Market-Oriented Data Distribution for Information System Survivability

Terence Kelly

tpkelly@rand.org
RAND Corporation
1700 Main Street
Santa Monica CA 90407-2138

tpkelly@eecs.umich.edu
http://ai.eecs.umich.edu/~tpkelly/
U. Michigan AI Lab
1101 Beal Avenue
Ann Arbor MI 48109-2110

1 September 1998

$Id: modd.tex 1.9 1998/09/01 10:37:07 tpkelly Exp $

CREDITS

Project leader: Bob Anderson

Guidance from Professors Michael Wellman, Jeff Mackie-Mason, and Sugih Jamin of the University of Michigan

Valuable input from Laura Baldwin, Cyndie Gareleck, Iris Kameny, Ed Keating, Dick Neu, and Ellen Pint of RAND

Cache trace data courtesy National Laboratory for Applied Network Research (NLANR), NSF grants NCR-9616602 and NCR-9521745

Wardrobe by Paul Dryer

OUTLINE

1. Markets in the military
2. Markets in information systems
   - Example: PDP-1 time at Harvard
   - Example: UNIX charges at RAND
3. Military information systems
   - Replication for robust C2
   - Civilian analogy: USENET news
   - The need for markets
4. Markets for information survivability
   - Grand vision (vaporware)
   - Current research
   - Case study: cache management

MARKETS IN THE MILITARY: MYTHS

- The military allocates resources by fiat, not by markets
- Military culture favors discipline and C2 over local autonomy guided by indirect incentives
- The military will never use real money in resource allocation schemes

THE AIR FORCE USES MARKETS

Depot-level repair pricing implemented in 1992. Goal: provide incentives for making cost-effective repair decisions at local level, e.g. encourage use of central depot repair facilities when appropriate (wings normally prefer local repair for operational reasons).

Problem: price to customers of depot-level repair is artificially high.

Example: over 50% of price charged for some of the most expensive F-16 repairs is a surcharge unrelated to actual repair costs.

Result: wings have incentive to reproduce depot repair capabilities at local level, even when this increases overall Air Force costs.

Reference:
Laura Baldwin & Glenn Gotz, *Pricing Air Force Depot-Level Reparables*, DRU-1224/1-AF, June 1996.

THE ARMY USES MARKETS

Depots are required to charge full-cost prices for repairs, which include surcharges and *averaged* repair costs, rather than the actual cost of repairing a particular item.

Local facilities can charge lower prices closer to the true cost of repairs.

Result: artificial incentive to use local facilities for cheap repairs and to send most costly jobs to central depot.

Reference:
Marygail Brauner, Ellen Pint, Dan Relles, John Bondanella, and Rick Eden, *Modifying Credit, Price, and Surcharge Policies to Encourage More Cost-Effective Supply and Repair Decisions*, AB-157-1-A, September 1997.

MARKETS IN THE MILITARY: OBSERVATIONS

1. The military is willing to use markets to decentralize decisionmaking.

2. Local commanders respond rationally to internal prices.

3. Poorly-designed markets don't align local incentives with global goals.

4. Common phenomenon: average-cost pricing is misapplied to services with high fixed costs and low marginal costs.

5. "Downward spiral of death": smaller numbers of customers face rising prices due to annually-adjusted average-cost pricing.

EARLY CPU MARKETS

Marginal cost of CPU consumption is near zero, so goal of policy should be to keep machine running continuously & serving the most important jobs.

Early example of CPU rationing: auctions to reserve time on Harvard's PDP-1.

Users bid "yen" to reserve time slots, recover yen as soon as they use time. Budget constrains ability to reserve time.

Results:
High utilization: poor users get idle time.
Principled preemption: rich buy time from poor.

Reference:
I. E. Sutherland, "A Futures Market in Computer Time," *Communications of the ACM*, v. 11, n. 6, June 1968.

DOWNWARD SPIRAL AT RAND?

One CPU *second* on ULTRAsparc 2:
\$1.521 day/\$0.761 nite ⇒ \$93,110.40/day
Probably exceeds cost of new computer!
(Don't worry, Bob — I ran my simulations at Michigan.)

CPU prices
- reflect annual average full costs (cf. USAF & Army examples)
- insensitive to demand & priorities
- aren't RAND's fault!

Results:
- shared UNIX system underutilized
- projects reproduce capabilities of central system or adopt costly desktop systems ("every user a sysadmin")
- less bang for more bucks — taxpayers lose

Reference:
http://rww.rand.org/departments/os/guide/computing.addendum.html

CALL TO ARMS

RAND has the right combination of expertise in Computer Science and Economics to devise a sound CPU pricing policy.

RAND is in a better position than most contractors to change policy: Our job is to sell common sense to Washington.

Thousands of innocent CPU cycles are dying every millisecond. Stop the madness!

