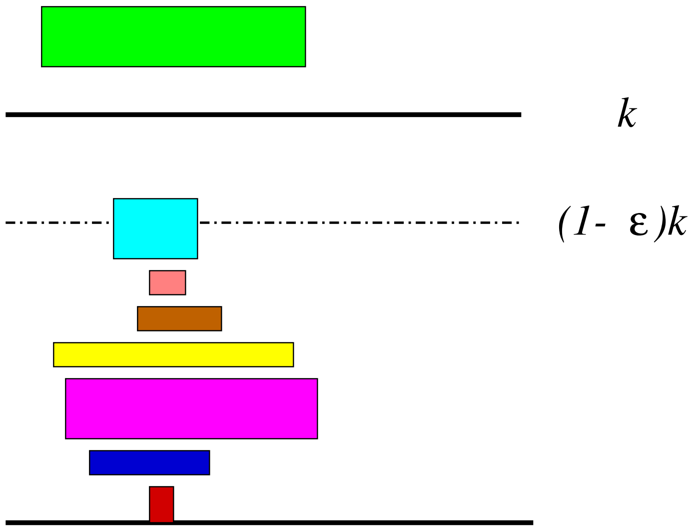
$$\frac{size * \Delta_r}{profit} \leq V$$

is approximately the size of the cache.

Linking between the models

We say that A is in ϵ-steady state for a request sequence s and cache size k if $\exists V(k)$ such that

- Page p out of the cache has $f(p) > V(k)$.
- The sum of the sizes of cached pages with $f(p) \leq V(k)$ is at least $(1-\epsilon)k$.



Linking between the models *(continued ...)*

On steady-state sequences:

$$\text{OPT}((1-\epsilon)k) \;\leq\; A_I(V(k)) \;\leq\; A(k)$$

$$(1-\epsilon)\text{OPT}(k) \;\leq\; A_I(V(k))$$

Proof:

Predictable load

The steady state assumption seems restrictive.

Everything generalizes using a more plausible predictable load assumption:

You have a price function $w(t)$ such that if you price time by $w(t)$ the sequence is in steady state.

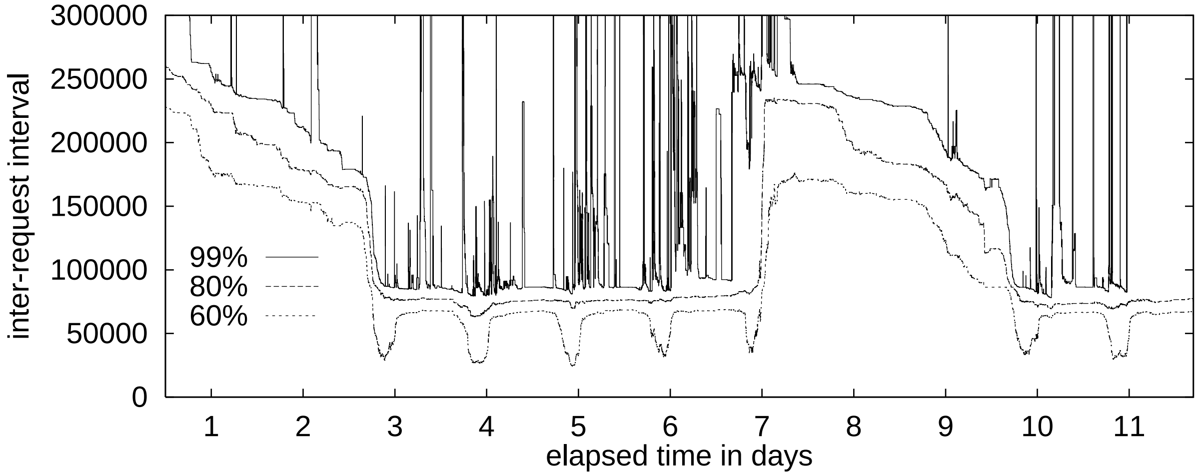
To generalize:

$$\Delta_t \equiv \int_{t_1}^{t_2} w(t)$$

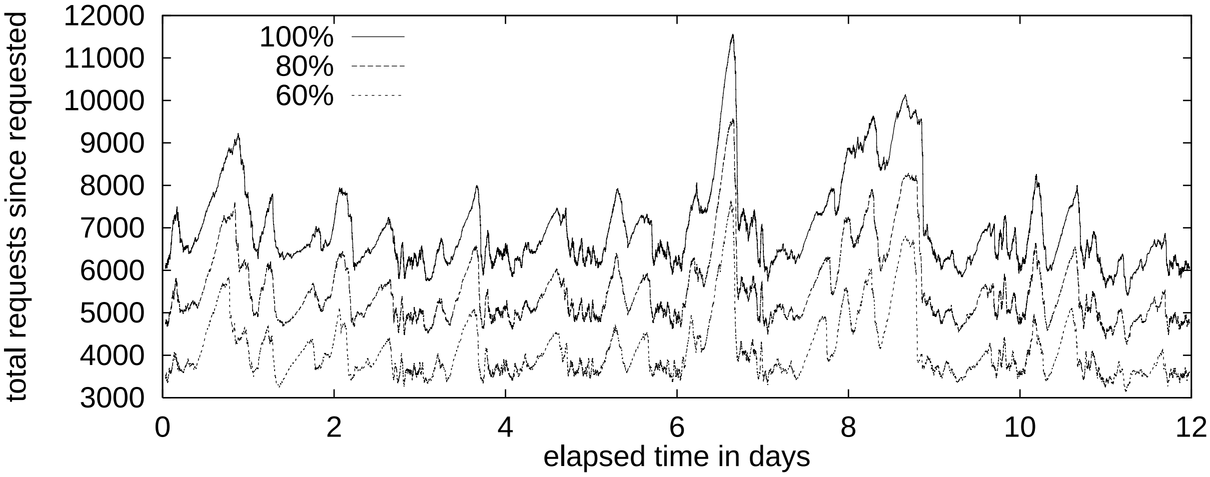
Predictable load *(continued ...)*

Many different users share a proxy.
Generate statistical behavior.

Predictable load seems to hold.



Predictable load *(continued ...)*



From a distribution to an interval

Let

$$\text{Eprofit}(t) \equiv F(t)$$

$$\text{Ecost}(t) \equiv \int_0^t (1 - F(x))dx$$

cache p for an interval of length Δ where

$$\Delta = \max\{z | \forall y < z \ \ \frac{\text{Ecost}(z) - \text{Ecost}(y)}{\text{Eprofit}(z) - \text{Eprofit}(y)} \leq V\}.$$

From a distribution to an interval *(continued ...)*

Equivalently,

Let $t_0 = 0$

and let t_i be the largest z that achieves the following minimum

$$\min_{z > t_{i-1}} \frac{\text{Ecost}(z) - \text{Ecost}(t_{i-1})}{\text{Eprofit}(z) - \text{Eprofit}(t_{i-1})}$$

Let $score(t_i)$ be the value of the minimum.

An alternative definition of the algorithm is to cache p for an interval $[0, t_i]$ for the maximum i such that $score(t_i) \leq V$.

Summary

Solution generalizes to the more
general settings of a known inter-request time distributions.

Good experimental results in CKR'98.

Further study:

- To what extent
predictable load conditions hold ?

- Experiment using calculated price functions.