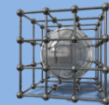


Issues about Automatic Scaling of Multi-tier architecture in Cloud

王文婷

2011-10-13





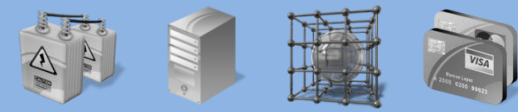
➤ Motivation

➤ Open Issues & Related Work

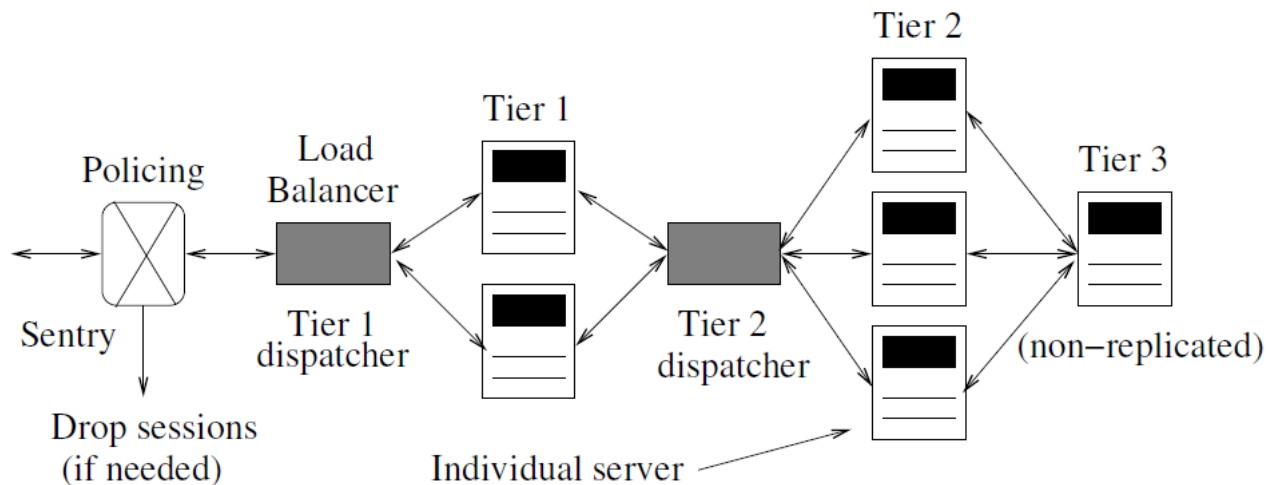
➤ Industrial Example

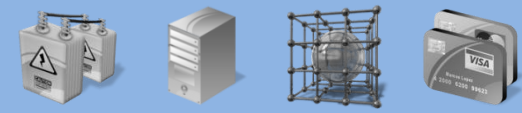
➤ Summary





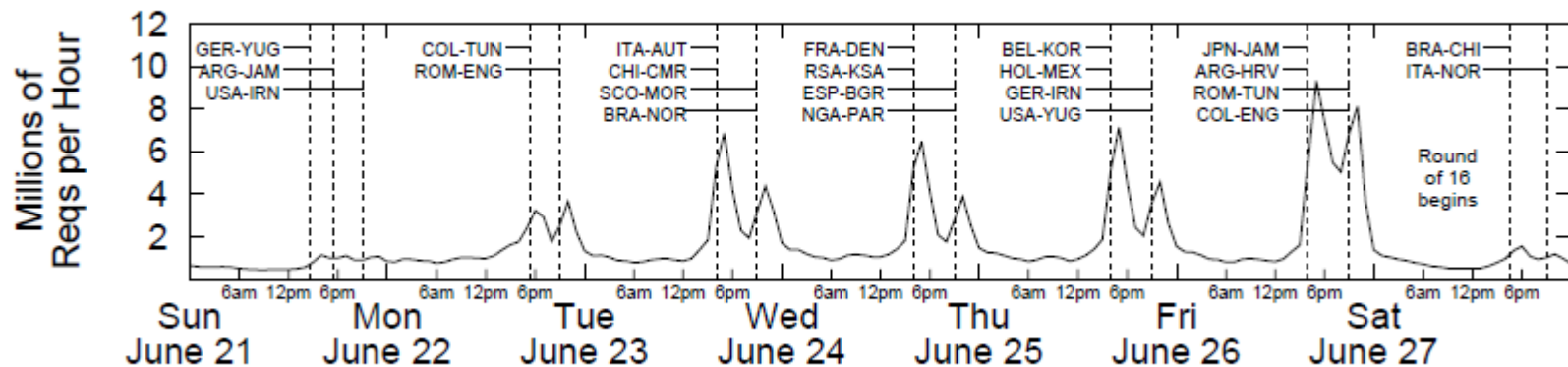
- Modern Internet applications
 - online retail sales, online auctions, wikis
- multiple tiers
 - A multi-tier architecture provides a flexible, modular approach for designing such applications.

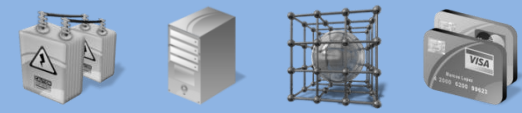




- Visit changes extend/reduce the number of server
 - Trend change
 - Seasonal change
 - Noise

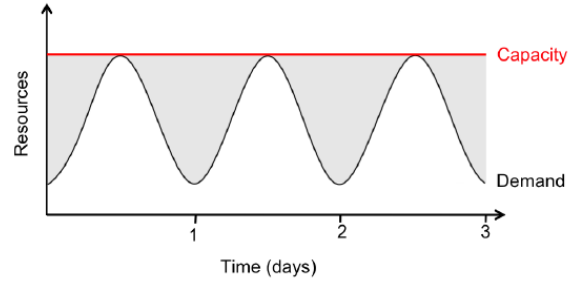
● 360buy.com



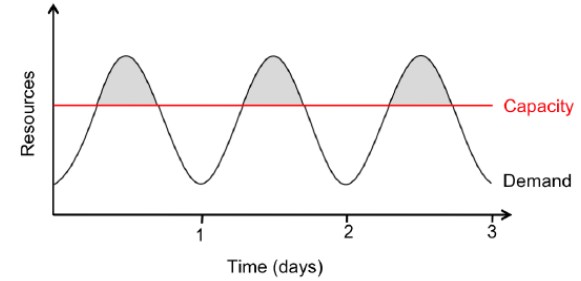


➤ Cloud

– Elastic



(a) Provisioning for peak load



(b) Underprovisioning 1

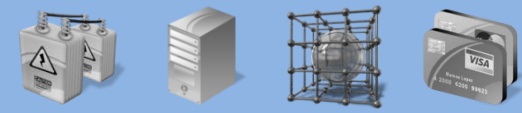
– Flexible

- multiple instance types

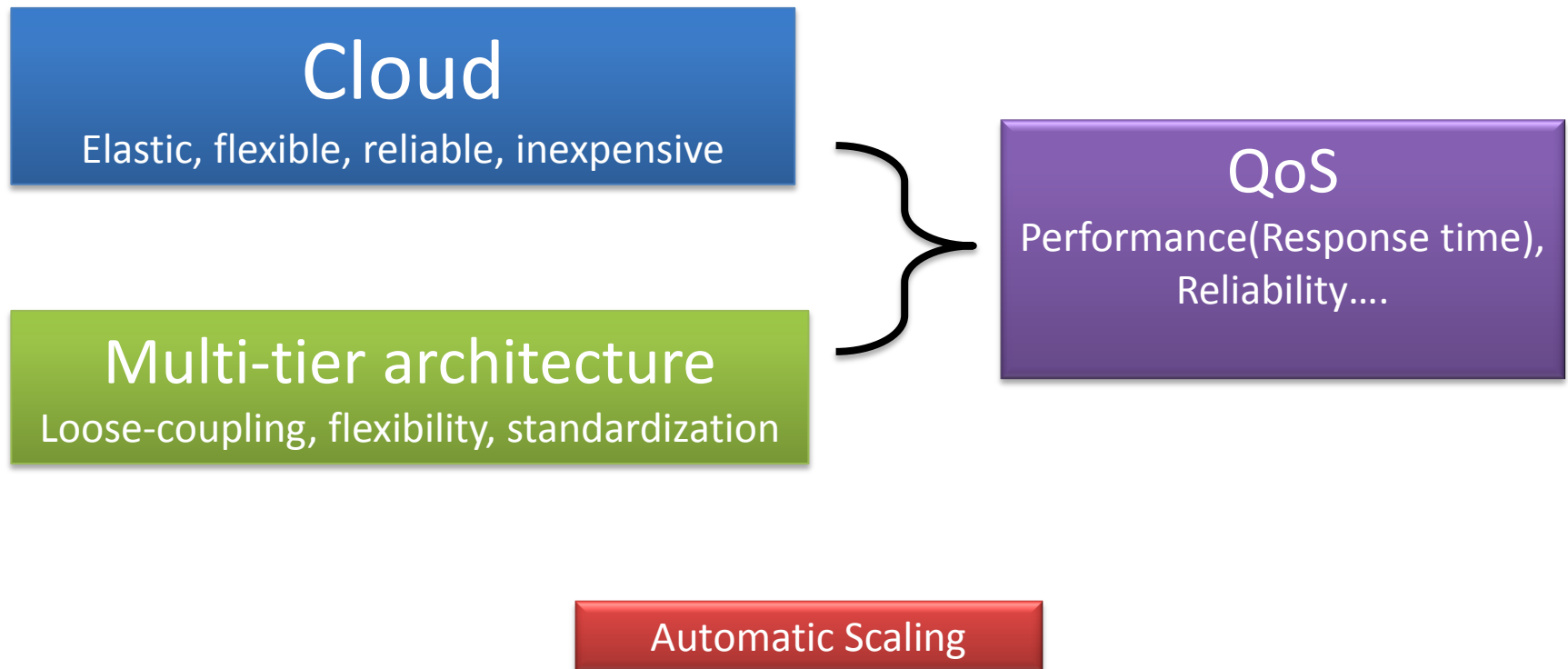
– Reliable

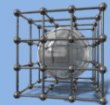
– Inexpensive



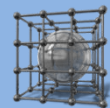


➤ Scale up/down





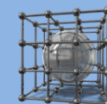
- Motivation
- Open Issues & Related Worked
- Industrial Example
- Summary



➤ Scale up or down

- **Who** should decide to scale up/down?
- **When** to scale up/down?
- **Which tier** should be scaled up/down?
- **How many** VMs should be added or reduced?
- What is the **policy** of scaling?
- **How** to add/ reduced? Resize or quantity change?
- **Which type** of VM should be added/reduced?
- **Where** the new VM should be placed? Or which old VMs should be terminated?