MILITARY INFORMATION SYSTEMS

Commanders at every level require data from a wide variety of distributed sources for situational awareness (e.g., operational orders propagate down chains of command, situation reports move upward).

Problem: Communication link failures due to physical destruction, congestion, and jamming can compromise the availability of critical data.

Solution: Push-driven active data replication as a hedge against the possibility of network partition. Producers push data out to receivers *before* demand for the data arises.

Example: Platoon leader broadcasts SITREP immediately instead of waiting for headquarters to request it, because network might not be there later.

Reference:
Iris Kameny, *An Approach to Replicated Databases for Robust Command and Control*, MR-669-A/ARPA, 1995.

CIVILIAN ANALOGY: USENET NEWS

Unconstrained "push" system: Any user can post an arbitrarily long message for worldwide distribution. Marginal cost to sender is zero.

Problem: "the tragedy of the commons" — negative externality at the receiving end, because disk space to store incoming news postings is finite (16 GB at Michigan) yet available at no cost to those who consume it.

Solution: filter incoming postings aggressively, purge old ones frequently. (At Michigan, discard 33% of incoming postings immediately, remove others after 7 days.)

Side effects: re-posting is common — wastes bandwidth, increases delay for readers.

LESSONS OF USENET

Giving away unlimited quantities of scarce resources for free is a bad idea.

Data replication schemes must address the negative externality at the receiving end.

Sender needs information about the scarcity of recipient's storage, as well as information about the relative importance of *all* messages intended for the recipient.

Solution: provide this information as *prices*. Real estate at receiving end should go to senders who pay most.

(I'm not going to tell you how to do this.)

MARKETS FOR SURVIVABLE INFOSYSTEMS: THE GRAND VISION

- *Dynamically allocate* scarce resources in accordance with statically-defined priorities (in the form of budgets, utility functions, price mechanisms)
- *Decentralize* both static decision problems and dynamic allocations
- Enhance survivability by enabling replication
- Encourage efficient resource usage by facilitating exchange of resources (analogy: bullets for gasoline)
- Maximize *value to users*

MARKETS FOR SURVIVABILITY: CURRENT RESEARCH

Columbia University
- Focus on security
- Key tool: different currencies for administrative domains
- Example problems: intrusion detection

University of Michigan
- Focus on dynamic resource allocation
- Key tool: automated auctions
- Example problems: task allocation, supply chain formation
- Prophecy: interest in supply chain work will increase after 1/1/00

The Economic approach to survivability is currently under-funded. Two groups nationwide don't constitute a critical mass for intellectual cross-pollenation.

CASE STUDY: CACHE MANAGEMENT

Scarce resource: disk space in shared caches, e.g. on the WWW.

Benefits of shared caching:
- Clients: lower latency & limited protection against network partition
- Servers: fewer requests
- Network: less traffic

Goal: Use markets to allocate scarce cache space so that clients can pay for faster service. Provide adjustable quality-of-service. Maximize *value to users* rather than hit rate or byte hit rate.

BOUNDARIES OF INVESTIGATION

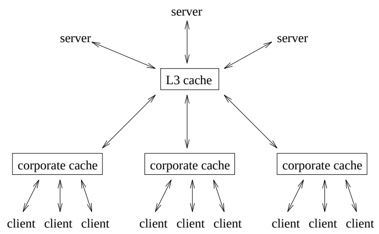
Consider only passive, client-pull-driven caching.

Reason:
Extensive trace data available. Relatively easy to retrofit market scheme onto existing pull-driven caches.

Do not consider
- Push from servers to cache
- Prefetching by cache

Reason:
Demand data not available from real replication or prefetching systems.

SHARED WEB CACHES



Level-one (L1) cache is inside individual client browsers. Campuses and corporations often operate L2 caches to reduce traffic on expensive Internet access lines. Currently there are at least ten L3 caches in operation nationwide.

CONVENTIONAL WEB CACHE MANAGEMENT

If requested URL is in cache ("HIT"), serve to client.

Otherwise ("MISS") retrieve URL from server, then serve to client and place in cache.

Remove URLs from cache to make room for new URL, e.g., kick out the "least recently used" (LRU).

MARKET-BASED CACHE MANAGEMENT

Associated with client requests for URLs are bids representing amount of money per block the client is willing to pay if a hit occurs.

Example client request:
"I want `http://www.nytimes.com/`
I'm willing to pay 3 yen/block* for a hit."

Service policy:
Serve request & collect client's money if offered price $\geq$ adjustable threshold.

Replacement policy:
Evict URLs that have generated lowest per-block revenue streams. Goal is to fill cache with URLs that make profitable use of the real estate they occupy.

*One block = 1024 bytes. We use blocks rather than bytes because disk space is allocated in blocks.

