Flexible Distributed Capacity Allocation and Load Redirect Algorithms for Cloud Systems

Danilo Ardagna\textsuperscript{1}, Sara Casolari\textsuperscript{2}, Barbara Panicucci\textsuperscript{1}

\textsuperscript{1}Politecnico di Milano, Dipartimento di Elettronica e Informazione, Italy
\textsuperscript{2}Università di Modena e Reggio Emilia, Dipartimento di Ingegneria dell’Informazione, Italy

IEEE Cloud 2011, Washington DC, 6\textsuperscript{th} July 2011
Research scenario

• Cloud Computing:
  ◆ On-demand software/hardware delivering on a pay per use basis
  ◆ End-users obtain the benefits of the infrastructure without the need to implement it directly
  ◆ Cloud providers maximize the utilization of their physical resources, minimizing energy costs and obtaining economies of scale

• Major issues:
  ◆ Development of efficient service provisioning policies
  ◆ Modern Clouds live in an open world characterized by continuous changes which occur autonomously and unpredictably
Our contribution

• Workload prediction-based **capacity allocation techniques** able to coordinate **multiple distributed resource controllers**

• Dynamic **load redirection mechanism** which determines the requests to be redirected during peak loads

• Requests’ distribution **optimized** according to the average response time
Problem statement

- **WS provider perspective** offering multiple transactional WSs hosted at **multiple sites** of an IaaS provider.

- **SLA contract**, associated with each WS class $k$ specifies the QoS levels: $R_k \leq \bar{R}_k$.

- **WSs** are deployed in **VMs**, each VM hosts a **single** Web service application.
Problem statement

- Multiple (homogeneous) VMs implementing the same WS class can **run in parallel**
- Services can be located on **multiple sites**
- IaaS provider **charges** the WS provider **on a hourly basis**
Problem statement

- **Capacity Allocation (CA):** Determine the optimal number of VMs for each WS class in each IaaS site according to a prediction of the incoming workload ($T_1$ mid-long time scale)

- **Load Redirection (LR):** If a site resources are insufficient, incoming requests redirected to other sites ($T_2 << T_1$ short-term time scale)
Cloud System Reference Framework

Local workload manager

Virtualized Servers

IaaS Provider

Flat VMs

On demand VMs

Local WS arrival rates

Execution rate of local arrivals

Redirect rate of local arrivals

Local CA and LR manager

Virtualized Servers

$\Lambda_1^2, \Lambda_2^2, ..., \Lambda_k^2$

$Z_1^3, Z_2^3, ..., Z_k^3$

$x_1^3, x_2^3, ..., x_k^3$

$N_1^2 + M_1^2$
Prediction model time scales

![Graph showing workload predictions and time intervals with labels T1 and T2.]

- T1
- T2
- Annotations for workload predictions (\^\Lambda_k^i)
<table>
<thead>
<tr>
<th>( I )</th>
<th>Set of sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K )</td>
<td>Set of WS classes</td>
</tr>
<tr>
<td>( C^i )</td>
<td>VM instances capacity at site ( i )</td>
</tr>
<tr>
<td>( \bar{c}^i )</td>
<td>Time unit cost for flat VMs at site ( i )</td>
</tr>
<tr>
<td>( \bar{c}^i )</td>
<td>Time unit cost for on demand VMs at site ( i )</td>
</tr>
<tr>
<td>( N^i )</td>
<td>Number of flat VMs available at site ( i )</td>
</tr>
<tr>
<td>( \mu_k )</td>
<td>Maximum service rate of a capacity 1 VM for executing WS class ( k ) requests</td>
</tr>
<tr>
<td>( d^{i,j}, i \neq j )</td>
<td>Delay (s) for requests redirecting from site ( i ) to site ( j )</td>
</tr>
<tr>
<td>( g^{i,j} = \frac{1}{d^{i,j}}, i \neq j )</td>
<td>“Conductance” of the communication link ((i,j))</td>
</tr>
<tr>
<td>( G^i = \sum_{j} g^{i,j}, i \neq j )</td>
<td>“Equivalent conductance” seen from site ( i ) to the other sites</td>
</tr>
</tbody>
</table>
### Problem formulation – Decision variables

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N^i_k$</td>
<td>Number of \textit{flat} VMs allocated for class $k$ request at site $i$</td>
</tr>
<tr>
<td>$M^i_k$</td>
<td>Number of \textit{on demand} VMs allocated for class $k$ request at site $i$</td>
</tr>
<tr>
<td>$x^i_k$</td>
<td>Execution rate of local arrivals for WS class $k$ request at site $i$</td>
</tr>
<tr>
<td>$z^i_k$</td>
<td>Redirect of WS class $k$ request at site $i$ toward other sites</td>
</tr>
</tbody>
</table>
Design assumptions

- Each WS class hosted in a VM is modeled as an M/G/1-PS queue.

- The fraction of workload redirected to other sites is inversely proportional to the network delay/\textbf{directly proportional} to the “conductance” \( g_{i,j} = 1/d_{i,j} \).

