Cloud Computing
A simple taxonomy

Cloud

On-premises

Software as a Service (SaaS)
Attached Services
Cloud Platforms

Application
Extra Functions
Platform

Users

Browser/Client
Application
Application

Developers
Comparing Cloud Platforms
Cloud Platforms
Defining categories

- **In the beginning:**
  - Infrastructure as a Service (IaaS)
    Example: Amazon Elastic Compute Cloud (EC2)
  - Platform as a Service (PaaS)
    Example: Salesforce.com Force.com

- **Today:**
  - The cloud platform world is more complex
  - The IaaS/PaaS dichotomy no longer makes much sense
## Levels of Functionality

### Which is IaaS? Which is PaaS?

<table>
<thead>
<tr>
<th>Service</th>
<th>Can run application code</th>
<th>Application runs in a visible VM</th>
<th>Have administrative access to VM</th>
<th>Can create your own OS installation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Web Services EC2</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Mosso Cloud Servers</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Windows Azure CTP</td>
<td>✗</td>
<td>✗</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google AppEngine</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salesforce.com Force.com</td>
<td>✗</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scenarios
An alternative approach

- A more meaningful way to compare cloud platforms today is by the application scenarios they support

- I’ve chosen five representative application scenarios
  - There are plenty of other possibilities
Cloud Platform Scenarios (1)

- Running an on-premises app unchanged in the cloud
  - Example: An enterprise moving an on-premises line-of-business app to the cloud

- Creating a moderately scalable Web app
  - Example: An enterprise building a custom application used by its employees
  - Example: An ISV creating a moderately popular consumer application
Creating a very scalable Web application
  – Example: An ISV creating the next Facebook
  – Example: A large bank creating a customer-facing app

Creating a parallel processing application
  – Example: An enterprise using MapReduce to process large amounts of data

Creating a very scalable Web application with background processing
  – Example: An ISV creating the next YouTube
Cloud Platform Scenarios
Running an on-premises application unchanged

Cloud

On-premises

Application

Virtual Machine(s)

Operating System

DBMS

Operating System

DBMS

Physical or Virtual Machine(s)
Cloud Platform Scenarios
Creating a moderately scalable Web application

Web Application

Scale-out Web App Platform¹
Relational Storage²

Cloud

On-premises

Users

¹ Such as virtual machines running replicated Web servers
² Such as virtual machines running a relational DBMS
Cloud Platform Scenarios
Creating a very scalable Web application

Cloud Platform Scenarios
Creating a very scalable Web application

Cloud

Web Application

Scale-out Web App Platform

Scale-out Storage

On-premises

Users
Examining Scale-Out Storage

An aside

- **Challenges:**
  - An unfamiliar structure
    - Moving relational data to it takes work
    - Supporting services are scarce, e.g., reporting
  - No SQL
  - No schema
  - Some platform lock-in
    - Because each vendor does it differently

- **Benefits**
  - Massive scalability
Cloud Platform Scenarios
Creating a parallel processing application

Parallel Processing Application

Cloud

Scale-out Batch App Platform

Blob Storage

On-premises

User
Cloud Platform Scenarios
A scalable Web app with background processing

Scalable Web Application with Background Processing

Cloud

Scale-out Storage
Scale-out Web Platform
Queues
Scale-out Batch Platform
Blob Storage

On-premises

Users
## Scenarios vs. Platform Capabilities

<table>
<thead>
<tr>
<th>Run On-Premises Apps</th>
<th>Create Moderately Scalable Web Apps</th>
<th>Create Very Scalable Web Apps</th>
<th>Create Parallel Processing Apps</th>
<th>Create Very Scalable Web Apps with Background Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VM with standard OS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scale-out web app platform</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scale-out batch app platform</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Relational storage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scale-out storage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Blob storage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Queues</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
GoGrid, Mosso, Flexiscale, Others