SETTING PRICE THRESHOLD

Trick: turn dynamic request stream into a static bid set.

Upon receipt of request for URL in cache, consider all requests for the URL to date, including current one. Pretend that all requests have arrived at the same time. Compute price which maximizes revenue.

Example:
Current request offers 5 yen/block.
Previous requests offered 2, 3, 7, 7, 9.
Price threshold is 7.
Request is not served.

Different URLs have different prices, which change over time.

Clients have incentive not to offer ϵ.

PERFORMANCE EVALUATION

Methodology:
trace-driven simulation using NLANR logs

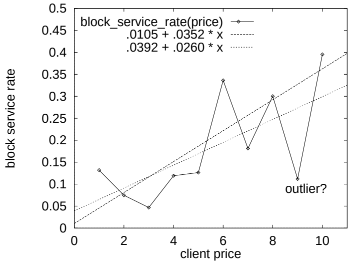
Client bid prices:
randomly assigned in range 1–10 yen

Measures of effectiveness:
- Block hit rate (BHR), traditional CS metric: fraction of blocks requested that are served from cache
- Value to users: revenue collected by cache
- BHR as function of clients' offered prices

TRACE DATA

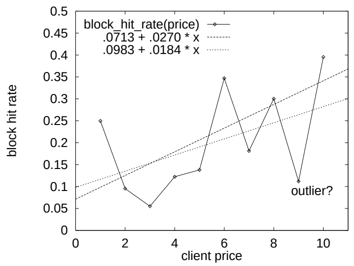
- L3 cache at NCSA in Urbana-Champaign
- Wednesday 12 August 1998
- requests from 57 "clients" (L2 caches)
- 516,139 requests for URLs
- 11,880,079 blocks requested
- Total size of all URLs: 5,690,810 blocks
- Maximum attainable hit rate: 30.4%
- Maximum block hit rate: 52.1%
- 447,548 blocks of cache required to attain max hit rates under optimal (clairvoyant) cache management policy
- Simulated cache size: 100,000 blocks

RESULTS



+ Pay 10X more $\Rightarrow$ 6.9X higher service rate
+ Overall block service rate is 15.8%
+ Total value to clients is 7,960,984 yen
— Price/service relationship noisy
— LRU: BHR 29.0%, value 11,491,717 yen
- One likely outlier: too few blocks requested

WHAT IF WE SERVE EVERY REQUEST?



+ Overall block hit rate is 22.2%
+ Total value to clients is 8,804,201 yen
— Pay 10X more ⇒ 2.5X higher BHR
— Still worse than LRU

Summary: this idea doesn't yet work well

MORE COMPARISONS

Policy	hit rate (%)	block service rate	value to users (yen)
LRU	17.8	**29.0**	**11,491,717**
LHR	15.7	22.4	8,030,451
LBHR			
SIZE	18.9	7.8	2,826,490
TOGGLE	**19.2**	12.5	4,722,104
MARKET	15.3	15.8	7,960,984
MARKET′	15.3	22.2	8,804,201

Explanation:
LRU: evict least recently used URL
LHR: remove URL with lowest hit rate
LBHR: remove URL with lowest block hit rate
SIZE: evict largest URL
TOGGLE: alternately apply LRU and SIZE
MARKET: remove low-revenue URLs, refuse low requests
MARKET′: same as MARKET, except serve all requests

BACK TO THE DRAWING BOARD

Exploit a new kind of regularity:
Clients with similar tastes for content have similar incomes.

People who read *Wall Street Journal* earn more than readers of *National Enquirer*, and presumably are willing to pay more for improved service.

Therefore shared cache should contain the *Wall Street Journal*. Fill the cache with the URLs that the big spenders want.

Assign bid prices to clients in such a way that clients who request similar sets of URLs have similar bid prices.

Can synthesize bid prices using voodoo; early attempts have failed. Ideally, use *real data* on client incomes from AOL, MSN.

BACK TO SURVIVABILITY

Push- and pull-driven data systems are disjoint, e.g. USENET and Web caches don't share disk. Result is sub-optimal value to users.

Integrate active replication and passive caching within unified market framework to solve negative externality problem that would otherwise hinder effective data replication, and allocate disk across push- and pull-driven mechanisms.

Example: In military infosystems, disk devoted to cached on-line equipment manuals during calm periods, because aggressive replication unnecessary when likelihood of network failure is low. During times of crisis, disk devoted to replicated OPORDs, SITREPs, etc.

LONGER-TERM GOALS:

Manage all major computing resources with markets (CPU, disk, bandwidth). Instead of using crude quotas or priorities to allocate these resources, simply assign fixed budgets to tasks or human users, who locally maximize utility.

Facilitate automatic exchanges among resource owners (analogy: bullets & gasoline).

Automatically deploy resources so as best to serve the mission.