Ginseng: Market-Driven Memory Allocation

Orna Agmon Ben-Yehuda¹  Eyal Posener¹
Muli Ben-Yehuda¹,²  Assaf Schuster¹  Ahuva Mu’alem¹,³

¹Department of Computer Science
Technion — Israel Institute of Technology
and
²Stratoscale
and
³Ort Braude

VEE, March 2014
What is the future of the IaaS Cloud?

- Fine resource granularity
- Fine time granularity
- Market-driven resource pricing
- Tiered service levels (like spot instances)

We predicted that these trends will culminate in a new cloud model, the Resource-as-a-Service (RaaS) Cloud.

More details in:


Maximize social welfare (sum of guest valuations=benefit)
  - Prefer production over debugging
  - On-line trading—prefer guests that have more expensive deals done

Pareto-Efficient (any guest’s allotment cannot be improved without hurting another guest)

We focus on the hardest resource of all—memory.
In a real commercial cloud the host and guests belong to different economic entities.

- **White box** approaches cannot work in a real commercial cloud.
  - What is the guest doing? What should be measured? How?
  - How much is the performance worth to the client?
  - Whose fault is it that the guest’s performance is low? Maybe the software is inefficient?

- **Black box** approaches cannot work in a real commercial cloud.
  - Host measurements: memory stress can be faked to induce the host to allocate more memory.
  - Guest measurements: results can be mis-reported.

Ginseng uses an **economic mechanism** that incentivizes guests to reveal how much memory is worth to them, so it can optimize the social welfare.

Each guest bids with a type (a valuation of the good - how much it is worth, subjectively)

The auctioneer finds the allocation that maximizes the social welfare.

The auctioneer charges guests according to the exclusion-compensation principle: the difference between the social welfare when they exist to the social welfare when they do not exist.

Prices are NOT uniform. Prices may drop to a minimal price (possibly zero) if there are enough resources.

A VCG auction is truthful: guests bid their real types, no matter what other guests do.
Example: Second Price Auction
Example: Second Price Auction
Example: Second Price Auction
How much should one bid in a second price auction?

The second price auction is truthful.

- Each guest bids with a required quantity and a unit-price (not with their full types).
- The highest unit-price bidders are allocated with resources at desired quantities.
- Guests are charged by the exclusion-compensation principle.
- Guests hear each other’s bids, change their requested quantities and converge to an equilibrium.
Challenges in Designing a Memory Allocation Mechanism for the Cloud

- Maintaining privacy. (spoiler: our Memory Progressive Second Price (MPSP) protocol is based on secret bids.)
- Keeping memory from changing hands too often.
- Supporting real-world valuation and performance profiles.
A Round in The MPSP Auction Protocol

TIME (s)

12

Make memory changes

HOST

Compute allocation

4

Announce results

Secret Bids

3

Announce free memory

GUESTS

Prepare for memory changes if needed

Agmon Ben-Yehuda, Posener, Ben-Yehuda, Schuster, Mu’alem
PSP is not Ideal for Memory Allocation: Cache Warmup

Memory is only beneficial if you use it long enough (e.g. allowing cache warmup)

- In case of a tie between guests, none of the PSP guests wins the good.
- In the MPSP auction, ties are broken in favor of the guest currently holding the memory, as well as by a random order, such that memory does not go unused if it is needed.
PSP is not Ideal for Memory Allocation: Concavity

- PSP is for concave monotonically rising valuation functions, and for a divisible good only.
- Clients may sometimes value memory thus, in smart applications:

![Graph of Performance vs Allocation](https://github.com/ladypine/memcached)

**Figure**: Elastic-memory memcached:  
https://github.com/ladypine/memcached
PSP is not Ideal for Memory Allocation: Concavity

- PSP is for concave monotonically rising valuation functions, and for a divisible good only.
- But legacy applications tend to have a step function performance graph, which is not concave:

![Performance graph with concave and step function behaviors](image)

**Figure:** Off-the-Shelf memcached

---

**Agmon Ben-Yehuda, Posener, Ben-Yehuda, Schuster, Mu’alem, Ginseng 16/30**
PSP is not Ideal for Memory Allocation: Non-Smooth Online Measurements

- PSP is for concave monotonically rising valuation functions, and for a divisible good only.
- Sometimes, especially when performance is measured on-line, the performance is not even monotonically rising.

![Graph showing performance vs memory for memcached](image)

**Figure:** Elastic Memory memcached, in a non-optimal environment
MPSP supports non-concave, non-monotonically rising functions

The MPSP auction supports memory ranges:
- A bid is composed of a single unit-price and multiple memory ranges.

(a) Single range

(b) Multiple range
The MPSP Allocation Choice

The PSP allocation prefers higher unit prices, assuming there are no forbidden ranges:

Unit Price

Available Memory

PSP
Illegal allocation

Indivisible

Memory

Agmon Ben-Yehuda, Posener, Ben-Yehuda, Schuster, Mu’alem

Ginseng 19/30
Free disposal of auction results supports forbidden ranged, but is inefficient:

![Graph showing free disposal with limited memory usage.]

- **Unit Price**: 10
- **Available Memory**: SW = 10
- **Indivisible**
- **Memory**
The MPSP allocation finds the allocation with the highest social welfare under the forbidden ranges constraints:

MPSP

\[ \text{SW} = 10 + 3 = 13 \]
Guests in the MPSP are close to being truthful

Bidding the true valuation of a memory quantity is the best course of action when:

- The guest asks for a specific quantity (not a range), or
- The valuation function is concave monotonically rising, or
- The system is at a steady state.

(c) Single range  
(d) Multiple range

Agmon Ben-Yehuda, Posener, Ben-Yehuda, Schuster, Mu’alem  
Ginseng 22/30
MPSP maximizes the *Social Welfare* given the bids, even for non-concave non-monotonically rising valuations.

Guests are almost always *p-truthful*.

Guests can learn on-line and bid for a quantity that improves their profit, thus taking into effect the exclusion compensation payments (more in the paper...).

As a result, Ginseng can optimize the social welfare.
Experimental Evaluation: Non-Concave Valuation Functions

Figure: Valuation functions for different loads

(e) MemoryConsumer (square of performance)

(f) Dynamic Memcached (partially linear)
Experimental Evaluation: Comparison to Other Methods on a Dedicated Dynamic-Memory Benchmark

Figure: MemoryConsumer, valuation is square of performance

×6.2 improvement!

Agmon Ben-Yehuda, Posener, Ben-Yehuda, Schuster, Mu’alem

Ginseng 25/30
Experimental Evaluation: Comparison to Other Methods on a Dynamic-Memory Memcached

Figure: Memcached, first guest valuation is piecewise linear

×15.8 improvement!
Money (economic valuations) is the best way to give priorities, especially in the cloud.

Black-box approaches are limited and error-prone.

White-box is not feasible.

But Ginseng, a selfishness-based mechanism, does the trick!
Thank You for Listening!

Contact us at:
{lady pine, posener, muli, assaf, ahumu} at cs.technion.ac.il
Experimental Evaluation: Performance is comparable

Figure: Memcached, first guest valuation is piecewise linear
Offline profiling is good enough

Figure: Memcached