Typical scenarios

<table>
<thead>
<tr>
<th>Run On-Premises Apps</th>
<th>Create Moderately Scalable Web Apps</th>
<th>Create Very Scalable Web Apps</th>
<th>Create Parallel Processing Apps</th>
<th>Create Very Scalable Web Apps with Background Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VM with standard OS</strong></td>
<td><strong>VMs</strong></td>
<td><strong>VMs</strong></td>
<td><strong>VMs</strong></td>
<td><strong>VMs</strong></td>
</tr>
<tr>
<td><strong>Scale-out web app platform</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scale-out batch app platform</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Relational storage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scale-out storage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Blob storage</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Queues</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Amazon Web Services

## Typical scenarios

<table>
<thead>
<tr>
<th>VM with standard OS</th>
<th>Scale-out web app platform</th>
<th>Scale-out batch app platform</th>
<th>Relational storage</th>
<th>Scale-out storage</th>
<th>Blob storage</th>
<th>Queues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run On-Premises Apps</td>
<td>Create Moderately Scalable Web Apps</td>
<td>Create Very Scalable Web Apps</td>
<td>Create Parallel Processing Apps</td>
<td>Create Very Scalable Web Apps with Background Processing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EC2 VMs</td>
<td>EC2 VMs</td>
<td>EC2 VMs</td>
<td>EC2 VMs</td>
<td>EC2 VMs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EC2 VMs (w/RDBMS)</td>
<td>EC2 VMs (w/RDBMS)</td>
<td>SimpleDB</td>
<td>Simple Storage Service (S3)</td>
<td>Simple Queue Service (SQS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimpleDB</td>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Windows Azure CTP
### Typical scenarios

<table>
<thead>
<tr>
<th>Run On-Premises Apps</th>
<th>Create Moderately Scalable Web Apps</th>
<th>Create Very Scalable Web Apps</th>
<th>Create Parallel Processing Apps</th>
<th>Create Very Scalable Web Apps with Background Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VM with standard OS</strong></td>
<td><strong>Web role</strong></td>
<td><strong>Web role</strong></td>
<td><strong>Worker role</strong></td>
<td><strong>Web role</strong></td>
</tr>
<tr>
<td><strong>Scale-out web app platform</strong></td>
<td><strong>SQL Data Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scale-out batch app platform</strong></td>
<td></td>
<td></td>
<td><strong>Worker role</strong></td>
<td><strong>Worker role</strong></td>
</tr>
<tr>
<td><strong>Relational storage</strong></td>
<td></td>
<td><strong>Tables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scale-out storage</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>Tables</strong></td>
</tr>
<tr>
<td><strong>Blob storage</strong></td>
<td></td>
<td></td>
<td><strong>Blobs</strong></td>
<td><strong>Blobs</strong></td>
</tr>
<tr>
<td><strong>Queues</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>Queues</strong></td>
</tr>
</tbody>
</table>
SQL Data Services (SDS)

An aside

- Provides relational storage in the cloud
  - A distinct service (not part of Windows Azure)
  - Built on Microsoft SQL Server technology

- Can be used by applications running in the cloud or on-premises
## Google AppEngine

### Typical scenarios

<table>
<thead>
<tr>
<th>VM with standard OS</th>
<th>Run On-Premises Apps</th>
<th>Create Moderately Scalable Web Apps</th>
<th>Create Very Scalable Web Apps</th>
<th>Create Parallel Processing Apps</th>
<th>Create Very Scalable Web Apps with Background Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale-out web app platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale-out batch app platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relational storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale-out storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blob storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queues</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Run On-Premises Apps**
  - **Create Moderately Scalable Web Apps**
    - **Create Very Scalable Web Apps**
    - **Create Parallel Processing Apps**
    - **Create Very Scalable Web Apps with Background Processing**

- **Java/Python runtime**
- **Datastore**
## Salesforce.com Force.com

### Typical scenarios

<table>
<thead>
<tr>
<th>Run On-Premises Apps</th>
<th>Create Moderately Scalable Web Apps</th>
<th>Create Very Scalable Web Apps</th>
<th>Create Parallel Processing Apps</th>
<th>Create Very Scalable Web Apps with Background Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM with standard OS</td>
<td></td>
<td><em>Force.com runtime</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale-out web app platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale-out batch app platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relational storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale-out storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blob storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queues</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparing AppEngine and Force.com

