Proactive Service Level Agreement Management in Cloud Computing Environments

Tong Lin

21.03.2011
Agenda

- Introduction and Motivation
  - Motivation
  - Use Cases
  - SLA Levels
  - Related Work
- Proactive SLA Management Architecture (pSLA)
  - Workflow
  - Conceptual Architecture
    - Master Service
    - Prediction Service
    - Reaction Service
- Example
  - UIMA and UIMA AS
  - Initialization Phase
  - Checkpoint Phase
Motivation

- Service Level Agreement (SLA)
  - Contracts between customers and service providers
  - Responsibility of both sides
    - Expected input data
    - Desired QoS target values
  - Problem Management: Penalty

- Proactive SLA Management Architecture (pSLA)
  - Web services in the cloud
  - Accurate SLA negotiation
  - Dynamic service execution

- Research Goal: Proactive Violation Prevention
  - Reduce penalty payments
  - Improve customer’s satisfaction
Use Cases

- **6 Waves**
  - Game developer on the Facebook platform
  - Based on Amazon Web Services

- **Advantages of Cloud Computing Platform**
  - “Infinite” computing resources
  - No up-front commitment
  - Only pay for used resources

- **Text Analytics**
  - Batch-oriented Application
    - Split into small tasks
    - Independent with each other
  - Benefit from the cloud
    - 1000 servers * 1 hour = 1 server * 1000 hours
SLA Levels

- Parties In the Business Model
  - Cloud Computing Provider
  - Service Provider
  - Customers

- SLA Levels
  - Technical SLA offered by IT Infrastructure Provider
    - Guaranteed Computing Utility
    - System Uptime
  - Business SLA offered by Service Provider
    - Expected QoS target values
    - Availability, Cost, Processing Time
Challenges and Related Work

- SLA Management: SLA Negotiation and SLA Enforcement

- SLA Negotiation: Accurate SLA Specifications
  - Historical performance data as background knowledge
  - Prediction about QoS target value and cloud configuration

<table>
<thead>
<tr>
<th>Typical Scenario:</th>
<th>Target:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms:</td>
<td>“Custom-made” algorithms</td>
</tr>
<tr>
<td>Performance:</td>
<td>Unaware</td>
</tr>
<tr>
<td></td>
<td>From service provider’s perspective</td>
</tr>
</tbody>
</table>

- SLA Enforcement: Violation Prevention
  - Related Work:
    - Detect possible violations at runtime using machine learning techniques
  - Target:
    - Proactively re-provision computing utility for violation prevention

- Organic Architecture: Complement Each Other

© SAP 2011 / Page 6
Agenda

- Introduction and Motivation
  - Motivation
  - Use Cases
  - SLA Levels
  - Related Work

- Proactive SLA Management Architecture (pSLA)
  - Workflow
  - Conceptual Architecture
    - Master Service
    - Prediction Service
    - Reaction Service

- Example
  - UIMA and UIMA AS
  - Initialization Phase
  - Checkpoint Phase
## Proactive Service Level Agreement Functionality

### Workflow

1. Customer’s request arrives
2. Query about historical performance data
3. Initial prediction
4. SLA negotiation and initialize processing
5. Monitor the cloud components
6. Periodic prediction at runtime
7. Proactive enforcement if violation detected
Proactive Service Level Agreement
Conceptual Architecture

- **Master Service**
  - SLA negotiation
  - Monitoring
  - Invokes other services
  - Keeps active

- **Prediction Service**
  - Runtime prediction
    - Machine learning algorithms
  - Dedicated instance
  - Destroyed when complete

- **Reaction Service**
  - SLA enforcement
  - Based on prediction results
Proactive Service Level Agreement

Master Service

- Manager
  - Initialization
    - SLA negotiation
  - Checkpoint
    - Check performance

- Feedback Database
  - Historical performance data
  - SQL-based database

- Monitor
  - Monitor performance data at runtime
  - Depend on cloud platforms
  - Amazon CloudWatch in AWS
Components

- Controller

- Predictor
  - Multiple
  - Implement prediction algorithms

- Evaluator
  - Mean Prediction Error
  - Prediction Error Standard Deviation
Proactive Service Level Agreement
Reaction Service

Components
- Enforcer
- Policy Pool
  - Various actions
  - Computing resources
  - Intensity
  - Reverse direction
- Notification Service
- Human Access Interface

Strategy
- Mild action first
- Powerful action if degradation
- Give up if unachievable
Agenda

- Introduction and Motivation
  - Motivation
  - Use Cases
  - SLA Levels
  - Related Work

- Proactive SLA Management Architecture (pSLA)
  - Workflow
  - Conceptual Architecture
    - Master Service
    - Prediction Service
    - Reaction Service

- Example
  - UIMA and UIMA AS
  - Initialization Phase
  - Checkpoint Phase
Scalable Text Analytics
UIMA and UIMA AS

- UIMA (Unstructured Information Management Architecture)
  - Apache Project based on JAVA
  - Processing unstructured information
  - CAS: common analysis structure
  - Analytics pipeline
    - Collection Reader
    - Analysis Engine (AE)
    - Flow Controller
    - CAS Consumer

