Reducing Energy Consumption and Improving Bioinformatics with Clouds

Ivona Brandić
Distributed Systems Group
Institute of Information Systems, Vienna University of Technology
Austria

ivona@infosys.tuwien.ac.at
Current challenge of our society: BIG DATA!
• ICT consumes 10.5% of energy in Germany and rising, (Deutscher Bundestag, 2010)

• ICT produces 2% of worldwide CO₂ emissions and rising - equivalent to aviation industry, (Gartner, 2007)
ARE THE CLOUDS CANDIDATE (ENERGY EFFICIENT) SOLUTION FOR BIG DATA PROBLEMS?
Cloud Computing and Energy Efficiency
Cloud Computing and Energy Efficiency

TU Vienna

Barcelona, Spain
Cloud Computing and Energy Efficiency

Helsinki, Finland

TU Vienna

Barcelona, Spain

Web based  Scientific

Scientific  Business

Wattmeter
Tools and Instruments

Business

![Image of a snowy landscape](image1.png)

![Image of wind turbines and solar panels](image2.png)
Cloud Future: The 5th Utility

- provider lock in?
- disaster recovery?
- security?
- business-models?
- privacy?

... ?

- systematic testing?
- self-configuration?
- cost / break even models?

Source:
Clouds + Big Data = BIG MESS?
ORGANIZE YOUR CLOUDS!
... like some efficient systems in nature ...
Guaranteeing SLA

Service Level Agreement (SLA)

<table>
<thead>
<tr>
<th></th>
<th>≥</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Power</td>
<td>512</td>
<td>MIPS</td>
</tr>
<tr>
<td>Memory</td>
<td>1024</td>
<td>MB</td>
</tr>
<tr>
<td>Storage</td>
<td>1000</td>
<td>GB</td>
</tr>
<tr>
<td>Incoming Bandwidth</td>
<td>≥ 10</td>
<td>Mbit/s</td>
</tr>
<tr>
<td>Outgoing Bandwidth</td>
<td>≥ 20</td>
<td>Mbit/s</td>
</tr>
</tbody>
</table>

Self-adaptability of Cloud Computing Systems?

SaaS

PaaS

IaaS

Costly?
Challenges

Service Level Agreement (SLA)

<table>
<thead>
<tr>
<th>Service</th>
<th>Requirement</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Power</td>
<td>≥ 512</td>
<td>MIPS</td>
</tr>
<tr>
<td>Memory</td>
<td>≥ 1024</td>
<td>MB</td>
</tr>
<tr>
<td>Storage</td>
<td>≥ 1000</td>
<td>GB</td>
</tr>
<tr>
<td>Incoming Bandwidth</td>
<td>≥ 10</td>
<td>Mbit/s</td>
</tr>
<tr>
<td>Outgoing Bandwidth</td>
<td>≥ 20</td>
<td>Mbit/s</td>
</tr>
</tbody>
</table>

Cloud characteristics:
- dynamic
- on demand: computing as utility
- unforeseen load changes
- autonomic adaptation and (re-)provisioning of resources
- very scalable

Two conflicting goals:
1. Minimize SLA violations
2. Maximize energy efficiency

Achieve 1. and 2. by as few time- and energy consuming reallocation actions as possible
Speculate!

Speculative approach: May we allocate less resources than agreed, but more than actually utilized at the specific point in time – and not violate SLAs?

<table>
<thead>
<tr>
<th>Service Level Agreement (SLA)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Power</td>
<td>≥ 512</td>
<td>MIPS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>≥ 1024</td>
<td>MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage</td>
<td>≥ 1000</td>
<td>GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incoming Bandwidth</td>
<td>≥ 10</td>
<td>Mbit/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing Bandwidth</td>
<td>≥ 20</td>
<td>Mbit/s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
<th>What do we provide?</th>
<th>What does the consumer utilize?</th>
<th>What was agreed in the SLA?</th>
<th>Violation?</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 GB</td>
<td>400 GB</td>
<td>&gt;= 1000 GB</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>500 GB</td>
<td>510 GB</td>
<td>&gt;= 1000 GB</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>1000 GB</td>
<td>1010 GB</td>
<td>&gt;= 1000 GB</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>
How to structure actions?

