Google Data Centers (2009)

Google Data Centers (2009)

Cost of Traditional Data Centers

- 11.8 million servers in data centers
- Servers are used at only 15% of their capacity
- 800 billion dollars spent yearly on purchasing and maintaining enterprise software
- 80% of enterprise software expenditure is on installation and maintenance of software
- Data centers typically consume up to 100 times more per square foot than a typical office building
- Average power consumption per server quadrupled from 2001 to 2006.
- Number of servers doubled from 2001 to 2006
Energy Conservation and Data Centers

- Standard 9000 square foot costs $21.3 million to build with $1 million in electricity costs/year
- Data centers consume 1.5% of US electricity (EPA)
  - 0.6% worldwide in 2000 and 1% in 2005
- Green technologies can reduce energy costs by 50%
- IT produces 2% of global carbon dioxide emissions
Cloud Economics

- Estimates vary widely on possible cost savings
- “If you move your data centre to a cloud provider, it will cost a tenth of the cost.” – Brian Gammage, Gartner Fellow
- Use of cloud applications can reduce costs from 50% to 90% - CTO of Washington D.C.
- IT resource subscription pilot saw 28% cost savings - Alchemy Plus cloud (backing from Microsoft)
- Preferred Hotel
  - Traditional: $210k server refresh and $10k/month
  - Cloud: $10k implementation and $16k/month
Hype Cycle for Data Management (Gartner, 2010)
Hype Cycle for CC (Gartner, 2010)

Over the course of the next five years, enterprises will spend $112 billion cumulatively on SaaS, PaaS and IaaS combined.
Growth of Cloud-based Apps

Source: Gartner (November 2010)
Growth of Compute Services

Source: Gartner (November 2010)
Pros of CC

- Lower-cost computers for end users
- Improved performance on users’ PC
- Lower IT infrastructure and software costs
- Fewer maintenance issues
- Instance software updates
- Unlimited storage capacity
- Increased data safety
- Easier group collaboration
- Universal access to data/documents
Cons of CC

- Requires a constant internet connection
- Doesn’t work well with low-speed connections
- Can be slower than using desktop software
- Features might be more limited
- Stored data might not be secure
- If the cloud loses your data, bing problem
Who Benefits from CC?

- Collaborators
- Road warriors
- Cost-conscious users
- Cost-conscious IT departments
- Users with increasing needs
Who Shouldn’t be Using CC?

- The Internet-impaired
- Offline workers
- The security conscious
- Anyone married to existing applications
  - Eg, MS Office (now one can use MS Office in CC too)
CC-Related Technologies

- Thin Client
- NIC: New Internet Computer
- Autonomic Computing
- Grid Computing
- Utility Computing
- Web 2.0
- Peer-2-Peer Computing
- Web Service
Thin Client

- Early Eg: PDP 11/34 mini computer serving multiple “dumb” terminals (as thin clients)
- Term coined by T. Negris @ Oracle in 1993
- User computers rely heavily on server machines to do actual tasks
- Cheap/simple client software/hardware
- ↔ Fat Client

IBM EXX Thin Client
NIC: New Internet Computer

- Larry Ellison (Oracle) championed the idea in 2000
- $199 machine boots from CD-Rom (Linux)
- 64MB RAM
- No HDD, just 4MB Flash memory
- Can’t install software
- Connect to hosted apps and files in the central supercomputer via Internet
- Selected as one of worst 10 PCs of all time
Autonomic Computing

- Initiated by IBM in 2001
- Computer systems capable of self-management
- IBM’s 4 properties: self-healing, self-configuration, self-optimization, self-protection
- Decrease needs for human administrators to perform lower level tasks
Utility Computing

- “Computing may someday be organized as a public utility” – John McCarthy, MIT Centennial in 1961
- “Packaging of computing resources, such as computation and storage, as a metered service similar to a traditional public utility, such as electricity”
  - A break in the clouds: towards a cloud definition, ACM SIGCOMM, 2009
- Hugh computational and storage capabilities available from utilities
- Metered billing (Pay-As-You-Go)
Grid Computing