- The overall load at site \( i \) due to the redirect of other sites is given by:

\[
\sum_{j \in I, j \neq i} \frac{g_{j,i} z_{j}^{i} G_{j}}{G_{j}}
\]

- The total rate of class \( k \) requests executed at site \( i \) is given by:

\[
x_{k}^{i} + \sum_{j \neq i} \frac{g_{j,i} z_{j}^{i} G_{j}}{G_{j}}
\]
Prediction models

- **Exponential Smoothing** (ES) models to predict the local arrival rate $\Lambda^i_k$
- **Simple model** motivated by the application context: **Short-time** predictions suitable for **real-time** autonomic decisions
- We consider a version of ES, where parameters are dynamically chosen:

$$
\hat{\Lambda}^i_k(t + T_1) = \gamma^i_k(t)\hat{\Lambda}^i_k(t) + (1 - \gamma^i_k(t))\Lambda^i_k(t), \quad t > T_1
$$

$$
\hat{\Lambda}^i_k(T_1) = \frac{1}{T_1} \sum_{t=1}^{T_1} \Lambda^i_k(t)
$$
Prediction models

- Dynamic ES model by re-evaluating the smoothing factor $\gamma^i_k(t)$ at each prediction sample $t$

- We use the Trigg and Leach procedure:

\[
\gamma^i_k(t) = \frac{A^i_k(t)}{E^i_k(t)}
\]

\[
A^i_k(t) = \phi \epsilon^i_k(t) + (1 - \phi) A^i_k(t - T_1)
\]

\[
E^i_k(t) = \phi |\epsilon^i_k(t)| + (1 - \phi) E^i_k(t - T_1)
\]
Capacity Allocation problem

- The goal of CA problem is to:
  - **Minimize** the overall costs for flat and on demand VM instances of multiple distributed IaaS sites
  - **Guaranteeing** that the average response time of each class is lower than the SLA threshold
  - Solved every $T_1$ time instant

- WS requests’ average response time:

$$R^i_{k} = \frac{1}{C^i \mu_k - \frac{\hat{\Lambda}^i_k}{N^i_k + M^i_k}}$$

$$R_k = \sum_i \frac{\hat{\Lambda}^i_k R^i_k}{\sum_j \hat{\Lambda}^j_k}$$
Capacity Allocation problem

\[
\min_{N_k^i, M_k^i} \sum_k \sum_i c^i N_k^i + c^i M_k^i
\]

\[
\sum_i \frac{\Lambda_k^i (N_k^i + M_k^i)}{C^i \mu_k (N_k^i + M_k^i) - \Lambda_k^i} \leq \bar{R}_k \sum_j \Lambda_j^j
\]

\[
\sum_{k \in K} N_k^i \leq \bar{N}^i, \forall i \in I
\]

Flat and on-demand instances costs

Resources are not saturated

Average response time lower than the threshold

Flat VMs ≤ the ones available
Load Redirect problem

• The goal of LR problem is to:
  ◆ Cooperatively minimize request average response times
  ◆ Avoid episodic local congestions due to the variability of the incoming workload
  ◆ Solved every $T_2$ time instant

• WS requests’ average response time:

$$\hat{R}_k^i = \frac{1}{C^i \mu_k - \left( x_k^i + \sum_{j \neq i} \frac{g_{j,i} z_j^k}{G_j^k} \right)}$$

$$R_k^i = \hat{R}_k^i + \sum_{j \neq i} \frac{z_j^k}{G_j^k} \left( x_k^i + \sum_{j \neq i} \frac{g_{j,i} z_j^k}{G_j^k} \right)$$
Load Redirect problem

\[
\min_{x^i_k, z^i_k} \sum_k \sum_i \left[ \frac{(N^i_k + M^i_k) \left( x^i_k + \sum_{j \neq i} g^{j,i} z^j_k \right)}{C^i \mu_k (N^i_k + M^i_k) - (x^i_k + \sum_{j \neq i} g^{j,i} z^j_k)} + \sum_{j \neq i} \frac{z^j_k}{G^j} \right]
\]

\[
x^i_k + z^i_k = \tilde{\Lambda}^i_k, \quad \forall k \in K, i \in I
\]

\[
x^i_k + \sum_{j \neq i} \frac{g^{j,i} z^j_k}{G^j} < C^i \mu_k (N^i_k + M^i_k)
\]

\[
x^i_k, z^i_k \geq 0
\]

Distributed decomposable solution relying on Lagrangian techniques

Requests are locally executed or redirected

Resources are not saturated
Load Redirect problem decomposition

\[
\begin{align*}
\min_{x^i, y^i, z^i, w^i} \sum_i & \left[ \frac{(N^i + M^i) (x^i + y^i)}{C^i \mu (N^i + M^i)} - (x^i + y^i) \right] + w^i \\
& x^i + z^i = \hat{\Lambda}^i \quad \forall i \in I \\
& x^i + y^i < C^i \mu (N^i + M^i) \quad \forall i \in I \\
& y^i = \sum_{j \neq i} \frac{g^{j,i} z^j}{G^j} \quad \forall i \in I \\
& w^i = \sum_{j \neq i} \frac{z^j}{G^j} \quad \forall i \in I \\
& x^i, y^i, z^i, w^i \geq 0 \quad \forall i \in I
\end{align*}
\]
Duality Theory