- Escalation levels:
How to structure actions?

- Escalation levels:
  1. Change VM configuration

```
1. Change VM configuration
   500 GB ?
```
How to structure actions?

- Escalation levels:
  1. Change VM configuration
  2. Migrate applications from one VM to another.
How to structure actions?

- Escalation levels:
  1. Change VM configuration
  2. Migrate applications from one VM to another.
  3. Migrate one VM from one PM to another or create new VM on appropriate PM.

![Diagram](image)
How to structure actions?

- Escalation levels:
  1. Change VM configuration
  2. Migrate applications from one VM to another.
  3. Migrate one VM from one PM to another or create new VM on appropriate PM.
  4. Turn on/off PM.
How to structure actions?

- Escalation levels:
  1. Change VM configuration
  2. Migrate applications from one VM to another.
  3. Migrate one VM from one PM to another or create new VM on appropriate PM.
  4. Turn on/off PM.
  5. Outsource to other Cloud provider.
How to structure actions?

- Escalation levels:
  1. Change VM configuration
  2. Migrate applications from one VM to another.
  3. Migrate one VM from one PM to another or create new VM on appropriate PM.
  4. Turn on/off PM.
  5. Outsource to other Cloud provider.
  6. Do nothing!
How to structure actions?

- Escalation levels:
  1. Change one VM.
  2. Migrate one VM to another.
  3. Migrate the entire PM to a new VM.
  4. Turn on a new PM.
  5. Outsource the PM to a new provider.
  6. Do nothing!

Methodologies:
- Rules: IF – THEN
  - Trend analysis (up, down, oscillations, volatility, … )
- Case Based Reasoning (CBR)
- Situation Calculus
- …

Which one should we use?

Quality of recommended actions (decisions) = Violations vs. Wasted resources vs. Number of actions
How to evaluate KM techniques?

- Generic simulation engine
  - implements the MAPE-K loop
  - simulates monitoring and execution parts of it.

1. What do we provide?
2. What does the customer utilize?
3. What did we agree in the SLA?

Quality of recommended actions (decisions) = Violations vs wasted resources vs # actions
Energy efficiency vs. SLAs

- Knowledge Management
  - Iterative approach
  - keeps representation and characteristics of relevant objects in the Cloud
  - Threat Thresholds (TTs)
  - Over- and under-provisioning

Credits: M. Maurer
Rules

- Trend analysis *(up, down, oscillations, volatility, ...)*

- Rules using Drools
  - Rule increasing
    1. **IF** \( ut^r > TT_{high} \) AND \( ut^r_{predicted} > TT_{high} \)
    2. **THEN**
    3. **ELSE**
      4. Set \( pr^r \) to \( \frac{use^r}{tv(r)} \) if there are plenty of resources
      5. Set \( pr^r \) to \( \min\left(\frac{use^r}{tv(r)}, SLO^r \ast (1 + \epsilon/100)\right) \) if resources are becoming scarce

- Rule decreasing
  1. **IF** \( ut^r < TT_{low} \) AND \( ut^r_{predicted} < TT_{low} \)
  2. **THEN**
  3. **ELSE**
    4. Set \( pr^r \) to \( \max\left(\frac{use^r}{tv(r)}, \min Pr^r\right) \).
Define **cost function**

\[ c(p, w, c) = \sum_r P^r (p^r) + W^r (w^r) + A^r (a^r) \]

**Cost function - Adapt TTs, if cost increased for last \( k \) iterations**

- Resource \( r \) that had the most cost increasement over the last \( k \) iterations (look-back horizon)
- Determine most appropriate TT(s) to adapt? \( \rightarrow \) different options
- How much should TT be adapted?