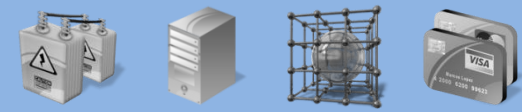


➤ Response time

- An Analytical Model for Multitier Internet Services and Its Applications, Bhuvan Urgaonkar, Giovanni Pacifici, Prashant Shenoy, Mike Spreitzer, and Asser Tantawi
- Chapter 2, Chapter 3, Dynamic Resource Management In Internet Hosting Platforms, Bhuvan Urgaonkar
 - Which tier should be scaled up/down?
 - How many VMs should be added or reduced?
 - When to scale up/down? (predict)



When, which tier & how many



上海交通大学 软件学院 高可靠实验室

➤ MVA algorithm

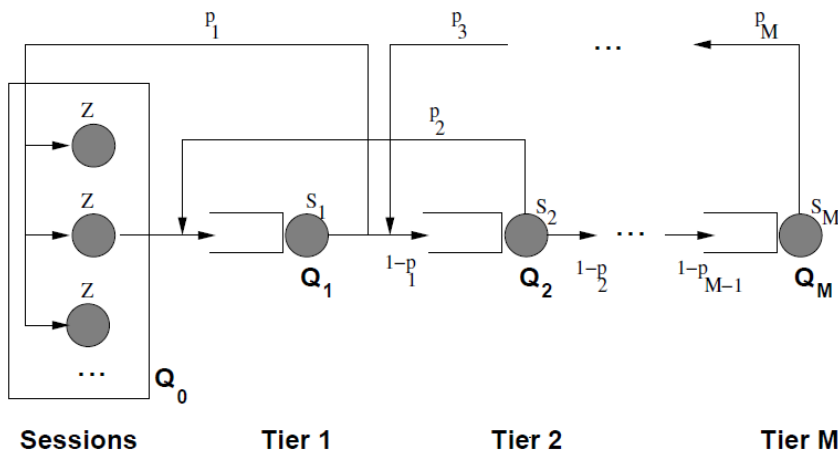


Figure 2.3. Modeling a multi-tier application using a network of queues.

input : $N, \bar{S}_m, V_m, 1 \leq m \leq M; \bar{Z}$
output : \bar{R}_m (avg. delay at Q_m), \bar{R} (avg. resp. time)

initialization:

$\bar{R}_0 = \bar{D}_0 = \bar{Z}; \bar{L}_0 = 0;$

for $m = 1$ **to** M **do**

$\bar{L}_m = 0;$

$\bar{D}_m = V_m \cdot \bar{S}_m$ /* service demand */;

end

/* introduce N customers, one by one */

for $n = 1$ **to** N **do**

for $m = 1$ **to** M **do**

$\bar{R}_m = \bar{D}_m \cdot (1 + \bar{L}_m)$ /* average delay */;

end

$\tau = \left(\frac{n}{\bar{R}_0 + \sum_{m=1}^M \bar{R}_m} \right)$ /* throughput */;

for $m = 1$ **to** M **do**

$\bar{L}_m = \tau \cdot \bar{R}_m$ /* Little's law */;

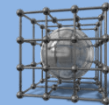
end

$\bar{L}_0 = \tau \cdot \bar{R}_0;$

end

$\bar{R} = \sum_{m=1}^M \bar{R}_m$ /* response time */;

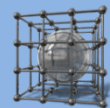




➤ Model Enhancements

- Replication and Load Imbalance at Tiers
- Handling Concurrency Limits at Tiers
- Handling Multiple Session Classes





➤ Replication and Load Imbalance at Tiers

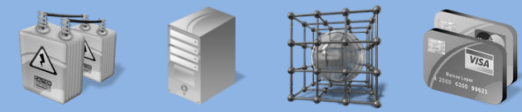
Let λ_i^j denote the number of requests forwarded to the j^{th} most loaded replica of tier T_i

– Imbalance factor

$$\beta_i^j = \left(\frac{\lambda_i^j}{\lambda_i} \right).$$



When, which tier & how many



上海交通大学 软件学院 高可靠实验室

➤ Handling Concurrency Limits at Tiers

we add a transition into an infinite server queuing subsystem $Q_{i,j}^{drop}$.

Let $V_{i,j}^{drop}$ denote the visit ratio for $Q_{i,j}^{drop}$ as shown in Figure 2.5.

$Q_{i,j}^{drop}$ has a mean service time of S_i^{drop} ;

Requests that are dropped at $Q_{i,j}$ experience some delay in the subsystem $Q_{i,j}^{drop}$ before returning to Q_0

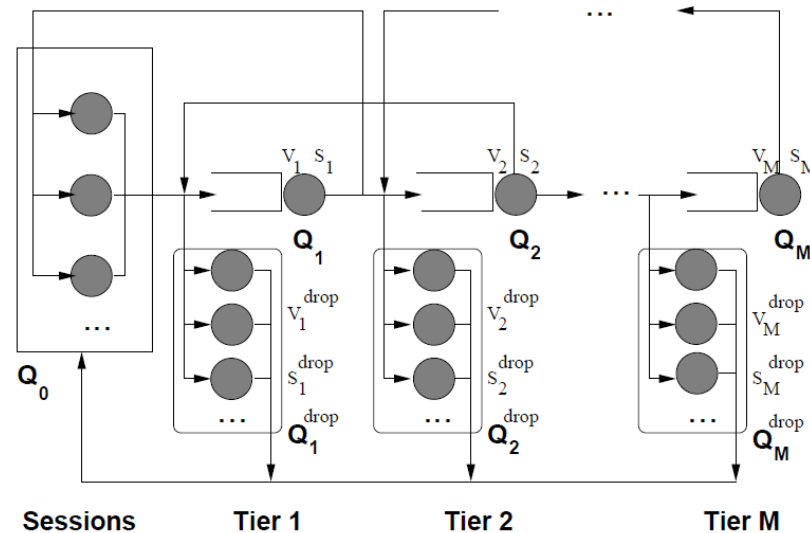
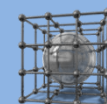


Figure 2.5. Multi-tier application model enhanced to handle concurrency limits. Since each tier has only one replica, we use only one subscript in our notation.



➤ Handling Multiple Session Classes

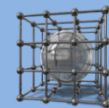
- The estimation of the drop probabilities, however, needs to be done on a per-class basis.

Step 1: *Estimate throughput of the queuing network if there were no concurrency limits:* Solve the queuing network using the multi-class MVA algorithm with $V_{c,i,j}^{drop} = 0, 1 \leq c \leq C$ (i.e., assuming that the queues have no concurrency limits). Let $\lambda = \sum_{c=1}^C \lambda_c$ denote the throughput computed by the MVA algorithm in this step.

Step 2: *Estimate $V_{c,i,j}^{drop}$:* Treat $Q_{i,j}$ as an open, finite-buffer M/M/1/ K_i queue with arrival rate $\lambda V_{i,j}$ (using the λ computed in Step 1). Let $p_{i,j}^{drop}$ denote the probability of buffer overflow in this M/M/1/ K_i queue [64]. Then $V_{c,i,j}^{drop}$ is estimated as: $V_{c,i,j}^{drop} = p_{i,j}^{drop} \cdot V_{c,i,j} \cdot \frac{\lambda_c}{\lambda}$. Also, $V_{c,i,j}$ is updated as: $V_{c,i,j} = (1 - p_{i,j}^{drop}) \cdot V_{c,i,j} \cdot \frac{\lambda_c}{\lambda}$.



When, which tier & how many



上海交通大学 软件学院 高可靠实验室

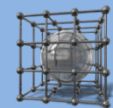
```

input      :  $N_c$  (num. sessions of class  $c$ ),  $\bar{S}_{c,m}, V_{c,m}, 1 \leq c \leq C, 1 \leq m \leq M; \bar{Z}$ 
output    :  $\bar{R}_{c,m}$  (avg. delays at  $Q_m$ ),  $\bar{R}_c$  (avg. resp. time for class  $c$ ),  $1 \leq c \leq C$ 

initialization:

for  $c = 1$  to  $C$  do
     $\bar{R}_{c,0} = \bar{D}_{c,0} = \bar{Z};$ 
end
 $\bar{L}_0(0) = 0;$ 
for  $m = 1$  to  $M$  do
     $\bar{L}_m(0) = 0;$ 
    for  $c = 1$  to  $C$  do
         $\bar{D}_{c,m} = V_{c,m} \cdot \bar{S}_{c,m}$  /* service demand */;
    end
end
/* introduce N customers, one by one */
for  $n = 1$  to  $N$  do
    for each feasible popl.  $\underline{n} = (n_1, \dots, n_C)$  s. t.  $n = \sum_{c=1}^C n_c, n_c \geq 0$ 
        for  $c = 1$  to  $C$  do
            for  $m = 1$  to  $M$  do
                 $\bar{R}_{c,m} = \bar{D}_{c,m} \cdot (1 + \bar{L}_m(\underline{n} - 1_c))$  /* average delay */;
            end
        end
        for  $c = 1$  to  $C$  do
             $\tau_c = \left( \frac{n_c}{\bar{R}_{c,0} + \sum_{m=1}^M \bar{R}_{c,m}} \right)$  /* throughput */;
            for  $m = 1$  to  $M$  do
                 $\bar{L}_m(\underline{n}) = \sum_{c=1}^C \tau_c \cdot \bar{R}_{c,m}$  /* Little's law */;
            end
        end
         $\bar{L}_0(\underline{n}) = \sum_{c=1}^C \tau_c \cdot \bar{R}_{c,0};$ 
    end
for  $c = 1$  to  $C$  do
        for  $m = 1$  to  $M$  do
             $\bar{R}_c = \sum_{m=1}^M \bar{R}_{c,m}$  /* response time */;
        end
    end

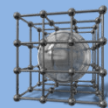
```



➤ How much to provision

- In case this is worse than the target, we use **the MVA algorithm** to determine, for each replicable tier, the response time resulting from **the addition of one more server to it**. We add a server to the tier that results in the greatest improvement in response time.
- We **repeat** this until we have an assignment for which the predicted response time is below the target





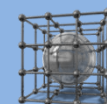
➤ DYNAMIC CAPACITY PROVISIONING

– When to Provision

- Predictive provisioning----estimate the workload for the next few hours and provision for it accordingly.
- Reactive provisioning ----correct errors in the long-term predictions or to react to unanticipated ash crowds.