Two different approaches

- AppEngine provides a Java or Python environment for building Web apps
  - Apps are built by developers

- Force.com provides a platform for creating data-oriented business applications with browser interfaces
  - Apps can be built by less technical people
  - Apps can also be built by developers
    - Using Apex Code
# Comparing Cloud Platforms

Summarizing typical scenarios

<table>
<thead>
<tr>
<th></th>
<th>Run On-Premises Apps</th>
<th>Create Moderately Scalable Web Apps</th>
<th>Create Very Scalable Web Apps</th>
<th>Create Parallel Processing Apps</th>
<th>Create Very Scalable Web Apps with Background Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoGrid, Mosso, Flexiscale, etc.</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Amazon Web Services</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows Azure CTP</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Google AppEngine</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Salesforce.com Force.com</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Cloud Platforms Today:
Other Topics
Categorizing the Scenarios
Who cares most about each one?

- **Enterprises**
  - Run an on-premises app unchanged
  - Others of interest:
    - Create moderately scalable Web apps
    - Create parallel processing apps

- **ISVs**
  - Create very scalable Web apps
  - Create very scalable Web apps with background processing
ISV apps and custom enterprise apps have similar scalability/reliability requirements.
The World with SaaS Applications

- ISV apps and custom enterprise apps have different scalability/reliability requirements
Cloud Platform Bifurcation?

- **Enterprises**
  - Want cloud platforms to lower their costs
  - Might want to be able to move apps and data easily between the cloud and on-premises

- **ISVs**
  - Want cloud platforms to provide scalability and reliability

- These different requirements are met by different cloud platform capabilities
Pricing
Some examples

- Amazon Web Services
  - Pay for your data storage and transfer
  - Pay per-hour for each of your running VMs

- Microsoft Windows Azure
  - Not yet announced

- Google AppEngine
  - Pay for your data storage and transfer
  - Pay for the CPU time your app uses

- Salesforce.com Force.com
  - Pay per user or based on page views
Pricing
The importance of reservations

- EC2 now offers reservations
  - This lowers prices for longer-term commitments

- Reservations help a cloud platform vendor do better capacity planning
  - And thus make more money

- Expect the idea to spread
Questions:

– How easy is it to expose my cloud platform app’s services to other apps?
– How easy is it to access my cloud platform app’s data from other apps?

The situation today is quite good:

– Cloud platform apps can expose and consume Web services
– Cloud platform app data can be accessed by external apps
Cloud Platform Lock-in
Portability

- Questions:
  - How easy is it to move my cloud platform code back on-premises or to another cloud platform?
  - How easy is it to move my cloud platform data back on-premises or to another cloud platform?

- The situation today varies with different cloud platforms
  - And different options within those platforms
Cloud Platform Lock-in
Estimating relative portability for code and data

Work required to port code to on-premises platform

- Windows Azure with SDS
- EC2 with standard DBMS and SQS
- EC2/GoGrid/etc. with standard DBMS
- EC2 with standard DBMS and SQS

Work required to port data to on-premises platform

- Windows Azure with Tables
- AppEngine
- Force.com
- Windows Azure with SimpleDB
- EC2 with SimpleDB
Conclusion

- Today’s cloud platform choices vary considerably
  - It’s more than just IaaS and PaaS

- Making the right decision requires:
  - Knowing what your goals are
  - Understanding what each option offers

- It’s a great time to build applications!
David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technology. David has been the keynote speaker for many events and conferences on five continents, and his seminars have been attended by tens of thousands of IT decision makers, architects, and developers in forty countries. His books have been published in a dozen languages and used regularly in courses at MIT, ETH Zurich, and other universities. In his consulting practice, he has helped clients such as Hewlett-Packard, IBM, Microsoft, Stanford University, and Target Corporation adopt new technologies, market new products, train their sales staffs, and create business plans. Earlier in his career, David wrote networking software, chaired a U.S. national standards working group, and played keyboards with the Peabody-award-winning Children’s Radio Theater. He holds a B.S. in Economics and an M.S. in Computer Science, both from the University of Wisconsin-Madison.
Copyright © 2009 David Chappell
Chappell & Associates

www.davidchappell.com