\[ 0\% < TT_{low} < TT_{high} < 100\% \]

\[ TT_{low}^{r,t+1} = TT_{low}^{r,t} - \frac{TT_{low}^{r,t}}{\alpha} \]

\[ TT_{high}^{r,t+1} = TT_{high}^{r,t} - \frac{100 - TT_{high}^{r,t}}{\alpha} \]

where \( 1/\alpha < 1 \) distance from \( TT_{low} \) to 0 and \( TT_{high} \) to 100
Dynamic Adaptation of Threat Thresholds

Minimize SLA Violation (Option A)

1) Penalties
2) Wastage
3) Actions

Emphasize Energy Efficiency (Option B)

Very speculative (Option C)
Workload Volatility

- Define **workload volatility** (WV)

\[
\Phi^{r,s}(m^{r,s}, m^{r,s-1}) = \left( \frac{\max(m^{r,s}, r_{\text{min}})}{\max(m^{r,s-1}, r_{\text{min}})} - 1 \right) \times 100
\]

- Define WV classes LOW, ..., MEDIUM, ..., HIGH

- Determine most appropriate TT(s) to adapt? \(\rightarrow\) different options

<table>
<thead>
<tr>
<th>Volatility ((\Phi))</th>
<th>Option (E)</th>
<th>Option (F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW</td>
<td>(70%)</td>
<td>(90%)</td>
</tr>
<tr>
<td>MEDIUM</td>
<td>(45%)</td>
<td>(70%)</td>
</tr>
<tr>
<td>MEDIUM HIGH</td>
<td>(30%)</td>
<td>(60%)</td>
</tr>
<tr>
<td>HIGH</td>
<td>(20%)</td>
<td>(50%)</td>
</tr>
</tbody>
</table>

\(\leftarrow\) Very speculative

\(\leftarrow\) Minimize SLA Violations

\(\leftarrow\) Minimize SLA Violations even more

\(\leftarrow\) Be sure to really minimize SLA violations
## MEDIUM workload volatility

<table>
<thead>
<tr>
<th></th>
<th>penalties</th>
<th>utilization</th>
<th>actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>off, (50%,75%)</td>
<td>1.8</td>
<td>69.9</td>
<td>39.8</td>
</tr>
<tr>
<td>Ai, k=50</td>
<td>1.68</td>
<td>68.3</td>
<td>39.4</td>
</tr>
<tr>
<td>Ai, k=25</td>
<td>1.59</td>
<td>67.2</td>
<td>39.4</td>
</tr>
<tr>
<td>Ai, k=22</td>
<td>1.58</td>
<td>67.0</td>
<td>39.7</td>
</tr>
<tr>
<td>Ai, k=21</td>
<td>1.58</td>
<td>67.0</td>
<td>39.8</td>
</tr>
<tr>
<td>Ai, k=20</td>
<td>1.56</td>
<td>67.0</td>
<td>39.9</td>
</tr>
<tr>
<td>Ai, k=19</td>
<td>1.57</td>
<td>66.9</td>
<td>40.1</td>
</tr>
<tr>
<td>Ai, k=18</td>
<td>1.58</td>
<td>66.9</td>
<td>40.3</td>
</tr>
<tr>
<td>Ai, k=15</td>
<td>1.57</td>
<td>66.9</td>
<td>40.5</td>
</tr>
<tr>
<td>Ai, k=5</td>
<td>1.57</td>
<td>66.9</td>
<td>41.5</td>
</tr>
<tr>
<td>Bj, k=5</td>
<td>1.57</td>
<td>66.9</td>
<td>35</td>
</tr>
<tr>
<td>Cl, k=5</td>
<td>1.57</td>
<td>72.7</td>
<td>51.9</td>
</tr>
<tr>
<td>Di</td>
<td>1.57</td>
<td>69.4</td>
<td>37.8</td>
</tr>
<tr>
<td>El</td>
<td>1.57</td>
<td>66.0</td>
<td>36.7</td>
</tr>
<tr>
<td>Fj</td>
<td>1.57</td>
<td>67.0</td>
<td>40.1</td>
</tr>
</tbody>
</table>

