Cloud computing: Impact on software engineering research and practice

Thomas Østerlie
Overview

• Presentation's main message
  – **Sobriety**: Cloud computing is a continuation of distributed models of software development and delivery
  – **Opportunities**: Cloud computing offers new possibilities for software engineering researchers to study multilateral software development

• Outline
  – Part I: Interpretation of the lecture topic
  – Part II: Getting to grips with cloud computing
  – Part III: Implications of cloud computing on software engineering research and practice
Part I: Interpretation of the lecture topic
Outline

• Part I: Interpretation of the lecture topic
  – Defining cloud computing
  – Cloud computing in the software engineering literature
  – The impact debate
  – Interpretation of the topic
• Part II: Getting to grips with cloud computing
• Part III: Implications of cloud computing on software engineering research and practice
Defining cloud computing

"A new paradigm"

"An overall confusing technological picture"

"Simply a hype and a buzzword"
Cloud computing in software engineering

<table>
<thead>
<tr>
<th>Type</th>
<th>Title</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research journals</td>
<td>IEEE Software</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>IEEE Transactions on Software Engineering</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Transactions on Software Engineering Methods</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Journal of Systems and Software</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Information Software and Technology</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Software Practice and Experience</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Confer.</td>
<td>Int. Conference on Software Engineering</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Int. Conf. on Software Maintenance</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Int. Symp. on Empirical Software Engineering</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mags</td>
<td>Communications of the ACM</td>
<td>0</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Computer</td>
<td>0</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

*Results from queries with the string "cloud computing" in full paper text*
Debating impact
Interpretation of the topic

• What is cloud computing?

• What could the impact of cloud computing be on software engineering research and practice?
Part II: Getting to grips with cloud computing
Outline

• Part I: Interpretation of the lecture topic
• Part II: Getting to grips with cloud computing
  – Historical development of cloud computing
  – Promises of cloud computing
  – Cloud computing models
  – Cloud computing compared
• Part III: Implications of cloud computing on software engineering research and practice
From mainframes to clouds

- Enduring processes in the development of computing*
  - Emergent challenges
  - Technological opportunities
- Core driver of computing development: Realizing cost savings
  - Expanding commodification
  - Increased abstraction

* Interpretive framework draws on Friedman and Cornford (1989)
Expanding commodification

1. Time-sharing (1960s)
   - Personal computers
   - Shrink-wrap software (late 1970s)

2. Mass computerization (1980s and 1990s)
   - Internetworking, hosting (late 1990s)

www.ntnu.no
Increasing abstraction

- Time-sharing (1960s)
- Personal computers, shrink-wrap software (late 1970s)
- Mass computerization (1980s and 1990s)
- Internetworking, hosting (late 1990s)
- 3G languages (1970s)
- General purpose libraries (1970s and 1980s)
- Software components (late 1980s)
- Software services (early 2000s)
Convergence into clouds

- **1960s**: Time-sharing
- **late 1970s**: Personal computers, shrink-wrap software
- **1980s and 1990s**: Mass computerization
- **late 1990s**: Internetworking, hosting
- **1970s**: 3G languages
- **1970s and 1980s**: General purpose libraries
- **late 1980s**: Software components
- **early 2000s**: Software services
- **late 2000s**: Cloud computing

www.ntnu.no
Promises of cloud computing

• Three promises for cost savings
  – Reduced costs of operations and delivery
  – Effective utilization of computing resources
  – Reduced development costs

• Realizing the promised cost savings
  – Models for software delivery
  – Distributed hardware resources
Conceptual level model for cloud computing

Layers

Stakeholders

Cloud service user

Cloud service provider

Infrastructure provider

*Adapted from Vacquero et al. (2009), Leavitt (2009), and Erdogmus (2009)
Three models for separating ownership from use

<table>
<thead>
<tr>
<th>Model</th>
<th>Cloud service user</th>
<th>Cloud service provider</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software-as-a-service</td>
<td>End-user</td>
<td>Turnkey application</td>
<td>Google Docs</td>
</tr>
<tr>
<td></td>
<td>Software developer</td>
<td>Web service</td>
<td>Amazon S3</td>
</tr>
<tr>
<td>Platform-as-a-service</td>
<td>Software developer</td>
<td>Application development and deployment platform</td>
<td>Force.com</td>
</tr>
<tr>
<td>Infrastructure-as-a-service</td>
<td>End-user organization</td>
<td>Computing infrastructure</td>
<td>ElasticHost</td>
</tr>
</tbody>
</table>
End-user Software-as-a-Service

- **Cloud service user**: End user (organization)
- **Cloud service provider**: Turnkey software application
- **Promises**
  - Subscription fee replaces cost of operations and delivery
  - Utilization of computing resources through pay-per-use model
Software-as-a-Service

