Análise e Modelagem de Desempenho de Sistemas de Computação

``Capacity management & datacenters’’

Virgilio A. F. Almeida

1st semester 2009
Week #9b

Computer Science Department
Federal University of Minas Gerais
Brazil
Capacity Management for Modern Virtualized IT Infrastructures

1. Motivation

2. Research results
Twitter is currently down for Unplanned maintenance.
We expect to be back in about an hour. Thanks for your patience.

IT’S COOL.
I CAN CHILL.

HURRY UP.
Motivation for Autonomic Capacity Management and Planning

• Modern IT infrastructure are complex and composed of multiple tiers

• The workload presents long term variations and short term variations with high peak-to-average ratios

• Service level has many different metrics: performance, availability, security, costs

• Many software and hardware parameters influence system performance:
  – Manual reconfiguration is not an option
  – Need autonomic systems
The Capacity Planning Scenario

- **Time Scale**
  - Quarter
  - Week
  - Hour

- **Resource Provisioning**
- **Workload Forecast**
- **Optimization Model**

- **SLA violations**
- **Revenue/loss**

- **Analysis & Reconfiguration**
- **Business goals**

- **Monitoring & Management**
- **SLA violations**
- **Revenue/loss**

- **Performance Model**

- **Hour Monitoring & Optimization Model**
- **Workload Forecast**
- **Optimization Model**

- **SLA**
Capacity Planning Process

1. Business Models & Measurable Goals
2. Understand Service Architecture
3. Cost-Performance Analysis & Actions
4. Performance & Model
5. Predict Service Performance
6. Characterize the Workload
7. Model Validation And Calibration
8. Obtain Model Parameters
9. Develop a Performance Model
10. Forecast Workload Evolution

- Understand Service Architecture
- Cost-Performance Analysis & Actions
- Performance & Model
- Predict Service Performance
- Characterize the Workload
- Model Validation And Calibration
- Obtain Model Parameters
- Develop a Performance Model
- Forecast Workload Evolution
Workload Characterization

- Process of understanding, quantifying, modeling, and searching for invariants in the workload.
- A *workload model* is a representation of the actual workload.
  - Model representativeness vs. complexity
- Workload models: overlooked issue
- Workload models are useful for:
  - Capacity planning
  - System selection
  - Performance tuning
  - Understanding user behavior
  - Designing more efficient systems
Workload Models

A bridge designer needs to know:

- the bridge’s type of service (e.g., highway, railroad, pedestrian)
- the bridge’s construction material (e.g., steel)
- the bridge’s operating rating, i.e., the absolute maximum permissible load level per vehicle (e.g., 44.1 metric tons).
- average daily traffic (e.g., 5,400 cars, 1,200 trucks)
- Peak traffic at certain times of the day

The bridge designer needs to understand and quantify the bridge’s workload: type of load and load levels.
Workload Characterization
Multi-scale Time Analysis

3600 sec

60 sec

1 sec
Challenges in Managing SLA for Large Data Centers

- Develop workload models for large data centers
- Develop new performance models for utility computing
- Develop efficient integrated capacity models for utility computing for autonomic management purposes.
Modern Mega Datacenters

New York Times

Though deliberately vague when discussing the capabilities of its data centers, Microsoft did divulge that the one in Quincy, Wash., could hold 6.75 trillion photographs. More Photos >

“Today” is nowadays likely to be increasingly large, powerful, energy-intensive, always-on and essentially out-of-sight data centers. These centers run enormously scaled software applications with millions of users. To appreciate the scope of this phenomenon, and its

Yet when it comes to a large company like Microsoft, it can be difficult to find out what any given data center is used for. The company, for reasons ranging from security to competitive advantage, won’t provide too much in the way of details, apart from noting that Quincy could hold 6.75 trillion photos. “We support over 200 online properties with very large scale,” Chrapaty offered. “And so when you think about Hotmail supporting 375 million users, or search supporting three billion queries a month, or Messenger supporting hundreds of millions of users, you can easily assume that those properties are very large properties for our company.”

Firms initially linked from their own centers, but that added precious fractions of milliseconds. So they moved into the data center itself. “If you’re in the facility, you’re eliminating that wire.” The specter of infinitesimal delay is why, when the Philadelphia Stock Exchange, the nation’s oldest, upgraded its trading platform in 2006, it decided to locate the bulk of its trading engines 80 miles — and three milliseconds — from Philadelphia, and into NJ2, where, as Thomas notes, the time to communicate between servers is down to a millionth of a second. (Latency concerns are not limited to Wall Street; it is estimated that a 100-millisecond delay reduces Amazon’s sales by 1 percent.)
“Self-Adaptive SLA-Driven Capacity Management for Internet Services”

Presented by B. Abrahao at IEEE NOMS, Vancouver 2006

Bruno Abrahao, Virgilio Almeida, Jussara Almeida
Universidade Federal de Minas Gerais - Brazil

