# Distributed Systems Meet Economics: Pricing in the Cloud



Presented by Sachin Shubham

# Introduction

- Pricing as a bridge between provider and user.
- Pay as you go model
  - Cloud provider have pricing scheme for users based on system usage
- Pricing depends on two factors:
  - System design and optimization
  - Fairness and Competitive pricing
- Studied conducted  on by using two complementary approaches for evaluations:
  1. A black-box approach with Amazon EC2.
  2. Set up a cloud-computing testbed, called Spring

# Background on Pricing

- Factors impacting pricing:
  - Fairness:
    - Personal : Price meets user's personal expectation.
    - Social : Price is same for all users.
  - Competition : provider cannot set their prices in a way most favorable to them. Instead, they gain a competitive advantage through adopting new technology and lowering their cost.

# Pay-as-you-go Model :

- the pricing scheme becomes an important bridge between users and providers.
- Computing price based on virtual-machine hours.
- Several alternative pricing schemes have been proposed for better system behavior in the cloud. E.g. Gurmeet and Karl suggested dynamic pricing on resource consumption.

# Workloads

- Postmark:
  - As an I/O-intensive benchmark, representing the file transactions for various web-based applications.
  - For experiment: the total file size is around 5 GB (1000 files, 5000 KB each); the number of transactions is 1000
- PARSEC:
  - Benchmark suite for chip-multiprocessor
  - Composed of multithread programs
  - 9 applications and 3 kernels
    - BlackScholes: High performancing compting
    - Dedup: Storage
  - For experiment: 184 MB input data for Dedup and 10 million options for Blackscholes

# Contd…

- Hadoop:
  - Use Hadoop 0.20.0 for large-scale data processing.
  - WordCount and StreamSort from the GridMix benchmark
  - For Experiment : Input dataset is 16 GB

# Methodologies

- Methodology on Amazon EC2:
  - Charged according to the pricing scheme of Amazon.
  - Calculate user expenses as $Cost_{user} = Price \times t$

    t : the total running time of the task in hours,

    Price: the price per virtual machine hour
  - Exclude the costs on storage and on data transfer between the client and the cloud
- Methodology on the Spring System:
  - Physical data center and provides virtual machines to users.
  - Consists of two major modules, VMM (Virtual Machine Monitor) and an auditor
  - Provider Profit = Payment from user - Total provider cost

# Experimental Setup in Amazon EC2

- Two virtual-machine types provided by EC2: small and medium instances

| Instance Type | CPU (#virtual core) | RAM (GB) | Storage (GB) | Price ($/h) |
|---|---|---|---|---|
| Small | 1 | 1.7 | 160 | 0.095 |
| Medium | 2 | 1.7 | 350 | 0.19 |

Table 1: The configurations and prices on different VM types on Amazon (Linux, California, America, Jan-2010)

# Experimental Setup in Spring

- Use VirtualBox to implement a virtual machine in Spring.
- The host OS is Windows Server 2003; the guest OS is Fedora 10.
- An eight-core machine to evaluate the single-machine benchmarks, and a cluster consisting of 32 four-core machines to evaluate Hadoop.

| | Eight-core machine | Four-core machine |
|---|---|---|
| CPU | Intel Xeon E5335 8-way 2.00GHz | Intel Xeon X3360 Quad 2.83GHz |
| RAM (GB) | 32 | 8 |
| Disk | RAID 5 (SCSI disks) | RAID 0 (SATA disks) |
| Network | 1 Gigabit | 1 Gigabit |
| Power model | $P_{idle} = 299$, $c_0 = 0.46$, $c_1 = 0.16$ | $P_{idle} = 250$, $c_0 = 0.4$, $c_1 = 0.14$ |