### Costs

<table>
<thead>
<tr>
<th></th>
<th>costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>off, (50%,75%)</td>
<td>373</td>
</tr>
<tr>
<td>Ai, k=50</td>
<td>366</td>
</tr>
<tr>
<td>Ai, k=25</td>
<td>362.8</td>
</tr>
<tr>
<td>Ai, k=22</td>
<td>362.6</td>
</tr>
<tr>
<td>Ai, k=21</td>
<td>362.7</td>
</tr>
<tr>
<td>Ai, k=20</td>
<td>361.3</td>
</tr>
<tr>
<td>Ai, k=19</td>
<td>362.6</td>
</tr>
<tr>
<td>Ai, k=18</td>
<td>363</td>
</tr>
<tr>
<td>Ai, k=15</td>
<td>384</td>
</tr>
<tr>
<td>Ai, k=5</td>
<td>466</td>
</tr>
<tr>
<td>Bj, k=5</td>
<td>509</td>
</tr>
<tr>
<td>Cl, k=5</td>
<td>727</td>
</tr>
<tr>
<td>Di</td>
<td>347</td>
</tr>
<tr>
<td>El</td>
<td>362</td>
</tr>
<tr>
<td>Fj</td>
<td>362</td>
</tr>
</tbody>
</table>
HIGH workload volatility

![Graph showing workload volatility with bars for penalties, utilization, and actions. The graph includes data points for off, A, B, C, D, E, and F conditions with varying values for different scenarios.]
PUT YOUR (BIG) DATA ON THE CLOUD
Use Case: DNA Sequencing Workflows

- Data analysis in bioinformatics

DNA Sequence Sample

Applications for Next Generation Sequencing

Collaboration with BOKU Vienna (Group D. Kreil)
Monitoring

SLA aware and Resource Shared Monitoring

Service Level Agreement (SLA)

<table>
<thead>
<tr>
<th>Resource</th>
<th>Requirement</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Power</td>
<td>≥ 512</td>
<td>MIPS</td>
</tr>
<tr>
<td>Memory</td>
<td>≥ 1024</td>
<td>MB</td>
</tr>
<tr>
<td>Storage</td>
<td>≥ 1000</td>
<td>GB</td>
</tr>
<tr>
<td>Incoming Bandwidth</td>
<td>≥ 10</td>
<td>Mbit/s</td>
</tr>
<tr>
<td>Outgoing Bandwidth</td>
<td>≥ 20</td>
<td>Mbit/s</td>
</tr>
</tbody>
</table>
Monitoring Technique

- Monitors **low level resource metrics** (e.g., CPU, uptime, downtime, etc)
- Maps metrics to **Quality of Service parameters** (e.g., response time, throughput, etc)
- Monitoring agents

Availability formula:

\[
\text{Availability} = \left(1 - \frac{\text{downtime}}{\text{uptime}}\right) \times 100(\%)
\]
Mapping / Monitoring Framework

- Low level Metrics to High level SLA
- Comprises of three parts:
  - **Host monitor**
    - Low level metrics
  - **Communication Mechanism**
    - Message transmission
  - **Run-time monitor**
    - Metrics mapping
    - High level service SLA monitoring

<table>
<thead>
<tr>
<th>Resource Metric</th>
<th>SLA Parameter</th>
<th>Mapping Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uptime, Downtime</td>
<td>Availability (A)</td>
<td>[A = 1 - \frac{\text{downtime}}{\text{uptime}}]</td>
</tr>
<tr>
<td>Inbytes, outbytes, packetsize, avail.</td>
<td>Response Time (R)</td>
<td>[R = \text{Rin} + \text{Rout}(ms)]</td>
</tr>
<tr>
<td>Bandwithin, avail. Bandwithout</td>
<td></td>
<td>[\text{Rin} = \frac{\text{packetsize}}{\text{avail.BandwithIn} - \text{Inbytes}}]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[\text{Rout} = \frac{\text{packetsize}}{\text{avail.BandwithOut} - \text{Outbytes}}]</td>
</tr>
</tbody>
</table>
Lessons Learned: Time sharing of Resources