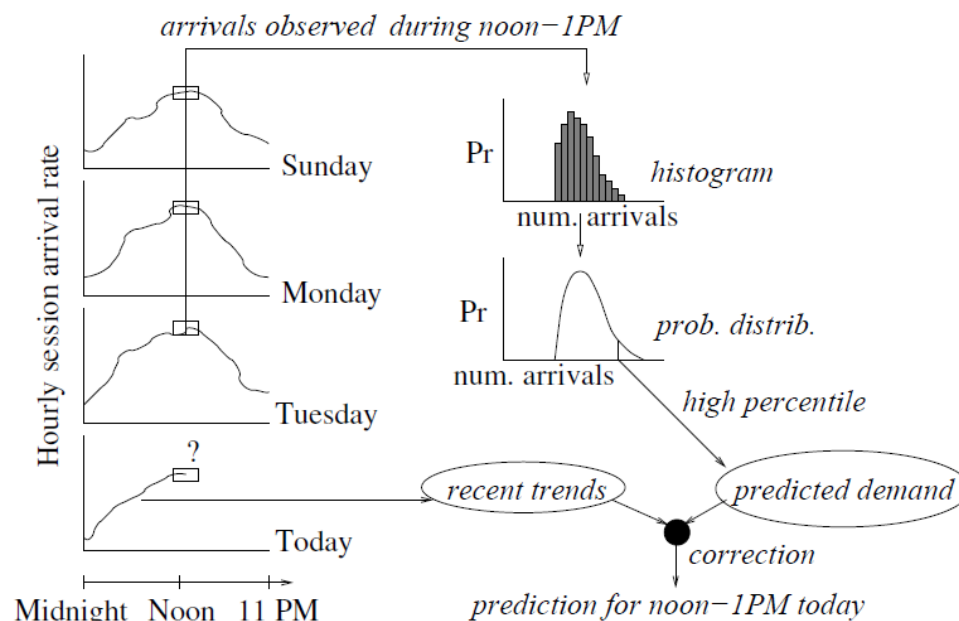


When, which tier & how many



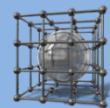
上海交通大学 软件学院 高可靠实验室

➤ Predictive Provisioning



$$\lambda_{pred}(t) = \lambda_{pred}(t) + \sum_{i=t-h}^{t-1} \frac{\max(0, \lambda_{obs}(i) - \lambda_{pred}(i))}{h},$$



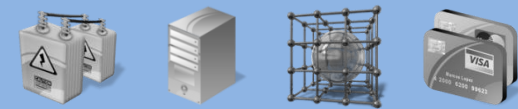


➤ Reactive Provisioning

- the workload on a given day deviates from its behavior on previous days
- sudden load spikes
- invoked once every few minutes if $\frac{\lambda_{obs}(t)}{\lambda_{pred}(t)} > \tau_1$ or drop rate $> \tau_2$



When, which tier & how many



上海交通大学 软件学院 高可靠实验室

➤ Predictive and Reactive Provisioning

Result:

We need reactive mechanisms to deal with large flash crowds.

However, reactive provisioning alone may not be effective, since its actions lag the workload.

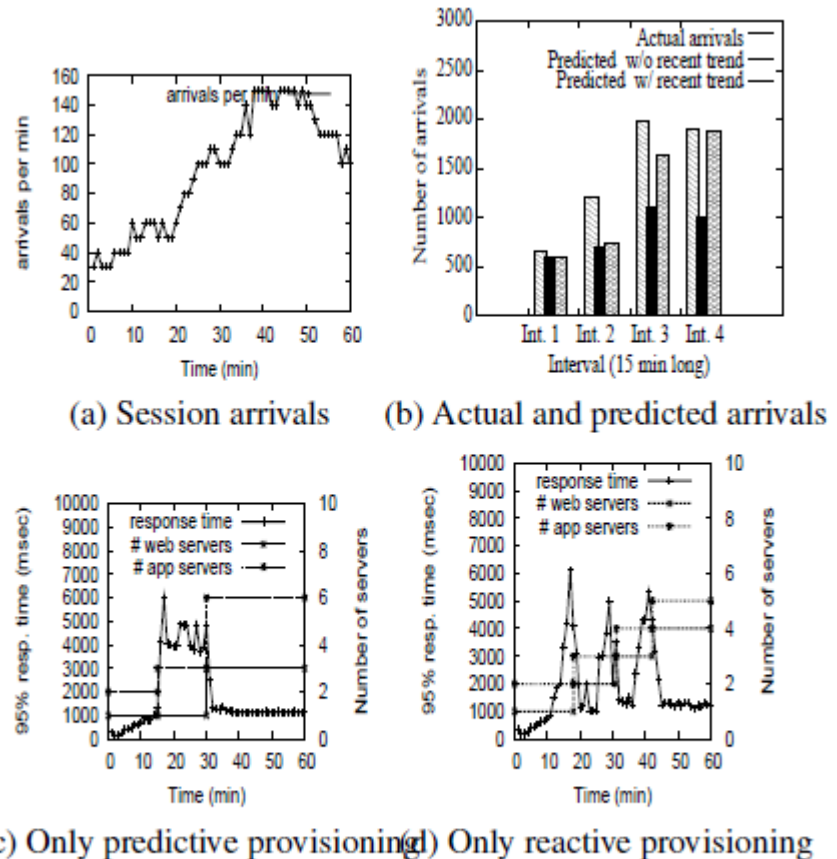
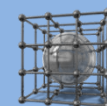


Figure 3.9. Provisioning on day 7—moderate overload



What is the policy of scaling?



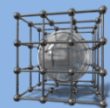
上海交通大学 软件学院 高可靠实验室

➤ Profit-driven

- Characterizing Web Application Performance for Maximizing Service Porvider' s Profits in Clouds, Xi Chen, Haopeng Chen, Qing Zheng, Wenting Wang, Guodong Liu
 - **Policy** of scaling



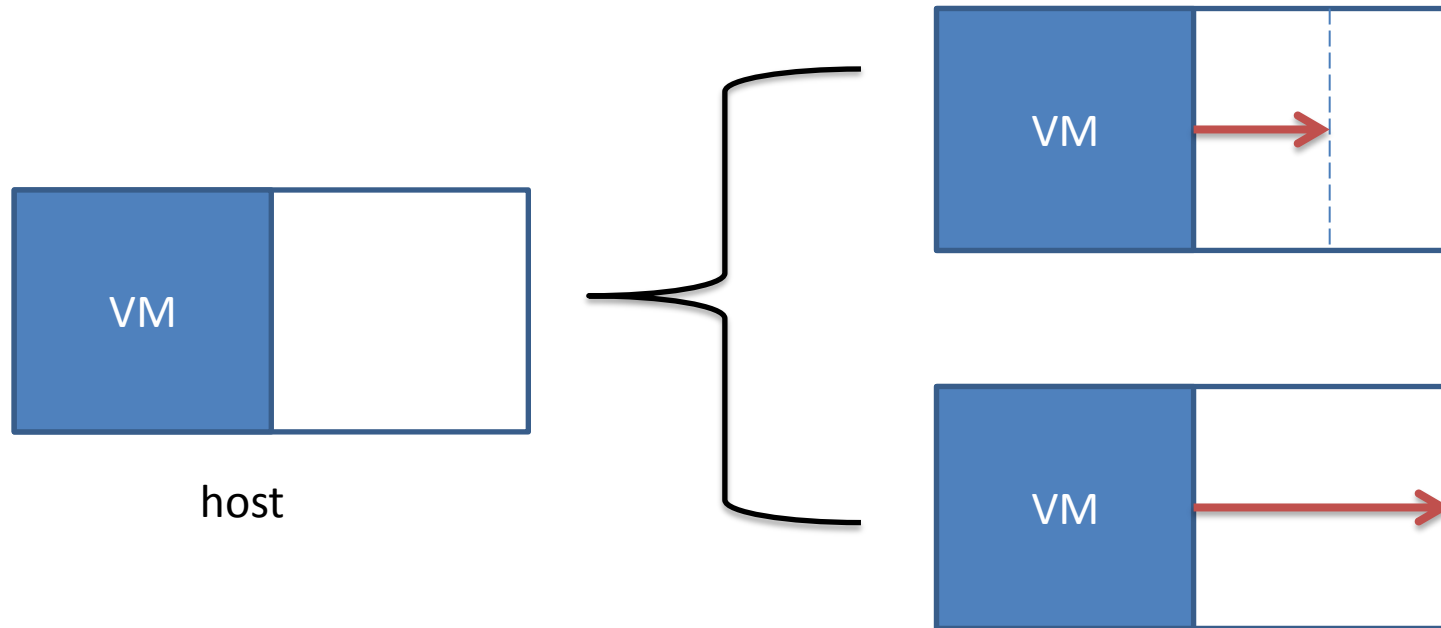
Resize or add/reduce number?



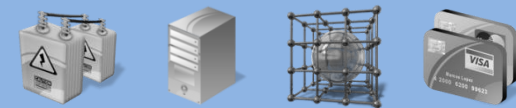
上海交通大学 软件学院 高可靠实验室

➤ Resize or quantity change?

- Performance
- VM fragment



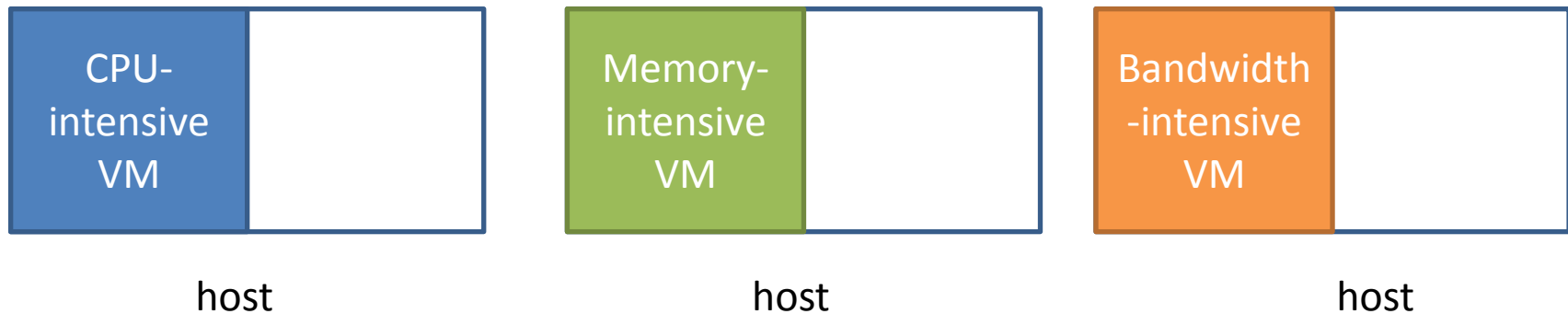
Where to allocate?



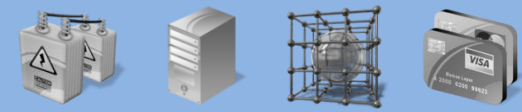
上海交通大学 软件学院 高可靠实验室

➤ Which host?

- Enhance Utility



Where to allocate?



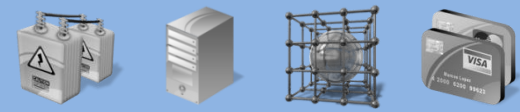
上海交通大学 软件学院 高可靠实验室

➤ Algorithm

- Goal: utility & reliability
- Multiple dimensions: CPU, Memory, Bandwidth, storage……
- Knapsack problem
- Heuristic algorithm



Where to allocate?



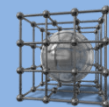
上海交通大学 软件学院 高可靠实验室

➤ Which availability zone and region?