Alex Zhang, Dirk Beyer, Fereydoon Safai
Hewlett-Packard Labs Palo Alto, CA
Summary

1. Introduction
2. Environment description
3. Self-Adaptive Capacity Management Model
   1. Autonomous Capacity Management Framework
   2. Cost Model
   3. Performance Model
   4. Optimization Model
4. Experimental Analysis
5. Conclusion and future work
Introduction

• Utility Computing
  – Traditional utility services (water, energy, phone, gas, etc.).
  – Capacity on-demand
  – Adaptability
  – Economy of scale

• IT outsourcing
  – Internet Services
  – Independent hosting versus outsourcing
  – Internet Data Centers (IDC)
  – Contracts with providers
Introduction

Internet Data Centers (IDC)

Flexible resource provisioning and quality of service (QoS).
Introduction

• SLA contracts
  
  – Define service rates.
    • e.g. US$ 10,000.00 / mo.
  
  – Performance requirements.
    • e.g. response time should be shorter than 1 sec.
  
  – Penalties for SLA contract violations.
    • e.g. Refund customer US$ 1.00 for each transaction that exceed the response time requirement.
Introduction

• Hosting of multiple services belonging to multiple customers.
• The ability to generate profits depends heavily on the maturity of the operations.
• Objective
  – Manage the capacity and configuration of an IDC in a cost effective manner with the objective of maximizing the financial potential of the provider, while satisfying the customers’ SLA requirements.
• Must consider
  – application requirements, their characteristics and workload variations, as well as satisfying the business goal of the provider.
• Enormous complexity!
Introduction

• Challenges
  – Application heterogeneity.
  – High workload fluctuations.
  – Unexpected workload peaks.
  – Manual management is impractical.
  – Must provide quick adaptability.
Introduction

• New customer demands

1. Per use accounting of services.

2. Probabilistic performance requirements:

\[
\overline{R} \leq R^{SLA} \quad \text{and} \quad P(R > \overline{R}^{SLA}) \leq \alpha
\]

3. Multiple metric requirements (e.g., throughput AND response time).

  – Business and performance models become even more complex.
An autonomous capacity management model, which combines an business model with a performance model so as to increase the financial potential of the provider.
Part II: Environment

- Internet services
- Virtualization
- Service hosting platform
Environment

• Internet Services
  – Transactional services. (Brazil: banking system – 8 billion transactions in 2008)
  – *Application classes*.
  – Each class is subjected to different performance requirements, has different system characteristics, is executed by a different software component.
  – Assumption: application classes are independent.
Environment

Virtualization

– Abstract and conceal the physical (or real) resources
– Create new functionalities (e.g., virtual memory)
– Partition the physical resources.
– Allows transparent and flexible capacity expansion / contraction and QoS.
Environment

Virtualization

- Class 1
- Class 2
- Class N

Utilization guaranteed for each queue in the resource

• utilization = busy time / total time
Environment

Hosting platform

- VMs
  - Collection of virtual resources
  - Admission control
Environment

- **Definition**
  - **Capacity allocation decision**
    - Determination of fractions of server capacity each VM obtains from the physical IDC infra-structure.

\[ f_i(i = 1, \ldots, N) \]
Part III: Capacity Management Model

Pieces:
1. Autonomous capacity management framework.
2. SLA penalty/reward cost model.
3. Queuing-based performance model.
4. Optimization model
   - Connect the pieces of the model
1. Autonomous Framework

Control Interval
2. Cost Model

• Allows per-use service accounting.

Metric: performance achieved by applications.

• Two-level SLA contracts
  – Normal operation mode
  – Surge operation mode

• Customers pay for extra capacity only when needed.

• Penalty/Reward model.
2. Cost Model

- Throughput normal requirement

- Extra processing limit

- Response time requirement threshold

- Response time violation percent tolerance

\[ P(R_i > R_i^{SLA}) \leq \alpha_i \]
2. Cost Model

• Service level requirement
  – Throughput, subjected to a guarantee in the response time of the processed transactions.

\[ \mu = \{ X \mid P(R \geq R^{SLA}) \leq \alpha \} \]

  – Valid throughput.
2. Cost Model

\[ Y = \lambda - \mu \]
2. Cost Model

\[ Y = X^{NSLA} - \mu \]
2. Cost Model

\[ Z = \mu - X^{NSLA} \]
2. Cost Model

\[ Z = X^{SSLA} - X^{NSLA} \]
2. Cost Model

Penalty Magnitude

\[ Y = \min(X^{\text{NSLA}}, \lambda) - \mu \]

Reward Magnitude

\[ Z = \min(X^{\text{SSLA}}, \mu) - X^{\text{NSLA}} \]

*Extensível para múltiplos modos de operação.*

**Provider’s business objective**

Maximize the payoff resulting from the services’ execution.
3. Performance Model

- Map the application systems characteristics into capacity needs.
- Based on queuing models.
- Each VM is modeled as a service center.