- Monitors **low level resource metrics** (e.g., CPU, uptime, downtime, etc)
- Maps metrics to **Quality of Service parameters** (e.g., response time, throughput, etc)
- Monitoring agents
- Model:
  - 1 PM : n VM
  - 1 VM : m Apps
  - 1 App -> 1 VM

- Model:
  - 1 PM : n VM
  - 1 VM : m Apps
  - X App -> 1 VM
Cloud Metric Classification (CMC)

Four (4) models:

- **Application based**
  - CPU time: Generic, Specific
  - Availability: Generic, Specific

- **Measurement based**
  - CPU time: Direct, Calculable
  - Availability: Direct, Calculable

- **Implementation based**
  - CPU time: Shared, Individual
  - Availability: Shared, Individual

- **Nature based**
  - CPU time: Quantity, Quality
  - Availability: Quantity, Quality

Credits: T. Mastelic
What metrics can we measure?

How do we calculate metrics?

How do we measure metrics?

How do we interpret metrics?

Cloud Management System

Scheduler

SLA violation detection

KB

Resource manager

report

monitor

Application

Resources

manage resources

M4Cloud model
Optimizing Workflow Executions

Cloud resource management Techniques

- Monitoring Framework
- Knowledge Management

Cloud Infrastructure

- Million reads
- Mapping to genome
- Monitoring Agent

- Aligned reads
- Not-aligned reads

- Splitting to sub-reads
- Monitoring Agent

- Sub-reads Set 1
- Sub-reads Set 2
- Sub-reads Set n

- Searching for splice-junctions
- Monitoring Agent

- Reads aligned to the reference

Credits:
Vincent Emeakaroha
Resource Allocation Results

- Three scenarios
  1. Static configuration
  2. Speculative approach
  3. "Oracle Approach" - Peak provisioning - we know everything just before it happens
Bioinformatics Workflow

The graph shows the performance of different configurations, with each bar representing a different set of parameters. The y-axis represents the performance metrics, while the x-axis lists various configurations.

- **Off, [50%, 75%]**
- **A), k=2**
- **A), k=5**
- **A), k=15**
- **A), k_st=10, k_cpu=5, k_mem=2**
- **A), k_st=10, k_cpu=2, k_mem=5**
- **E)**
- **F)**

The bars are color-coded to indicate different metrics:
- Green: Utilization
- Orange: Actions
- Yellow: Penalties

The specific values for each configuration are shown above the bars.
Future / Current Work: CO$_2$ trading

certified emission reductions (CERs)
Current / Future Work: Costs of moving to Clouds?

Cooperation with physics department (TU Vienna): Application for yocto second photon emission from quark gluon plasmas
## Current / Future Work

### Setup steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Physical</th>
<th>Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Platform setup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Installing an operating system</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>- Installing drivers</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>- Installing a hypervisor</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>- Installing CMS</td>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>2. Environment setup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Installing libraries</td>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>- Installing application</td>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>3. Run</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Deploy</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>- Run</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Environment setup

Setup steps:

1. Platform setup
2. Environment setup
3. Run

Complexity

Number of nodes

Three scenarios:

- Initial setup
- Next setup
- Migration
Summary and Current / Future Work

- **Work done**
  - Self-adaptable Cloud Infrastructures
    - Generic Monitoring
    - Knowledge Management
    - Applied to real world scenarios (Bioinformatics)

- **Future Work:**
  - Systematic Approaches for the Development of Cloud Infrastructures
    - Software Engineering, Model Checking, Verification, Formal Models, ....
  - Big Data & Energy Efficiency
    - Data is the “new petrol” in Europe
    - Management of CO₂ footprints
Thanks to my colleagues

Soodeh Farokhi

Michael Maurer

Ivan Bresković

Andres Garcia Garcia

Dražen Lučanin

Toni Mastelić

Vincent Emeakaroha
Questions & Contact information

Ivona Brandic
Distributed Systems Group
Information Systems Institute
Vienna University of Technology
Austria
email: ivona@infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/ivona