Programming Models for the Cloud

Marty Humphrey
Associate Professor
Department of Computer Science
University of Virginia
Cloud Programming: It’s Oddly Interconnected

- Session 1: Cloud Architectures
- Session 3: Data Portability, Consistency, and Management
- Session 5: Fault Masking in the Cloud
- Session 7: Cloud Debugging, Diagnosis, Certification and Update
- Session 9: Cloud Inter-operability and Standardization
Cloud Expo: Blog Feed Post

Cloud Computing Does Not Require Change in Programming Model – Microsoft

Microsoft’s cloud strategy assumes that they are going to retain the existing programming model stated that Microsoft’s cloud strategy assumes that they are going to retain the existing programming model for cloud. That is, programmers can develop their application without bothering about where it is going to be deployed.

This is how the logic goes:

1. When mini computers came in, IBM did not / could not retain the mainframe programming model for mini and hence lost the mini race to DEC
2. When PCs came DEC suffered in the same way
3. The movement from character interface to GUI saw many companies (Ashton-Tate, Word Perfect) losing out
4. When network based computing transitioned from file server to client

worry about the separation of GUI and business logic in order to support N-Tier applications. Publishing to IIS does NOT scale. For me, the beauty of the datawindow is that even an intermediate developer like me can quickly create robust applications with minimal code. It is great that you outline ways to...
Not Discussed (Directly)

• Virtual Clusters (John?)
• MapReduce/Hadoop/Dryad (Judy?)
Assumptions

• $$/time$$ will decrease
• $$/byte$$ will decrease

• $$/line-of-code$$ will increase

• Humans’ brains will stay the same size

• Speed of light will remain the same
Programming Model: Challenge

• How do we write a (cloud) application that provides the right info at the right time (to you) on your device of choice?

• How does my (cloud) application get exactly what it wants while still respecting the black-box nature of the cloud?
Programming Model: Issues

• Is it “new” or can we effectively adapt?
  – E.g., DFS: a step forward or backward?

• Scale or no scale?
  – Scalability of the cloud != scalability of my app

• *Should it hide or expose* the underlying characteristics of the platform?
Our experience: MODISAzure
Our Lessons from MODISAzure

• Scaling (“de-scaling”) compute is slow
• IaaS is too low-level, PaaS is too high-level
• Dev cycle was awful
• Granularity of computation (VMs) did not trivially match granularity of application parallelism
• Blobs, tables, queues – too low-level?
NSF Should Fund...

• Relationship between “programming model” and SLA/economics
• Programming model of a multi-cloud application
  – Same as a “uni-cloud”?  
• Adapt to the platform ...
  – ... or have the platform adapt to the app?

• *Activities that industry will not “naturally” solve*