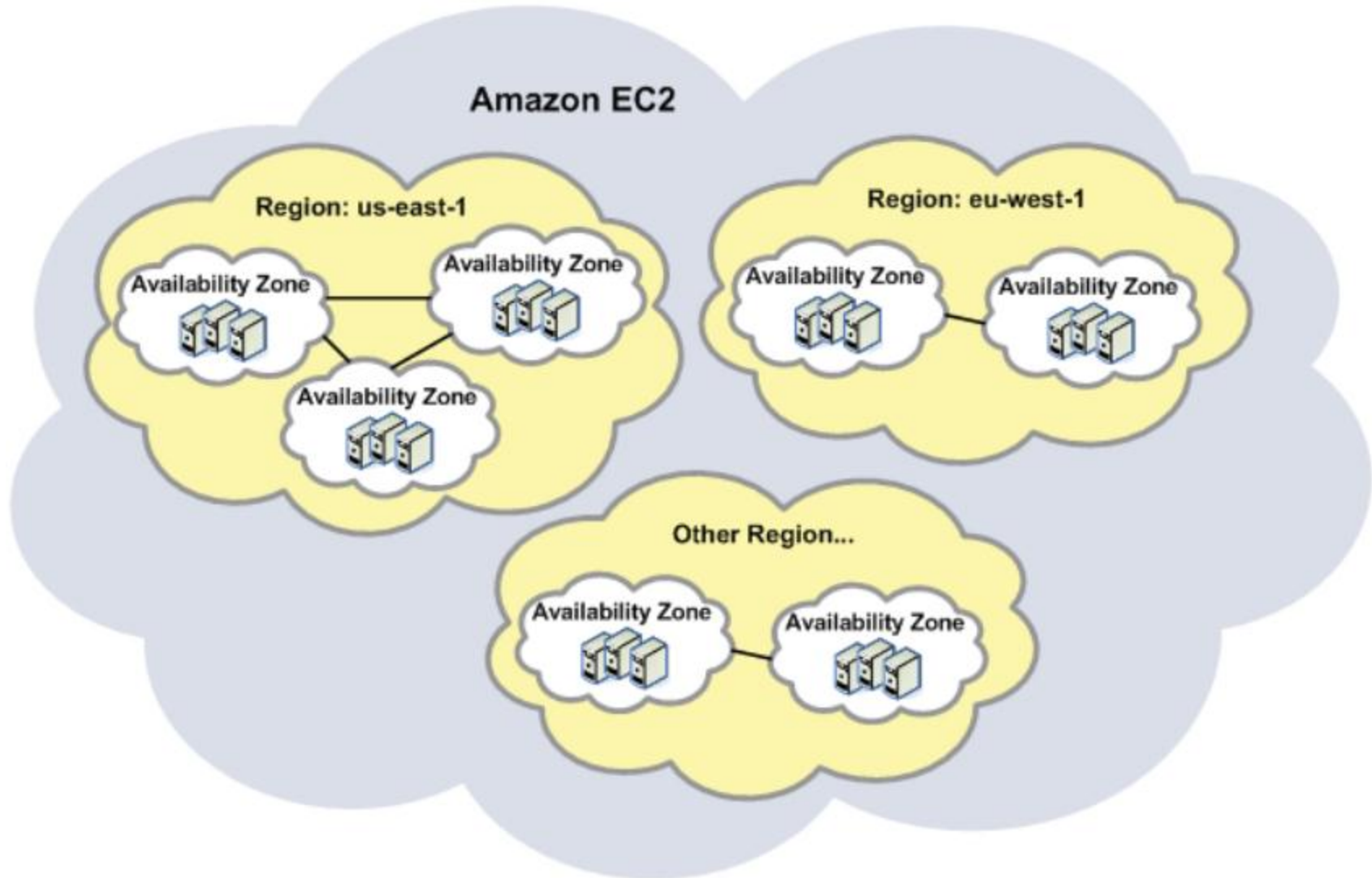
- Regions are dispersed and located in separate geographic areas
 - six regions: US East (Northern Virginia), US West (Northern California), EU (Ireland), Asia Pacific (Singapore), Asia Pacific (Tokyo), and [AWS GovCloud](#).
- Availability Zones are distinct locations **within** a Region that are engineered to be **isolated from failures** in other Availability Zones and provide **inexpensive, low latency network connectivity to other Availability Zones in the same Region**.
 - Availability Zones have independent networking, power, and cooling, and separation from risks such as flood and fire



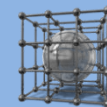
Where to allocate?



上海交通大学 软件学院 高可靠实验室



Where to allocate?



上海交通大学 软件学院 高可靠实验室

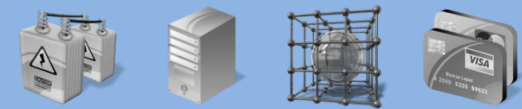
- By launching instances in separate Regions, you can design your application to **be closer to specific customers or to meet legal or other requirements.**

Region	Endpoint
US-East (Northern Virginia) Region	ec2.us-east-1.amazonaws.com
US-West (Northern California) Region	ec2.us-west-1.amazonaws.com
EU (Ireland) Region	ec2.eu-west-1.amazonaws.com
Asia Pacific (Singapore) Region	ec2.ap-southeast-1.amazonaws.com
Asia Pacific (Tokyo) Region	ec2.ap-northeast-1.amazonaws.com

- By launching instances in separate Availability Zones, you can **protect your applications from the failure** of a single location.



Where to allocate?



上海交通大学 软件学院 高可靠实验室

AWS Management Console

Navigation: Region: US

Request Instances Wizard

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL

Provide the details for your instance(s). You may also decide whether you want to launch "spot" instances.

Number of Instances: 1 **Availability Zone:** No Preference (dropdown menu open showing: us-east-1a, us-east-1b, us-east-1c, us-east-1d)

Instance Type: Small (m1.small, 1.7 GB)

☒ **Launch Instances**

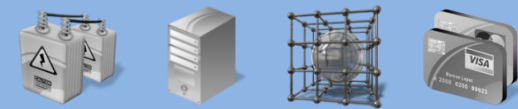
EC2 Instances let you pay for compute capacity by the hour with no long term commitment. They convert commonly large fixed costs into much smaller variable costs.

☐ Request Spot Instances

☐ Launch Instances Into Your Virtual Private Cloud



Where to allocate?



上海交通大学 软件学院 高可靠实验室

➤ Charge

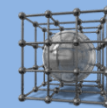
Internet Data Transfer

The pricing below is based on data transferred "in" and "out" of Amazon EC2.

Region: Asia Pacific (Tokyo)	
Pricing	
Data Transfer IN	
All data transfer in	\$0.000 per GB
Data Transfer OUT	
First 1 GB / month	\$0.000 per GB
Up to 10 TB / month	\$0.201 per GB
Next 40 TB / month	\$0.158 per GB
Next 100 TB / month	\$0.137 per GB
Next 350 TB / month	\$0.127 per GB
Next 524 TB / month	Contact Us
Next 4 PB / month	Contact Us
Greater than 5 PB / month	Contact Us



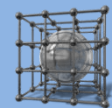
Where to allocate?



上海交通大学 软件学院 高可靠实验室

	different regions	different Availability Zones in the same Region	Same availability zone
Amazon EC2 -- Amazon S3	Internet transfer Charge both sides	no charge	no charge
between Amazon EC2	Internet transfer Charge both sides	Regional Data Transfer-\$0.01	no charge
between AWS services	Internet transfer Charge both sides	Regional Data Transfer-\$0.01	no charge

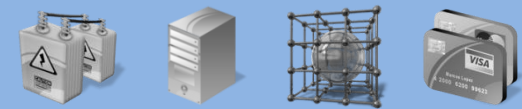




➤ Reliability in Distributed System

- 80s - 90s
- Software reliability, no environmental concerns(location)
- Focus on the topology, the reliability of the communication edges and file transfer
- Focus on Markov chain, execution graphs





- A study of service reliability and availability for distributed systems

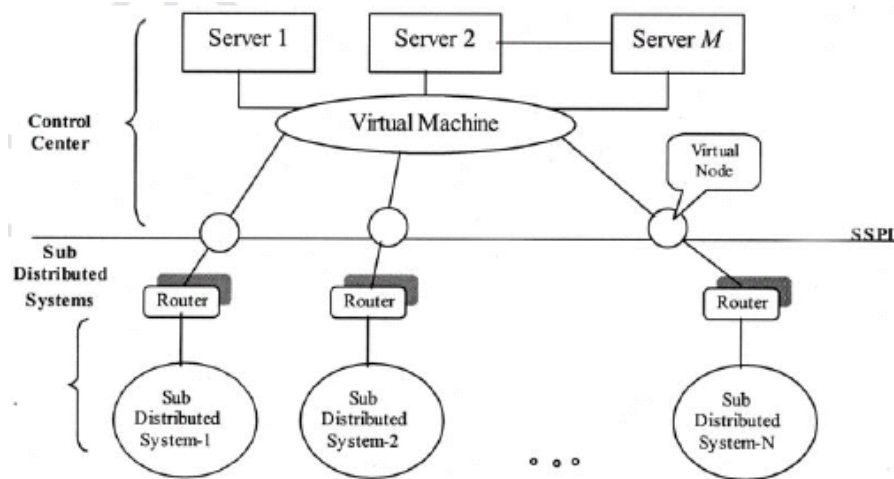


Fig. 1. Structure of the centralized heterogeneous distributed service system.

$$R_s(t_b) = \prod_{i=1}^N \text{DSR}_i \prod_{j=1}^J P_f(j) \prod_{k=1}^K P_{pr}(k).$$

GEAR algorithm presented by Kumar and Agrawal

$$P_f(j) = A(T_{bf}^j), \quad j = 1, 2, \dots, J.$$

$$P_{pr}(k) = \int_{T_{bp}^k}^{T_{bp}^k + T_{ex}^k} A(t) dt / T_{ex}^k, \quad k = 1, 2, \dots, K.$$



Step 1:

$$\beta_1 = 1, \beta_2 = 2, \beta_3 = 4, \beta_5 = \infty;$$

Step 2:

$$\Pr(Y_1) = p_1, \Pr(Y_2) = p_2 \cdot p_3 \cdot p_4,$$

$$\Pr(Y_3) = p_3 \cdot p_4, \{\Pr(Y_4) = p_4 \cdot p_5, \Pr(Y_5) = 0;$$

Step 3:

$$\Pr(R_0) = 0;$$

$$\Pr(U_1) = p_1, \Pr(R_1) = \Pr(R_2) = \Pr(R_3) = \Pr(U_1) = p_1;$$

$$i = 2: \Pr(U_3) \Pr(U_2) = \Pr(U_1) + [1 - \Pr(R_0)] \cdot q_1 \cdot \Pr(Y_2) \\ = p_1 + q_1 \cdot p_2 \cdot p_3 \cdot p_4$$

$$i = 3: \Pr(U_3) = \Pr(U_2) + [1 - \Pr(R_1)] \cdot q_2 \cdot \Pr(Y_3) \\ = p_1 + q_1 \cdot p_2 \cdot p_3 \cdot p_4 + q_1 \cdot q_2 \cdot p_3 \cdot p_4$$

$$\Pr(R_4) = \Pr(U_3) = p_1 + q_1 \cdot p_2 \cdot p_3 \cdot p_4 + q_1 \cdot q_2 \cdot p_3 \cdot p_4$$

$$i = 4: \Pr(U_4) = \Pr(U_3) + [1 - \Pr(R_2)] \cdot q_3 \cdot \Pr(Y_4) \\ = p_1 + q_1 \cdot p_2 \cdot p_3 \cdot p_4 + q_1 \cdot q_2 \cdot p_3 \cdot p_4 + q_1 \cdot q_3 \cdot p_4 \cdot p_5$$

$$\Pr(R_5) = \Pr(U_4) = p_1 + q_1 \cdot p_2 \cdot p_3 \cdot p_4 + q_1 \cdot q_2 \cdot p_3 \cdot p_4 + q_1 \cdot q_3 \cdot p_4 \cdot p_5$$