- Distributed parallel processing across network
- Combination of computer resources from multiple administrative domains to reach a common goal
- Loosely coupled, heterogeneous, and geographically dispersed
- Coined by Ian Foster in 1990s
  - As a metaphor for making computer power as easy to access as an electric “power grid”
- Globus toolkit
Grid Computing Example

- SETI@Home Project
- CPU Scavenging
- Volunteer Computing
- Initially released in 1999

- Two initial goals:
  - To do useful scientific work by supporting an observational analysis to detect intelligent life outside Earth → Failed
  - To prove the viability and practicality of the “volunteer computing” concept → Succeeded
Web 2.0

- “Web 2.0 is the network as platform, spanning all connected device.” -- Tim O'Reilly
- Trend of using the full potential of the web
  - Viewing the Internet as a computing platform
  - Running interactive applications through a web browser
  - Leveraging interconnectivity and mobility of devices
  - The “long tail” (profits in selling specialized small market goods)
  - Enhanced effectiveness with greater human participation
- Transition from isolated information silos (Web 1.0)
- CC is defined by its architecture/infrastructure
- Web 2.0 is defined by how users see
Web 2.0
Peer-2-Peer Computing

- A distributed application architecture that partitions tasks or work loads between equally privileged peers
  - Peers donate a portion of their resources (e.g., CPU, HDD) directly available to other network participants, without the need for central coordination
  - Peers are both suppliers and consumers of resources
  - In the traditional client–server model, servers supply and clients consume
- Popularized by Napster
Web Service

- Self-describing and stateless modules that perform discrete units of work
- “Web service providers offer APIs that enable developers to exploit functionality over the Internet, rather than delivering full-blown applications.” — Infoworld
- Standards based interfaces
  - Eg, SOAP, WSDL, WS-Security
SOA: Service-Oriented Architecture

- Model for using web services
  - Service requestors, service registry, service providers
- Use of web services to compose complex, customizable, distributed applications
- Encapsulate legacy applications
- Organize stovepiped applications into collective integrated services
- Interoperability and extensibility
Virtualization

- Host operating system provides an abstraction layer for running virtual guest OSs
- Key is the hypervisor or virtual machine monitor
  - Enables guest OSs to run in isolation of other OSs
  - Run multiple types of OSs
- Increases utilization of physical servers
- Enables portability of virtual servers between physical servers
- Increases security of physical host server
- Eg, Amazon EC2
Using CC Services (SaaS)

- Calendars, Schedules, & Task Management
- Event & Contact Management
- Email
- Project Management
- Word Processing, Spreadsheets, & Presentations
- Databases
- Storing & Sharing files
- Sharing digital photographs
- Sharing songs and videos
Presentation Services in CC

- Collaborating on presentations in CC
  - Web-based or Phone-based presentation app

- Pros
  - Users from multiple locations can co-work
  - No need to carry around presentation files
  - Cost-effective—free or nominal fee!

- Cons
  - Network access is critical
  - Don’t always have the same range of features
  - Compatibility issue with existing presentations
Sharing Presentations in CC

- Web-based services that aim at sharing (not editing) existing presentation files in CC
  - Upload existing presentation files for sharing
  - Supports formats like PPT, PDF, or ODP
  - Cannot edit existing files

- Eg,
  - AuthorStream.com
  - SlideBoom.com
  - SlideShare.net
Database Services in CC

- Dabbledb.com ➔ acquired by Twitter (2010)
- Teamdesk.net
- Trackvia.com
- Baseportal.com
- Springbase.com
- Viravis.com
- Infodome.com
- Creator.zoho.com
- Quickbase.intuit.com
Reference

- **Cloud Computing Tutorial**, Peter Mell and Tim Grance, NIST, 2009