- **Cloud service user:** Software developer (organization)
- **Cloud service provider:** Reusable software services
- **Promises**
  - Reduced development time through use of high-level service
  - Maximum utilization of resources: pay for what you use
  - Reduced operations costs
Platform-as-a-Service

- **Cloud service user:** Software development organizations

- **Cloud service provider:** Virtual platform for application development and deployment

- **Promises**
  - Reducing cost of operations and delivery
  - Increased development productivity
Infrastructure-as-a-Service

- **Cloud service user:** Computing organization
- **Cloud service provider:** Offers scalable virtual machines
- **Promises**
  - Utilization of computing resources
  - Reduced costs of operations (of hardware)
Cloud computing compared

Cloud computing

Grid computing

Service-oriented architecture

Web 2.0
Grid computing
Grid vs. cloud computing

• Similarities
  – Deliver computing power through distributed networks of computers

• Differences
  – Grids typically batch-oriented
  – Cloud computing business flows and functionality

• Proposed convergence of the two
Service-oriented architecture
Services in clouds vs. SOA

• Similarities
  – Most similar to software-as-a-service model
  – Emphasizes services over products
  – Web services seem the best current technology for realizing both

• Differences
  – Cloud computing specifically over the Internet
  – SOA also intranet within single or among several organizations
  – Cloud computing mainly for developing new applications
  – Legacy applications important for SOA
Web 2.0
Web 2.0 in the clouds

- Focus on interactive web applications
- Overlaps with end-user oriented software-as-a-service
- Software-as-service enabling for rapid development of Web 2.0 applications
Summary

• Cloud computing: Commodification of computing resources

• Conceptual-level model for cost reduction
  – Reduced costs of operations and delivery
  – Effective utilization of computing resources
  – Reduced development costs

• Three cloud computing models
  – Software-as-a-Service
  – Platform-as-a-Service
  – Infrastructure-as-a-Service
Part III: Impact of cloud computing on software engineering research and practice
Outline

• Part I: Interpretation of the lecture topic
• Part II: Getting to grips with cloud computing
• Part III: Implications of cloud computing on software engineering research and practice
  – Assessing possible impact
  – Multilateral software development
  – Impact on software engineering practice
  – Impact on software engineering research
Cloud computing mainly an industry phenomenon

<table>
<thead>
<tr>
<th>Type</th>
<th>Title</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research journals</td>
<td>IEEE Software</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>IEEE Transactions on Software Engineering</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Transactions on Software Engineering Methods</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Journal of Systems and Software</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Information Software and Technology</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Software Practice and Experience</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Confer.</td>
<td>Int. Conference on Software Engineering</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Int. Conf. on Software Maintenance</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Int. Symp. on Empirical Software Engineering</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mags</td>
<td>Communications of the ACM</td>
<td>0</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Computer</td>
<td>0</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

*Results from queries with the string "cloud computing" in full paper text
Assessing possible impact

• Software engineering: Research-informed software development
  – Application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software
  – Research: The study of these applications
  – Practice: The application of these approaches

• Focus
  – Lifecycle issues over technology issues
  – Research perspective
# Multilateral software development

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Application software development</th>
<th>Multilateral software development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software composition</td>
<td>Coherent set of software modules</td>
<td>Interoperable third-party components</td>
</tr>
<tr>
<td>Source code</td>
<td>Full source available</td>
<td>No source code for third-party components</td>
</tr>
<tr>
<td>Execution model</td>
<td>Single computer</td>
<td>Often distributed between multiple computers</td>
</tr>
<tr>
<td>Ownership and control</td>
<td>Single team or organization</td>
<td>Distributed between multiple organizations</td>
</tr>
</tbody>
</table>

### Challenges to practice

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Cloud computing</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software composition</td>
<td>Interoperable third-party services</td>
<td>• Service selection</td>
</tr>
<tr>
<td>Source code</td>
<td>No source code for third-party components</td>
<td>• System comprehension</td>
</tr>
<tr>
<td>Execution model</td>
<td>Often distributed between multiple computers</td>
<td>• State inspection and debugging</td>
</tr>
<tr>
<td>Ownership and control</td>
<td>Distributed between multiple organizations</td>
<td>• Separation of ownership and possession from use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Interaction effects among services from different providers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Coping with evolution of third-party services</td>
</tr>
</tbody>
</table>
Implications for research

• Theory generation
  – Existing theory aimed at application software development
  – Generate theory for multilateral software development
  – Cloud computing one of many venues for this

• Research approaches
  – Experimental studies reiterates old theoretical assumptions
  – Theory generation requires practice studies
  – Importance of bringing the socio-technical into account

• Research challenges
  – Provider perspective remains underdeveloped
  – Depart with the modernistic images of ‘control’
  – Embrace contingent forms of control and opportunism
  – Manage increased complexity leniently
Summary

• Cloud computing is a continuation of distributed models of software development and delivery

• Cloud computing offers new possibilities for software engineering researchers to study multilateral software development