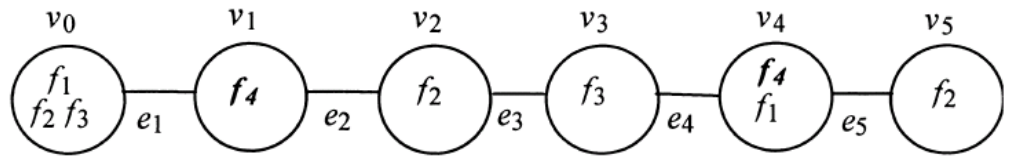
$$i = 5: \Pr(U_5) = \Pr(U_4) + [1 - \Pr(R_3)] \cdot q_4 \cdot \Pr(Y_5) \\ = \Pr(U_4) \quad // \text{ since } \Pr(Y_5) = 0 //$$

$$\text{DPR} \leftarrow \Pr(U_n);$$

Y_i event: all edges in I_i function

R_j event: there exists an operating event Y_i between edges e_1 and e_j

$$U_i \bigcup_{j=1}^i Y_j$$

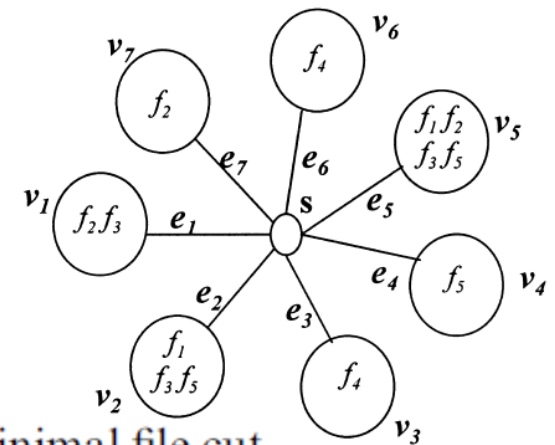


Program f_4 needs data files f_1, f_2 , and f_3 for its execution.

Fig. 4. A DCS with a linear structure.



Software Reliability



Step 1: // find all file cut sets //

Step 2: // set the values of α_i and β_i for $1 \leq i \leq m$ //

Step 3: // find all minimal file cut set //

Step 4: reorder the minimal file cut sets in Φ for two distinct minimal file cut sets C_i and C_j , $i < j$ if and only if $\alpha_i < \alpha_j$;

Step 5: // compute $\Pr[X(j+1, \beta_i)]$, for $2 \leq i \leq r$ and $\alpha_{i-1} \leq j \leq \alpha_i - 1$, by Eq. (6) //

Step 6: // Apply Theorem 1 and Eq. (7) to compute $\Pr(W_i)$ and $\Pr(F_j)$ //

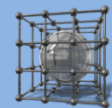
$$\Pr(F_k) = \begin{cases} \Pr(W_{i-1}) & \text{for } \beta_{i-1} \leq k \leq \beta_i - 1, \\ 0 & \text{for } k \leq \beta_1 - 1. \end{cases}$$

$H(i, j) \equiv \{e_{\pi(i)}, e_{\pi(i+1)}, \dots, e_{\pi(j)}\}; 1 \leq i \leq j \leq n$ (note that $C_i \equiv H(\alpha_i, \beta_i)$)

$X(i, j)$ event: all edges in $H(i, j)$ fail

$\Pi \equiv [\pi(1), \pi(2), \dots, \pi(n)]$ a permutation of numbers $\{1, 2, \dots, n\}$ such that if file $f_d \in A_{\pi(i)}$ and $f_d \in A_{\pi(j)}$, then $f_d \in A_{\pi(k)}$ for all k , $i < k < j$

C_d the minimal file cut set for file f_d if it consists of all edges (s, v_i) such that node v_i contains file f_d , i.e. $C_d = \{(s, v_i) \mid f_d \in A_i\}$.
(Without loss of generality, we reorder the minimal file cut sets, if necessary, by their minimal component, i.e. for two distinct minimal file cut sets C_i and C_j , $i < j$ if and only if $\min\{k \mid (s, v_{\pi(k)}) \in C_i\} < \min\{k \mid (s, v_{\pi(k)}) \in C_j\}$.)

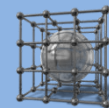


- DTMC
- Absorbing:
 - If at least one state has no outgoing transition
- Let $X_{i,j}$ represent the number of visits to state j starting from state i before the process is absorbed

$$P = \begin{bmatrix} Q & C \\ 0 & 1 \end{bmatrix} \longrightarrow P^k = \begin{bmatrix} Q^k & C' \\ 0 & 1 \end{bmatrix} \quad M = (I - Q)^{-1} = I + Q + Q^2 + \cdots = \sum_{k=0}^{\infty} Q^k$$

$$E[X_{i,j}] = m_{i,j}$$





state i . Define $\mathbf{M_D} = [md_{i,j}]$ such that

$$md_{i,j} = \begin{cases} m_{i,j} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{M_2} = [m_{i,j}^2],$$

$$\sigma^2 = \mathbf{M}(2\mathbf{M_D} - \mathbf{I}) - \mathbf{M_2}$$

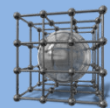
Hence

$$\text{Var}[X_{i,j}] = \sigma_{i,j}^2$$

$$R = \prod_i^n R_i^{X_{1,i}} \longrightarrow E[R] = E\left[\prod_i^n R_i^{X_{1,i}}\right] = \prod_i^n E[R_i^{X_{1,i}}] \longrightarrow E[R_i^{X_{1,i}}] = R_i^{E[X_{1,i}]} + \frac{1}{2}(R_i^{E[X_{1,i}]})^2 \text{Var}[X_{1,i}]$$

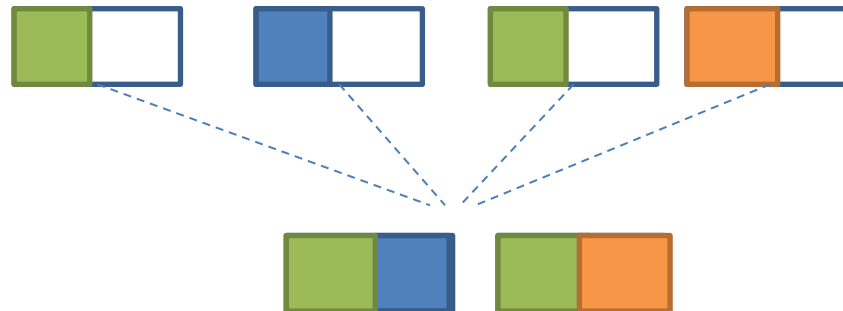
$$E[R] \approx \left[\prod_i^{n-1} R_i^{m_{1,i}} \right] R_n$$

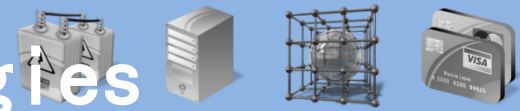




➤ VM governance/ migration

- When, how, cost
- Related to VM allocation & Reliability Issues





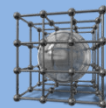
➤ Instance Type

- Towards Characterizing Cloud Backend Workloads: Insights from Google Compute Clusters, Asit K. Mishra, Joseph L. Hellerstein, The Pennsylvania State University Walfredo Cirne, Chita R. Das, Google Inc, 2010

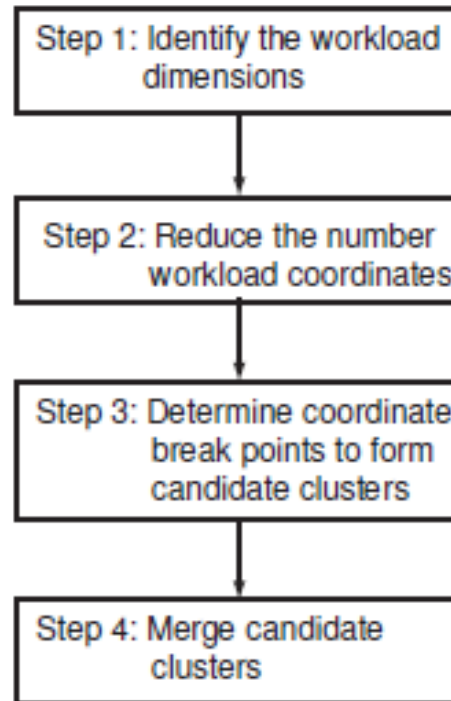
➤ Pricing Strategies

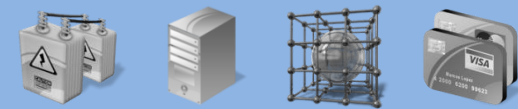
- Amazon EC2
- Microsoft Azure
- IBM Smart Cloud
- Google App Engine





➤ a methodology



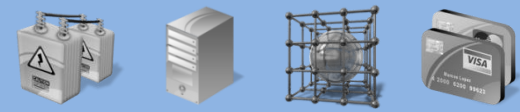


- Step 4 reduces the total number of task classes by merging
 - Merge adjacent classes if the CV of the merged task class is much less than 100%.
 - CV= the ratio of the standard deviation to the mean (often expressed as a percent)

Final Class	Duration(Hours)	CPU (cores)	Memory (GBs)
1: sss	Small	Small	Small
2: sm*	Small	Med	all
3: slm	Small	Large	Small+Med
4: sll	Small	Large	Large
5: lss	Large	Small	Small
6: lsl	Large	Small	Large
7: llm	Large	Med+Large	Small+Med
8: ll	Large	Med+Large	Large

Table 3. Final task classes (workloads)

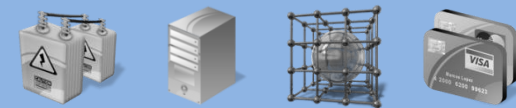




➤ Insights from Task Classification

- We see that task durations are bimodal, either somewhat less than 30 minutes or larger than 18 hours.
 - The first are **user-facing**. A second type of long-running tasks are compute intensive, such as processing web logs.
- we see that tasks with short duration dominate the task population.
 - sss tasks are short, highly parallel operations such as **index lookups and searches**.
 - sml tasks are short memory-intensive operations such as **map reduce workers computing an inverted index**.
 - slm tasks are short cpu-intensive operations such as **map reduce workers computing aggregations of log data**.
- observe that a small number of long running tasks consume most of the CPU and memory.
 - The first are computationally intensive, user-facing services such as work done by **a map reduce master** in processing web search results.
 - The second kind of long-running tasks relate to log-processing operations, such as **analysis of click through**.