Table 2: Hardware configuration of machines in Spring

# Contd…

- Power meter to measure the actual power consumption of a server
- Estimating the total dollar cost based on Hamilton's parameter settings
- An Intel 80 GB X25-M SSD (Solid State Drives) to replace a SATA hard drive result in adjust the amortized cost in the machine with an SSD to $0.09 per hour.
- system throughput=numbers of tasks finished per hour + user costs + provider profits
- the efficiency of a provider's investment:

$$ROI = Profit / Cost_{provider} \times 100\%.$$

# User Optimization on EC2

- Focus on application-level optimizations for a fixed instance type, choosing the suitable instance type is important for both performance and cost.
- Results indicates that choosing the suitable instance type for cost may not result in the best performance, and vice versa.

| | On a small instance | | On a medium instance | |
|---|---|---|---|---|
| | Elapsed time (sec) | Cost ($) | Elapsed time (sec) | Cost ($) |
| Postmark | 204.0 | 0.0054 | 203.2 | 0.0106 |
| Dedup | 45 | 0.0012 | 14 | 0.0008 |
| BlackScholes | 934 | 0.0246 | 215 | 0.0113 |

Table 3: Elapsed time and costs of single-machine benchmarks on small and medium instances on EC2

# Provider Optimization on Spring

- Focus on VM consolidation optimization by tune the number of concurrent VMs running on the same physical machine.
- Smaller cost on small and medium VM types

| #VM per physical machine | One VM | Two VMs | Four VMs |
|---|---|---|---|
| Average elapsed time (sec) | 127 | 125.5 | 425 |
| Average cost per task ($) | 0.004 | 0.004 | 0.012 |
| Total cost of users ($) | 0.014 | 0.014 | 0.047 |
| $P_{raw}$ (kWh) | 0.046 | 0.024 | 0.038 |
| $Cost_{provider}$ ($) | 0.024 | 0.012 | 0.020 |
| *Profit* ($) | -0.009 | 0.002 | 0.028 |
| ROI (%) | -40.0% | 17.2% | 142.0% |
| Throughput (tasks/h) | 28.3 | 56.4 | 33.9 |

Table 4: Effects of virtual-machine consolidation in Spring (every four Postmark, small VM type)

| #VM per physical machine | One VM | Two VMs | Four VMs |
|---|---|---|---|
| Elapsed time (sec) | 231.2 | 239.5 | 334.5 |
| Average cost per task ($) | 0.013 | 0.013 | 0.019 |
| Total cost of users ($) | 0.051 | 0.053 | 0.074 |
| $P_{raw}$ (kWh) | 0.080 | 0.043 | 0.032 |
| $Cost_{provider}$ ($) | 0.042 | 0.022 | 0.016 |
| *Profit* ($) | 0.010 | 0.031 | 0.058 |
| ROI (%) | 22.8% | 140.8% | 365.2% |
| Throughput (tasks/h) | 15.6 | 30.1 | 40.0 |

Table 5: Effects of virtual-machine consolidation in Spring (every four BlackScholes, medium VM type)

# Observations

- Consolidation greatly reduces power consumption by 150% and 21% on $P_{raw}$ for BlackScholes and Postmark respectively.
- As decrease in power cost and increase in user cost resulted in provider's profit increases significantly(ROI increase 180% to 340%)
- As more tasks are consolidated to the same physical machine, the throughput reaches a peak at consolidating two VMs with Postmark, and then degrades.
- The system throughput can degrade up to over 64% compared with the peak

# Pricing Fairness

- Personal Fairness:
  - Users mostly need clarification on how they are charged and complaints on charging too high for certain scenarios.
  - Indicate personal unfairness in the pricing scheme.
- Social Fairness:
  - As more VMs are consolidated onto the same physical machine, users pay more money for the same task
  - The cost variations on both Amazon and Spring indicate social unfairness of the current pricing scheme

# Conclusion

- By embracing a pricing scheme that connects providers with users, cloud computing has managed to bridge between distributed systems and economics.
- The experiment conducted on Amazon EC2 and spring shows that the cost variation on both result in social as well as personal unfairness of the current pricing scheme.
- Providers need to tune its pricing scheme to balance between their profit and user.


Q & A


THANK YOU