Primal problem

D* → P*

Dual problem

P* → D*

Strong duality

D* ↔ P*

Lagrangian relaxation (LB)

Any feasible solution (UB)
Load Redirect problem Lagrangian relaxation

\[
\min_{x^i, y^i, z^i, w^i} \sum_i \left[ \frac{(N^i + M^i) (x^i + y^i)}{C^i \mu (N^i + M^i) - (x^i + y^i)} + w^i + \Theta_i (y^i - \sum_{j \neq i} G_j \frac{g^{j,i} z^j}{G_j}) + \eta_i (w^i - \sum_{j \neq i} \frac{z^j}{G_j}) \right]
\]

\[x^i + z^i = \Lambda^i \quad \forall i \in I\]
\[x^i + y^i < C^i \mu (N^i + M^i) \quad \forall i \in I\]
\[x^i, y^i, z^i, w^i \geq 0 \quad \forall i \in I\]

The relaxed problem further separates into \(|I|\) sub-problems.
Dual decomposition

- For a given set of $\Theta_i$’s and $\eta_i$’s defines the dual function $L(\Theta, \eta)$ and the dual problem is then given by:

$$\max_{\Theta, \eta} L(\Theta, \eta)$$

- The dual problem can be solved by using a sub-gradient method:

$$\Theta_i(t + 1) = \Theta_i(t) + \alpha_t \left( y^i - \sum_{j \neq i} \frac{g^{j,i} z^j}{G^j} \right)$$

$$\eta_i(t + 1) = \eta_i(t) + \beta_t \left( w^i - \sum_{j \neq i} \frac{z^j}{G_j} \right)$$
Experimental results – Scalability analysis

• **Large set of randomly** generated instances:
  - $|I|$ has been varied between 20 and 60
  - $|K|$ has been varied between 100 and 1000

• Average **execution time** required to solve instances of maximum size is **lower than 3 minutes and one minute** for the CA and LR problems, respectively
Experimental results – Comparison with alternative methods

• **Heuristic 1:**
  - The CA is performed every 5 minutes and the number of VMs is determined according to utilization thresholds.
  - The number of VMs is determined such that the utilization of the VMs is equal to a given threshold $\tau_1$.
  - VM provisioning is **further triggered** if the prediction of the VMs utilization is higher than a second threshold $\tau_2 > \tau_1$.
  - **Multiple analyses** have been performed by adopting different thresholds: $(\tau_1, \tau_2) = (40\%, 50\%), (50\%, 60\%), \text{and} (60\%, 80\%)$.
Experimental results – Comparison with alternative methods

• **Heuristic 2**: Same as Heuristic 1 but the number of VMs is determined by *optimally solving* our **CA** problem every 5 minutes.

• **Heuristic 3**: Same as Heuristic 2 but with a **10 minutes** time horizon.
Experimental results – Comparison with alternative methods

- **Local incoming workload** has been obtained from the traces of a very large dynamic Web-based system:
  - **Normal day scenario**: It describes the baseline workload (bi-modal requests profile)
  - **Heavy day scenario**: It exhibits a 40% increment in the number of the client requests with respect to the baseline
  - **Noisy day scenario**: It is characterized by the same request profile belonging to the heavy day scenario with an additional noise component
Experimental results – Comparison with alternative methods

![Graph showing comparison of methods]

- Our Solution
- Heuristic 2
- Heuristic 3
- Heuristic 1 (60%, 80%)
- Heuristic 1 (50%, 60%)
- Heuristic 1 (40%, 50%)

Time [5 min]
## Experimental results – Comparison with alternative methods

<table>
<thead>
<tr>
<th>Alternative solution</th>
<th>Normal day</th>
<th>% Savings Heavy day</th>
<th>Noisy day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic 1 - (40%, 50%)</td>
<td>35.47</td>
<td>34.86</td>
<td>36.84</td>
</tr>
<tr>
<td>Heuristic 1 - (50%, 60%)</td>
<td>19.53</td>
<td>18.83</td>
<td>21.4</td>
</tr>
<tr>
<td>Heuristic 1 - (60%, 80%)</td>
<td>3.12</td>
<td>2.25</td>
<td>4.93</td>
</tr>
<tr>
<td>Heuristic 2</td>
<td>4.3</td>
<td>3.26</td>
<td>4.44</td>
</tr>
<tr>
<td>Heuristic 3</td>
<td>11.56</td>
<td>10.27</td>
<td>6.98</td>
</tr>
</tbody>
</table>
Experimental results – Validation on Amazon EC2

![Graph showing response time over time with peaks and troughs.]

Response time [sec]

Time [10 sec]
Conclusions and future work

• Prediction-based distributed CA and LR algorithms for IaaS cloud system minimizing the cost of the running VMs

• Experimental results shown that our solutions significantly improve other heuristics

• Future work will extend the validation of our solution considering a larger experimental setup
Questions?

Danilo Ardagna
Politecnico di Milano,
Dipartimento di Elettronica e Informazione, Italy
ardagna@elet.polimi.it

Thank you!