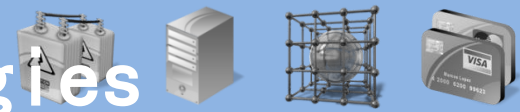


➤ Amazon Pricing System

Family	Description
Standard	Have memory-to-CPU ratios suitable for most general purpose applications
Micro	Provide a small amount of consistent CPU resources and allow you to burst CPU capacity when additional cycles are available. They are well suited for lower throughput applications and web sites that consume significant compute cycles periodically (for more information, see Micro Instances)
High-CPU	Have proportionally more CPU resources than memory (RAM) and are well suited for compute-intensive applications
High-Memory	Have proportionally more memory resources and are well suited for high throughput applications, such as database and memory caching applications
Cluster Compute	Have a very large amount of CPU coupled with increased networking performance, making them well suited for High Performance Compute (HPC) applications and other demanding network-bound applications (for more information, see Cluster Instance Concepts)
Cluster GPU	Provide general-purpose graphics processing units (GPUs), with proportionally high CPU and increased network performance for applications that benefit from highly parallelized processing. They're well suited for HPC applications as well as rendering and media processing applications (for more information, see Cluster Instance Concepts)



Instance type & pricing Strategies



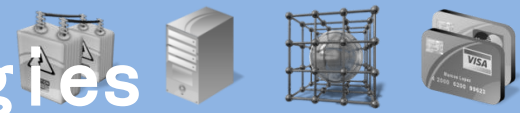
上海交通大学 软件学院 高可靠实验室

Available Instance Types

When you launch an instance, you specify the *instance type* (the value in the *Name* column in the following table). We launch an m1.small if you don't specify a particular instance type.

Type	CPU	Memory	Local Storage	Region: Asia Pacific (Tokyo)	
Small	1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)	1.7 GB	160 GB instance storage (150 GB plus 10 GB root partition)		Linux/UNIX Usage
Large	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	7.5 GB	850 GB instance storage (2 x 420 GB plus 10 GB root partition)		Standard On-Demand Instances
Extra Large	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)	15 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)		Small (Default) \$0.10 per hour
Micro	Up to 2 EC2 Compute Units (for short periodic bursts)	613 MB	None (use Amazon EBS volumes for storage)		Large \$0.40 per hour
High-CPU Medium	5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each)	1.7 GB	350 GB instance storage (340 GB plus 10 GB root partition)		Extra Large \$0.80 per hour
High-CPU Extra Large	20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)	7 GB	1690 GB instance storage (4 x 420 GB plus 10 GB root partition)		Micro On-Demand Instances
High-Memory Extra Large	6.5 EC2 Compute Units (2 virtual cores with 3.25 EC2 Compute Units each)	17.1 GB	420 GB instance storage (1 x 420 GB)		Micro \$0.027 per hour
High-Memory Double Extra Large	13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each)	34.2 GB	850 GB instance storage (1 x 840 GB plus 10 GB root partition)		Hi-Memory On-Demand Instances
High-Memory Quadruple Extra Large	26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)	68.4 GB	1690 GB instance storage (2 x 840 GB plus 10 GB root partition)		Extra Large \$0.60 per hour
Cluster Compute	33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)	23 GB	1690 GB instance 64-bit storage (2 x 840 GB plus 10 GB root partition)		Double Extra Large \$1.20 per hour
					Quadruple Extra Large \$2.39 per hour
					Hi-CPU On-Demand Instances
					Medium \$0.20 per hour
					Extra Large \$0.80 per hour
					Cluster Compute Instances
					Quadruple Extra Large N/A*
					Cluster GPU Instances
					Quadruple Extra Large N/A*
					Gbps Ethernet)

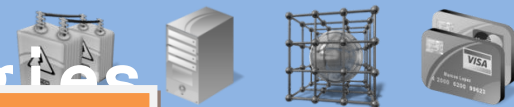




- Resource rent:
 - On-Demand: pay by hour
 - Reserved: pay by year
 - Spot: auction
- Storage
- Value-added Service
 - Data transfer
 - Auto scaling
 - Elastic IP address
 - Load balance



Instance type & pricing Strategies



上海交通大学 软件学院 高可靠实验室

➤ On-demand & Reserved

On-demand
租一年是
\$262.8
\$1051.2
\$2102.4

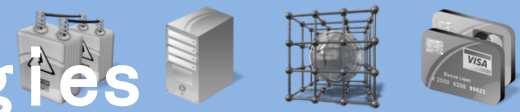
Region: US East (Virginia)		Region: US East (Virginia)				
Linux/UNIX Usage		1 yr Term		3 yr Term	Linux/UNIX Usage	Windows Usage
Standard On-Demand Instances		Standard Reserved Instances				
Small (Default)	\$0.085 per hour	Small (Default)	\$227.50	\$350	\$0.03 per hour	\$0.05 per hour
Large	\$0.34 per hour	Large	\$910	\$1400	\$0.12 per hour	\$0.20 per hour
Extra Large	\$0.68 per hour	Extra Large	\$1820	\$2800	\$0.24 per hour	\$0.40 per hour
Micro On-Demand Instances		Micro Reserved Instances				
Micro	\$0.02 per hour	Micro	\$54	\$82	\$0.007 per hour	\$0.013 per hour
Hi-Memory On-Demand Instances		High-Memory Reserved Instances				
Extra Large	\$0.50 per hour	Extra Large	\$1325	\$2000	\$0.17 per hour	\$0.24 per hour
Double Extra Large	\$1.00 per hour	Double Extra Large	\$2650	\$4000	\$0.34 per hour	\$0.48 per hour
Quadruple Extra Large	\$2.00 per hour	Quadruple Extra Large	\$5300	\$8000	\$0.68 per hour	\$0.96 per hour
Hi-CPU On-Demand Instances		High-CPU Reserved Instances				
Medium	\$0.17 per hour	Medium	\$455	\$700	\$0.06 per hour	\$0.125 per hour
Extra Large	\$0.68 per hour	Extra Large	\$1820	\$2800	\$0.24 per hour	\$0.50 per hour
Cluster Compute Instances		Cluster Compute Reserved Instances				
Quadruple Extra Large	\$1.60 per hour	Quadruple Extra Large	\$4290	\$6590	\$0.56 per hour	N/A*
Cluster GPU Instances		Cluster GPU Reserved Instances				
Quadruple Extra Large	\$2.10 per hour	Quadruple Extra Large	\$5630	\$8650	\$0.74 per hour	N/A*

* Windows® is not currently available for Cluster Compute or Cluster GPU Instance

* Windows® is not currently available for Cluster Compute or Cluster GPU Instances

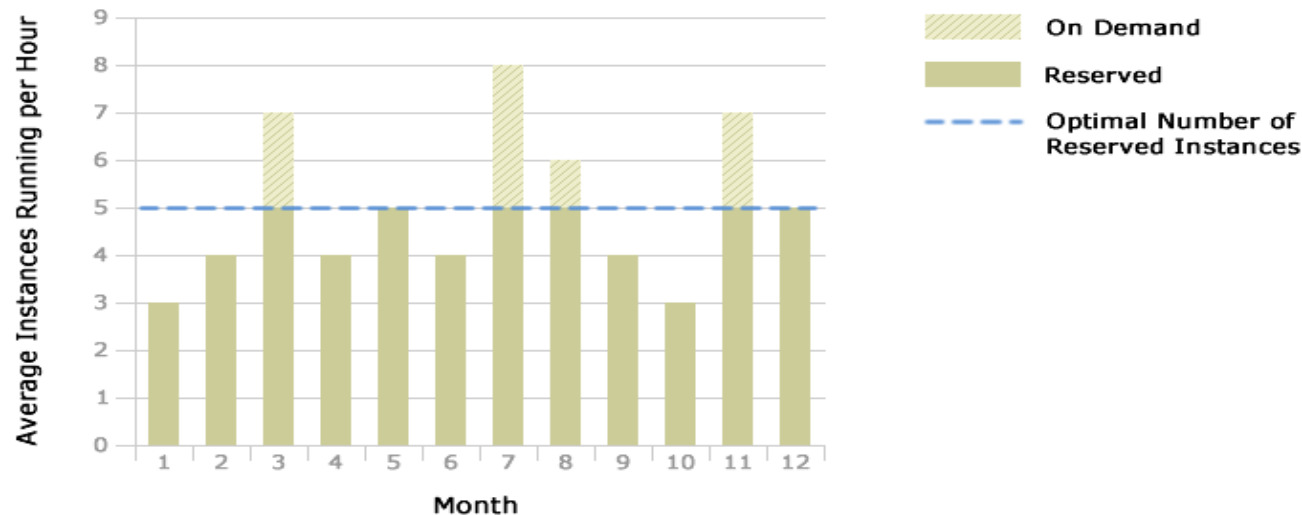


Instance type & pricing Strategies

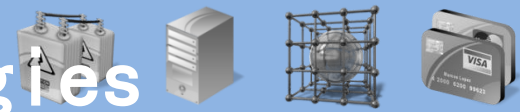


上海交通大学 软件学院 高可靠实验室

Linux/UNIX-Based Example		Windows-Based Example	
Effective Hourly Rate Comparison*			
Annual Utilization	On-Demand	1 Year Term Reserved	3 Year Term Reserved
30%	\$0.68	\$0.93	\$0.60
55%	\$0.68	\$0.62	\$0.43
75%	\$0.68	\$0.52	\$0.38
100%	\$0.68	\$0.45	\$0.35



Instance type & pricing Strategies

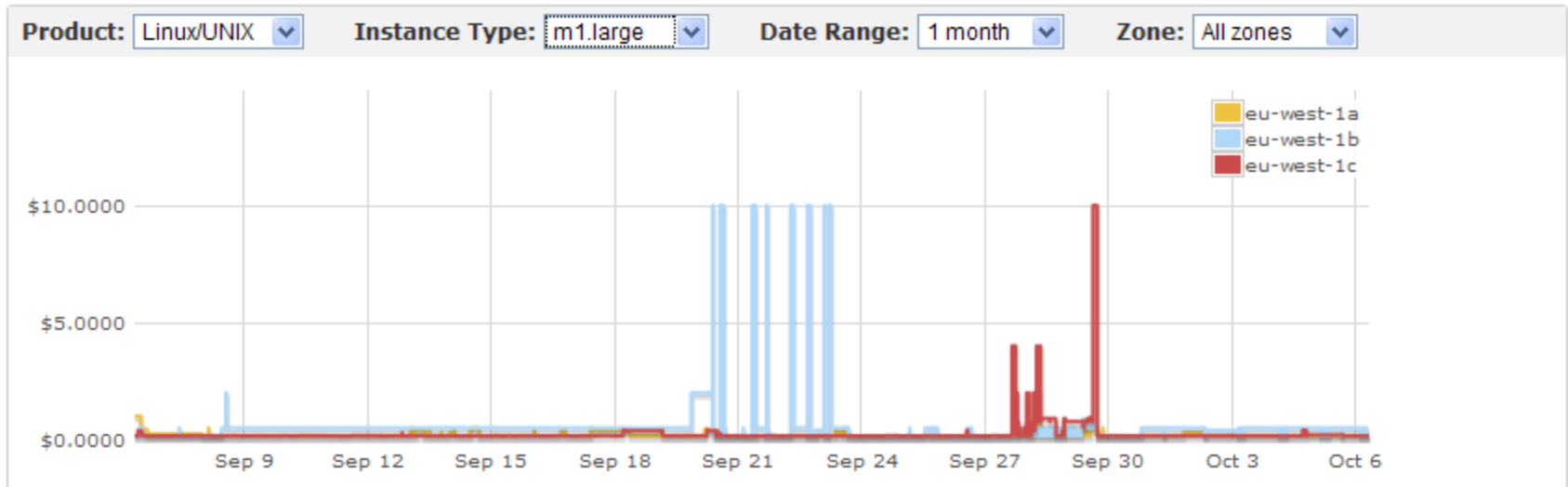


上海交通大学 软件学院 高可靠实验室

➤ Spot

Spot Instance Pricing History

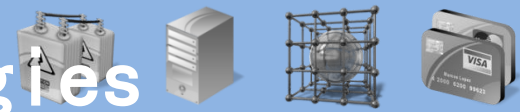
Cancel X



Close



Instance type & pricing Strategies



上海交通大学 软件学院 高可靠实验室

Amazon Elastic Block Store

Region

Internet Data Transfer

The pricing below is based on data transferred "in" and "out" of Amazon EC2.