- Assumptions
  - Arrivals follows Poisson.
  - Service time exponentially distributed.
3. Performance Model

- Average service time
- Queue utilization
- Physical capacity fraction
- Throughput

\[ \lambda \quad \mathbb{E}[S] \quad \rho \quad f \quad X \]
3. Performance Model

- **Queuing model results**
  - Average response time M/M/1
    
    \[ E[R_i] = \frac{E[S_i]}{f_i} \frac{1}{1 - \rho_i} \]

  - Response time variation M/M/1
    
    \[ Var[R_i] = \frac{(E[S_i]/f_i)^2}{(1 - \rho_i)^2} \]

  - Queue utilization
    
    \[ \rho_i = \frac{E[S_i]}{f_i} \times X_i \]
3. Performance Model

- Response time probability distribution approximations

- Markov

\[ P(R_i > R_i^{SLA}) \leq \frac{E[R_i]}{R_i^{SLA}} \]

- Chebyshev

\[ P(R_i > R^{SLA}) \leq \frac{\text{var}[R_i]}{(R^{SLA} - E[R_i])^2} \]

- Percentile (M/M/1)

\[ P(R_i > R_i^{SLA}) \leq e^{-R_i^{SLA} (f_i / E[S_i])(1-\rho_i)} \]
4. Optimization model

• Connect the parts of the capacity management model.

• Provides IDC configurations that maximize the business objective of the provider.

![Diagram showing the relationship between Model Parameters, Predicted Workload, Cost Model, Performance Model, Optimization Model, Capacity Allocation Decision, and Admission Control Parameters.]
4. Optimization model

\[ \max \sum_{i=1}^{N} -c_i Y_i + \pi_i Z_i P\{Z_i\} \]

\(\text{s.t.} \)

\(Y_i \geq \min(X_i^{NSLA}, \lambda_i^*) - X_i\) \quad (a)

\(Y_i \geq 0\) \quad (b)

\(Z_i = \delta_i(\min(X_i^{SSLA}, X_i) - X_i^{NSLA})\) \quad (c)

\(Z_i \geq 0, \quad \delta_i \in \{0, 1\}\) \quad (d)

\(\lambda_i^* = \lambda_i^{acc} + \lambda_i^{rej}\) \quad (e)

\(\lambda_i^{acc} = X_i\) \quad (f)

\(P(R_i \geq R_i^{SLA}) \leq \alpha_i\) \quad (g)

\(\rho_i = \frac{\lambda_i^{acc}}{\lambda_i^{sat}} \leq \nu_i\) \quad (h)

\(\sum_{i}^{N} f_i \leq 1\) \quad (i)

\(0 \leq f_i \leq 1, \quad \rho_i \geq 0\) \quad (j)

\(X_i, \lambda_i^{acc}, \lambda_i^{rej}, \lambda_i^{sat} \geq 0\) \quad (k)
Part IV: Experimental Analysis

- Simulation
- Model efficacy
- Precision degree achieved with approximations
- Behavior of applications
Experimental Analysis

Experimental setup

Application configuration

<table>
<thead>
<tr>
<th>$v_i$</th>
<th>$\alpha_i$</th>
<th>$X_i^{NSL A}$</th>
<th>$X_i^{SSL A}$</th>
<th>$R_i^{SL A}$</th>
<th>$c_i$</th>
<th>$\pi_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95</td>
<td>0.1</td>
<td>500</td>
<td>1000</td>
<td>0.1</td>
<td>1.0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Service demand: $E[S_i] = 10^{-3}$ sec
Experimental Analysis

Net result of the provider
Experimental Analysis

Queue size M/M/1

- Theoretical value: \( Q_i = \frac{\rho_i^2}{1-\rho_i} = \frac{(0.95)^2}{1-0.95} = 18.05 \)
Experimental Analysis

Response time M/M/1

- Requirement: \( P(R > 0.1) \leq 0.10 \)
Experimental Analysis

Penalty/Rewards M/M/1

(a) VM 1

(b) VM 2
Part V: Conclusions

- Summary
- Results
Conclusions

• Summary of proposals
  – A Penalty / Reward SLA-based cost model which allows for per-use service accounting.
  – A Queuing-based approximations to predict the performance of services.
  – An autonomous capacity management model for IDCs which combines the cost and performance models.
  – Capacity management approach which is able to increase the business potential of providers as well as maintaining service operations stable.
Conclusions

• Results
  – The autonomous capacity management model together with the approximations considered are able to maintain the application stability.
    • Stable queues
    • Response time requirement satisfaction
  – Markov’s approximation overestimates capacity needs.
  – Chebyshev percentile result in an equivalent degree of precision with M/M/1 model.

• Future Work: energy management!!!!!