- UIMA-AS (Asynchronous Scaleout of UIMA)
  - Client, Queue, Service
  - Main features
    - Parallel processing
    - Workload balancing

Text
- Email
- Audio
- Video

Indices
- DBs

UIMA

Unstructured Information
Structured Information
- Identify Semantic Entities
- Induce Structure
- Chats, Phone Calls, Email
- People, Places, Times

CPE

Collection Reader
- Aggregate Analysis Engine
- Date Time Annotator
- Room Number Annotator
- Meeting Annotator
- CAS Consumer

AS Service
CR AE CAS Consumer
AS Service
CR AE CAS Consumer
AS Service
CR AE CAS Consumer
Scalable Text Analytics
Cloud Configuration

- Scalable UIMF in AWS
  - Client
    - High I/O rate
  - Queue Manager
  - Service
    - CPU utility

![Diagram of cloud configuration with nodes and queues]
Scalable Text Analytics
Initialization Phase

- Receive Customer’s Request
  - Characteristics:
    - Processing Algorithm
    - Input file

- Query Feedback Database
  - Response
    - Historical performance data
    - Cloud configuration

- Initial Prediction
  - Instantiate a set of algorithms
  - Evaluate prediction results
  - Select the appropriate algorithm

- SLA Negotiation

- Setup Cloud
  - Suggested by background knowledge

- Start processing

<table>
<thead>
<tr>
<th>Request ID</th>
<th>Algorithm</th>
<th>Type</th>
<th>Max Size</th>
<th>Scaling Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>9e107d9d</td>
<td>ThingFinder</td>
<td>PDF</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>e4d909c2</td>
<td>ThingFinder</td>
<td>TXT</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>d41d8cd9</td>
<td>NLP</td>
<td>TXT</td>
<td>100</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Request ID</th>
<th>File Size</th>
<th>Processing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>9e107d9d</td>
<td>12450</td>
<td>6875</td>
</tr>
<tr>
<td>9e107d9d</td>
<td>5630</td>
<td>4253</td>
</tr>
<tr>
<td>9e107d9d</td>
<td>22652</td>
<td>14230</td>
</tr>
<tr>
<td>9e107d9d</td>
<td>9985</td>
<td>7012</td>
</tr>
<tr>
<td>e4d909c2</td>
<td>7524</td>
<td>3240</td>
</tr>
<tr>
<td>d41d8cd9</td>
<td>3258</td>
<td>522</td>
</tr>
</tbody>
</table>
Scalable Text Analytics
Prediction Algorithms

- Linear Regression
  - Workflow
    - Training data
    - Build Model: $Y = 0.0303 \times X - 223.5753$
    - Estimated target values
  - Advantages:
    - Simple, Efficiency
    - Use case: ThingFinder
    - Polynomial model

- Weka
  - Open source machine learning toolkit
  - Extensible
Scalable Text Analytics
Checkpoint Phase

- Checkpoint

- Runtime Prediction
  - Periodical execution
  - Selected algorithm
  - Check estimated target values

- If Violations Detected
  - Invoke reaction service
  - Re-provisioning computing resources
Conclusion

- Outcomes: Less Violations and Better Customer’s Satisfaction
  - Proactive SLA Management Architecture
    - Background Data
    - Runtime Prediction
    - Proactive Enforcement
- Evaluation of pSLA
  - Based on UIMA/UIMA AS
  - In Cloud Computing Environment

- Future Perspective
  - Other Cloud Platforms
  - Tests in other use cases
Thank you!
**Foundation**

**Service Level Agreement**

- **Service Level Agreement (SLA)**
  - Contracts between customers and service providers
  - Responsibility of both sides
    - Expected input data
    - Desired QoS target values
  - Problem Management: Penalty

- **Service-oriented Architecture (SOA)**
  - A flexible set of principles
    - Systems design, development, deployment and management
  - Provide reusable functions
  - Consumed by customers to compose business applications
  - Quality requirements described by SLA
Cloud Computing

- “Infinite” computing resources
- No up-front commitment
- Only pay for used resources

Cloud Computing Platform

- Amazon Web Services
- Google App Engine
- Windows Azure Platform

Components in Amazon Web Services (AWS)

- Amazon Elastic Compute Cloud (Amazon EC2)
  - Instance Types
  - Features
- Amazon CloudWatch
- AutoScaling
Challenges and Related Work

- SLA Management: SLA Negotiation and SLA Enforcement

- SLA Negotiation: Accurate SLA Specifications
  - Related work:
    - From service provider’s perspective
    - Be aware of exact performance of provided services
  - Current:
    - “Custom-made” algorithms
    - Historical performance data as background knowledge
    - Prediction about QoS target value and cloud configuration

- SLA Enforcement
  - Runtime Prediction
  - Proactive Enforcement

- Organic Architecture: Complement Each Other
Proactive Service Level Agreement
Prediction Service

Components

- Controller
- Predictor
  - Multiple
  - Implement prediction algorithms
- Evaluator
  - Mean Prediction Error
  - Prediction Error Standard Deviation

System Phases

- Initialization
  - Select appropriate prediction algorithm
- Checkpoint
  - Runtime Prediction based on monitoring data