Region: Asia Pacific (Tokyo)

Data Transfer

All data

Data Transfer

First 1 GB

Up to 1 GB

Next 40 GB

Next 10 GB

Next 35 GB

Next 52 GB

Next 4 GB

Greater

Elastic IP Addresses

Region

No cost

Amazon CloudWatch

Region: Asia Pacific (Tokyo)

Detailed Monitoring for Amazon EC2 Instances

\$3

Basic Monitoring

\$0

Monitoring Alarms

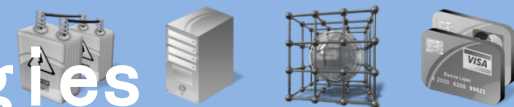
\$0

Elastic Load Balancing

Region: Asia Pacific (Tokyo)

- \$0.028 per Elastic Load Balancer-hour (or partial hour)
- \$0.008 per GB of data processed by an Elastic Load Balancer

Instance type & pricing Strategies



上海交通大学 软件学院 高可靠实验室



NEW! - [Introducing Amazon ElastiCache](#)

FREE USAGE TIER: New Customers get free usage tier for first 12 months

Services

Estimate of your Monthly Bill (\$ 115.31)

Choose region: US-East Northern Vir

Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month ☒

Amazon EC2

Amazon S3

Amazon SQS

Amazon SES

Amazon SNS

Amazon Route 53

Amazon CloudFront

Amazon RDS

Amazon ElastiCache

Amazon CloudWatch

Amazon SimpleDB

Amazon VPC

Amazon Elastic MapReduce

AWS Import Export

AWS Premium Support

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier provides persistent storage to Amazon EC2 instances.

+ Compute: Amazon EC2 On-Demand Instances:

	Instances	Description	Operating System	Instance Type	Usage	Detailed Monitoring
<input checked="" type="checkbox"/>	1		Linux/OpenSolaris	Large	21 Hours/Week	<input checked="" type="checkbox"/>

+ Compute: Amazon EC2 Reserved Instances:

	Instances	Description	OS	Type	Term	Usage
<input checked="" type="checkbox"/>	1		Linux	High-MEM Extra Large	1 yr t	168 Hours/Month

+ Storage: Amazon EBS Volumes:

	Volumes	Description	Provisioned Storage	Average IOPS in volume	Snapshot Storage*
<input checked="" type="checkbox"/>	2		5 GB-month	100	2 GB-month of Storage

Elastic IP:

Number of Elastic IPs:

Elastic IP Non-attached Time:
Hours/Month

Number of Elastic IP Remaps: Times/Month

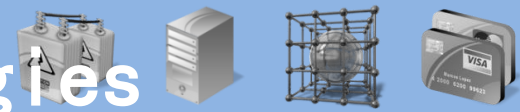
Amazon EC2 Data Transfer:

Data Transfer In: GB/Month

Data Transfer Out: GB/Month

Regional Data Transfer: GB/Month



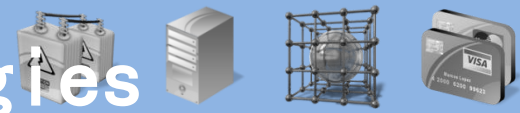


➤ Microsoft Azure

Compute Instance Size	CPU	Memory	Instance Storage	I/O Performance	Cost Per Hour
Extra Small	1.0 GHz	768 MB	20 GB	Low	\$0.04
Small	1.6 GHz	1.75 GB	225 GB	Moderate	\$0.12
Medium	2 x 1.6 GHz	3.5 GB	490 GB	High	\$0.24
Large	4 x 1.6 GHz	7 GB	1,000 GB	High	\$0.48
Extra Large	8 x 1.6 GHz	14 GB	2,040 GB	High	\$0.96



Instance type & pricing Strategies



上海交通大学 软件学院 高可靠实验室

Pricing details for data transfers

North America and Europe regions: \$0.15 per GB out

Asia Pacific Region: \$0.20 per GB out

Standard pay-as-you-go pricing for storage

\$0.15 per GB stored per month based on the daily average

\$0.01 per 10,000 storage transactions

Standard pay-as-you-go monthly pricing for the CDN

\$0.15 per GB for data transfers from European and North American locations

\$0.20 per GB for data transfers from other locations

\$0.01 per 10,000 transactions

Standard pay-as-you-go pricing for caching

128 MB cache for \$45.00

256 MB cache for \$55.00

512 MB cache for \$75.00

1 GB cache for \$110.00

2 GB cache for \$180.00

4 GB cache for \$325.00

Database, based on size of the database:

The SQL Azure database is available in two editions: Web and Business

The Web Edition Relational Database provides up to 5 GB of T-SQL k edition is best suited for Web application, and Departmental custom

Standard pay-as-you-go (Web edition) pricing

\$9.99 per database up to 1GB per month

\$49.95 per database up to 5GB per month

The Business Edition SQL Azure DB provides up to 50 GB of T-SQL k edition is best suited for SaaS ISV apps, custom Web application, and

Standard pay-as-you-go (Business Edition) pricing

\$99.99 per database up to 10GB per month

\$199.98 per database up to 20GB per month

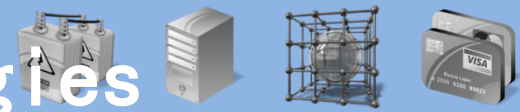
\$299.97 per database up to 30GB per month

\$399.96 per database up to 40GB per month

\$499.95 per database up to 50GB per month




Instance type & pricing Strategies



上海交通大学 软件学院 高可靠实验室

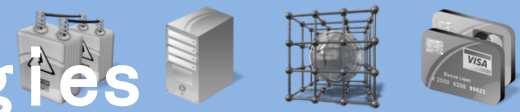
➤ IBM Smart Cloud

Virtual machine instances									
Virtual machines 32-bit configurations		Copper	Bronze	Silver	Gold	Customer scenarios		Software Charge	Infrastructure Charge
Virtual CPUs with 1.25GHz		1	1	2	4	You "bring your own IBM license" ("BYOL") You own an IBM software license and can use the pre-built IBM images in the portal catalog		Prepaid for software license	Per virtual machine (VM) per hour
Virtual memory (Gigabytes)		2	Internet data transfer Price per GB transferred in or out No charge until 30 September 2011, then First 10 TB Next 40 TB (10 TB up to 50 TB) Next 100 TB (50 TB up to 150 TB) All additional usage above 150 TB			You "pay-as-you-go" ("PAYG") You choose the desired software, accept the license terms online, and receive a monthly usage bill		Per Image per hour	Per VM per hour
Instance storage (Gigabytes)		60				You "bring your own software and licenses" You bring your own software or software for which you hold valid licenses and install them on the servers you provision		Prepaid for software licenses	Per VM per hour
Reserved price per hour (in addition to monthly reservation fee; see Reserved capacity table below)						You want to test "pre-release" software From time to time, pre-released software images will be made available on a temporary basis for test (non-productive) use		No charge for restricted use	Per VM per hour
with Red Hat Linux OS		£0.061				You are an eligible ISV/SI developer You can use selected IBM "development use only" ("DUO") software for development, test, proof of concept and sales demos on the IBM SmartCloud		No charge for restricted use	Per VM per hour
with Novell SUSE Linux OS		£0.041				Options available vary by software package. For software image descriptions, see the software images page			
with Windows Server		£0.044							
Unreserved price per hour									
with Red Hat Linux OS		£0.086	£0.10	£0.158	£0.247				
with Novell SUSE Linux OS		£0.065	£0.079	£0.137	£0.227				
with Windows Server		£0.069	£0.082	£0.165	£0.254				
Virtual machines 64-bit configurations		Copper	Bronze	Silver	Gold	Platinum			
Virtual CPUs with 1.25GHz		2	2	4	8	16			
Virtual memory (Gigabytes)		4	4	8	16	16			
Instance storage (Gigabytes)		60	850	1024	1024	2048			
Reserved price per hour (in addition to monthly reservation fee; see Reserved capacity table below)									
with Red Hat Linux OS		£0.113	£0.144	£0.172	£0.254	£0.495			
with Novell SUSE Linux OS		£0.093	£0.124	£0.151	£0.234	£0.447			
with Windows Server		£0.114	£0.151	£0.188	£0.270	£0.655			

 Snapshot

 [Start over](#)

[← Previous](#) [Next →](#)



➤ Google App Engine

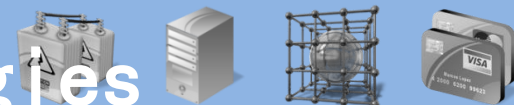
Billable Quota Unit Cost

The cost for computing resources is as follows:

Resource	Unit	Unit cost
Outgoing Bandwidth	gigabytes	\$0.12
Incoming Bandwidth	gigabytes	\$0.10
CPU Time	CPU hours	\$0.10
Stored Data	gigabytes per month	\$0.15
High Replication Storage	gigabytes per month	\$0.45
Recipients Emailed	recipients	\$0.0001
Always On	N/A (daily)	\$0.30
Backends (B1 class)	Hourly per instance	\$0.08
Backends (B2 class)	Hourly per instance	\$0.16
Backends (B4 class)	Hourly per instance	\$0.32
Backends (B8 class)	Hourly per instance	\$0.64



Instance type & pricing Strategies

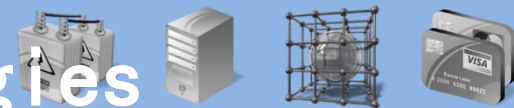


学院 高可靠实验室

		Free quota per app per day	Price
	On-demand Frontend Instances	28 free instance hours	\$0.08 / hour
Platf	Reserved Frontend Instances		\$0.05 / hour
	High Replication Datastore	1G	\$0.24 / G / month
Dyn	Outgoing Bandwidth	1G	\$0.15 / G
Java	Incoming Bandwidth	1G	\$0.10 / G
Pyth			\$0.10/100k write ops
Usa	Datastore API	50k free read/write/small	\$0.07/100k read ops \$0.01/100k small ops
Infin	Blobstore API	5G	\$0.17 / G / month
SLA	Email API	100 recipients	\$0.01 / 100 recipients
Ope	XMPP API	1000 stanzas	\$0.01 / 1k stanza
Tool	Channel API	100 channels opened	\$0.01 / 100 channels opened
	Image Manipulation API	✓	✓
Go	Memcache API	✓	✓
Coc	Users API	✓	✓
Gra	Task Queue	✓	✓
Rec	Files API	✓	✓
Dev	URL Fetch API	✓	✓
	Cron	✓	✓
	Prospective Search API (experimental)	✓	✓



Instance type & pricing Strategies



上海交通大学 软件学院 高可靠实验室

Google app engine

wwwtvanessa@gmail.com | [My Account](#) | [Help](#) | [Sign out](#)

A comparison of your current bill against the [new pricing model](#) is now available in the Billing History page.

[Dismiss](#)

Application: wwwtvanessa [High Replication] No version deployed!

[« My Applications](#)

Main

[Dashboard](#)

[Instances](#)

[Logs](#)

[Versions](#)

[Backends](#)

[Cron Jobs](#)

[Task Queues](#)

[Quota Details](#)

Data

[Datastore Indexes](#)

[Datastore Viewer](#)

[Datastore Statistics](#)

[Blob Viewer](#)

[Prospective Search](#)

[Datastore Admin](#)

Administration

[Application Settings](#)

[Permissions](#)

Billing Status: Free

This application is operating within the free quota levels. Enable billing to grow beyond the free quotas. [Learn more](#)

[Enable Billing](#)

[Transfer app to a premier account](#)

Billing Administrator: None

Since this application is operating within the free quota levels, there isn't a billing administrator.

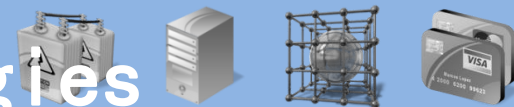
Current Balance: n/a [Usage History](#)

Resource Allocations:

Resource	Budget	Unit Cost	Paid Quota	Free Quota	Total Daily Quota
CPU Time	n/a	\$0.10/CPU hour	n/a	6.50	6.50
Bandwidth Out	n/a	\$0.12/GByte	n/a	1.00	1.00
Bandwidth In	n/a	\$0.10/GByte	n/a	1.00	1.00
Stored Data	n/a	\$0.005/GByte-day	n/a	1.00	1.00
Recipients Emailed	n/a	\$0.10/1000 Emails	n/a	2.00	2.00
High Replication Storage	n/a	\$0.008/GByte-day	n/a	0.50	0.50
Backend Usage	n/a	Prices	n/a	\$0.72	\$0.72
Always On	n/a	\$0.30/Day	n/a	none	
Max Daily Budget:	n/a				



Instance type & pricing Strategies



Google app engine

wwtvanessa@gmail.com | [My Account](#) | [Help](#) | [Sign out](#)

靠实验室

A comparison of your current bill against the [new pricing model](#) is now available in the Billing History page.

[Dismiss](#)

Application: wwtvanessa [High Replication] No version deployed!

[« My Applications](#)

Main

[Dashboard](#)

[Instances](#)

[Logs](#)

[Versions](#)

[Backends](#)

[Cron Jobs](#)

[Task Queues](#)

[Quota Details](#)

Data

[Datastore Indexes](#)

[Datastore Viewer](#)

[Datastore Statistics](#)

[Blob Viewer](#)

[Prospective Search](#)

[Datastore Admin](#)

Administration

[Application Settings](#)

[Permissions](#)

[Blacklist](#)

[Admin Logs](#)

Billing

[Billing Settings](#)

[Billing History](#)

Resources

[Documentation](#)

Set Budget

Max Daily Budget:

Budget Preset:

Optional

- CPU Intensive
- Standard
- CPU Intensive
- Bandwidth Intensive
- Storage Intensive
- Custom

	Budget	Unit Cost	Paid	Free	Total Daily Quota
Bandwidth Intensive	\$3.75	\$0.10 / CPU hour	37.50	6.50	44.00
Storage Intensive	\$0.40	\$0.12 / GByte	3.33	1.00	4.33
Bandwidth In 2%	\$0.10	\$0.10 / GByte	1.00	1.00	2.00
Stored Data 5%	\$0.25	\$0.005 / GByte-day	50.00	1.00	51.00
High Replication Storage 10% (limit \$40)	\$0.50	\$0.008 / GByte-day	62.50	0.50	63.00
Recipients Emailed 0%	\$0.00	\$0.10 / 1000 Emails	0.00	2.00	2.00
Backend Usage ^{New!} 0%	\$0.00	Prices	\$0.00	\$0.72	\$0.72

Select Additional Resources

Discounted Instance Hours: / Week

Note: Discounted instance hours will be available once the [new pricing model](#) comes into effect. For now we will only record the number of hours you wish to commit to. Committing to a number of instance hours for a weekly billing period in advance can help lower your bill. However, once new pricing is live you will be charged \$0.05 per hour for the hours you commit to here, even if you don't use them during the billing period.

When you authorize a weekly payment you will need to authorize for your normal weekly budget amount, as well as the cost of all your discounted instance hours. We do this so we can charge the cost of the hours you've committed to at the end of a billing period, even if you consume your daily budget with non-instance-hour charges every day.

☐ Always On \$0.30 per day (\$9.00 per month) [Learn more](#)

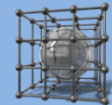
Set Country

Your Country:

Please select...



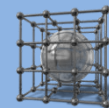
System Property	<u>Amazon</u> Elastic Compute Cloud (EC2)	<u>Google</u> App Engine	<u>Microsoft</u> Live Mesh	<u>Sun</u> Network.com (Sun Grid)	<u>GRIDS Lab</u> Aneka
Focus	Infrastructure	Platform	Infrastructure	Infrastructure	Software Platform for enterprise Clouds
Service Type	Compute, Storage (Amazon S3)	Web application	Storage	Compute	Compute
Virtualisation	OS Level running on a Xen hypervisor	Application container	OS level	Job management system (Sun Grid Engine)	Resource Manager and Scheduler
Dynamic Negotiation of QoS Parameters	None	None	None	None	SLA-based Resource Reservation on Aneka side.
User Access Interface	Amazon EC2 Command-line Tools	Web-based Administration Console	Web-based Live Desktop and any devices with Live Mesh installed	Job submission scripts, Sun Grid Web portal	Workbench, Web-based portal
Web APIs	Yes	Yes	Unknown	Yes	Yes
Value-added Service Providers	Yes	No	No	Yes	No
Programming Framework	Customizable Linux-based Amazon Machine Image (AMI)	Python	Not applicable	Solaris OS, Java, C, C++, FORTRAN	APIs supporting different programming models in C# and other .Net supported languages



- Motivation
- Open Issues & Related Worked
- Industrial Example
- Summary



Industrial Example

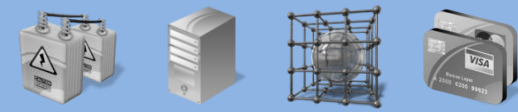


上海交通大学 软件学院 高可靠实验室

- Provider
 - Google app engine

- User
 - Amazon, Microsoft……

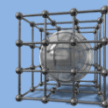




➤ Amazon: Auto scaling

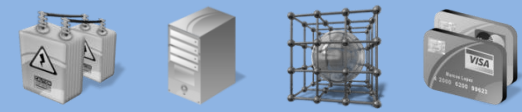
- Manual Scaling
- Scaling by Schedule
- Scaling by Policy
- Auto Scaling Group
 - An Auto Scaling group is a representation of multiple Amazon EC2 instances that share similar characteristics, and that are treated as a logical grouping for the purposes of instance scaling and management.
- Health Check
 - A *health check* is a call to check on the health status of each instance in an Auto Scaling group.
- Launch Configuration
 - A launch configuration captures the parameters necessary to create new EC2 instances.





- Trigger
 - Alarm
 - a CloudWatch alarm
 - An Amazon CloudWatch *alarm* is an object that watches over a single metric.
 - Policy
 - A *policy* is a set of instructions for Auto Scaling that tells the service how to respond to CloudWatch alarm messages.



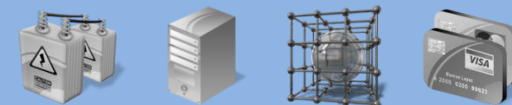


➤ Scaling Activity

- Auto Scaling Instance Termination
- Cooldown
- Instance Distribution and Balance Across Multiple Zones
 - Auto Scaling attempts to **distribute instances evenly between the Availability Zones** that are enabled for your Auto Scaling group.
 - Certain operations and conditions can cause your Auto Scaling group to become unbalanced. Auto Scaling compensates by creating **a rebalancing activity**
 - Auto Scaling always launches new instances before attempting to terminate old ones, so **a rebalancing activity will not compromise the performance or availability of your application.**



Industrial Example



上海交通大学 软件学院 高可靠实验室

Monitoring Dashboard

Overview of Your Alarms

Your CloudWatch Alarms			
Create Alarm Modify Delete			
Viewing: All alarms			
1 to 3 of 3 Items			
	State	Name	Threshold
<input type="checkbox"/>	ALARM	Fleet CPU	CPUUtilization is < 20 for 15 minutes
<input type="checkbox"/>	ALARM	DiskWriteOpsForMicros	DiskWriteOps is < 10 for 10 minutes
<input checked="" type="checkbox"/>	OK	DiskWriteOps for instance	DiskWriteOps is >= 10 for 30 minutes

1 Alarm selected

Alarm: DiskWriteOps for instance

Description

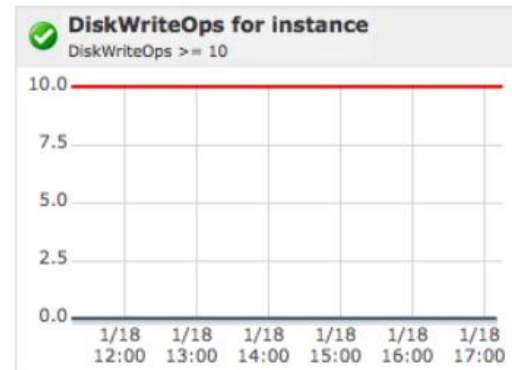
Metric

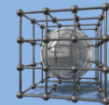
State Details: State changed to ' OK' at 2011/01/13 13:58 UTC. Reason: Threshold Crossed: 1 datapoint (0.0) was not greater than or equal to the threshold (10).

Description: Disk Write Ops is high

Threshold: DiskWriteOps is >= 10 for 30 minutes

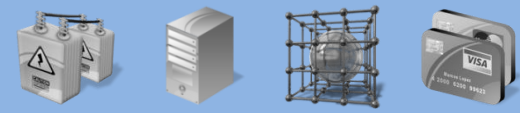
Actions: in ALARM state - Send message to topic "Briande" (briande@amazon.com)
in INSUFFICIENT_DATA state -





- Motivation
- Open Issues & Related Worked
- Industrial Example
- Summary

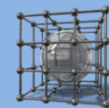




➤ Scale up or down

- Who should decide to scale up/down?
- When to scale up/down?
- Which tier should be scaled up/down?
- How many VMs should be added or reduced?
- What is the policy of scaling?
- How to add/ reduced? Resize or quantity change?
- Which type of VM should be added/reduced?
- ☒ Where the new VM should be placed? Or which old VMs should be terminated?





Thank you~

