Understanding Cloud Computing: Experimentation and Capacity Planning

Daniel A. Menascé and Paul Ngo
The Volgenau School of Information Technology and Engineering
Department of Computer Science
George Mason University
www.cs.gmu.edu/faculty/menasce.html

What is Cloud Computing?

- A modality of computing characterized by *on demand* availability of resources in a dynamic and scalable fashion.
  - resource = infrastructure, platforms, software, services, or storage.
- The *cloud provider* is responsible to make the resources available on demand to the *cloud users*.
  - the cloud provider must manage its resources in an efficient way so that the user needs can be met when needed at the desired QoS level.

GEORGE MASON UNIVERSITY

The NIST Cloud Definition Framework

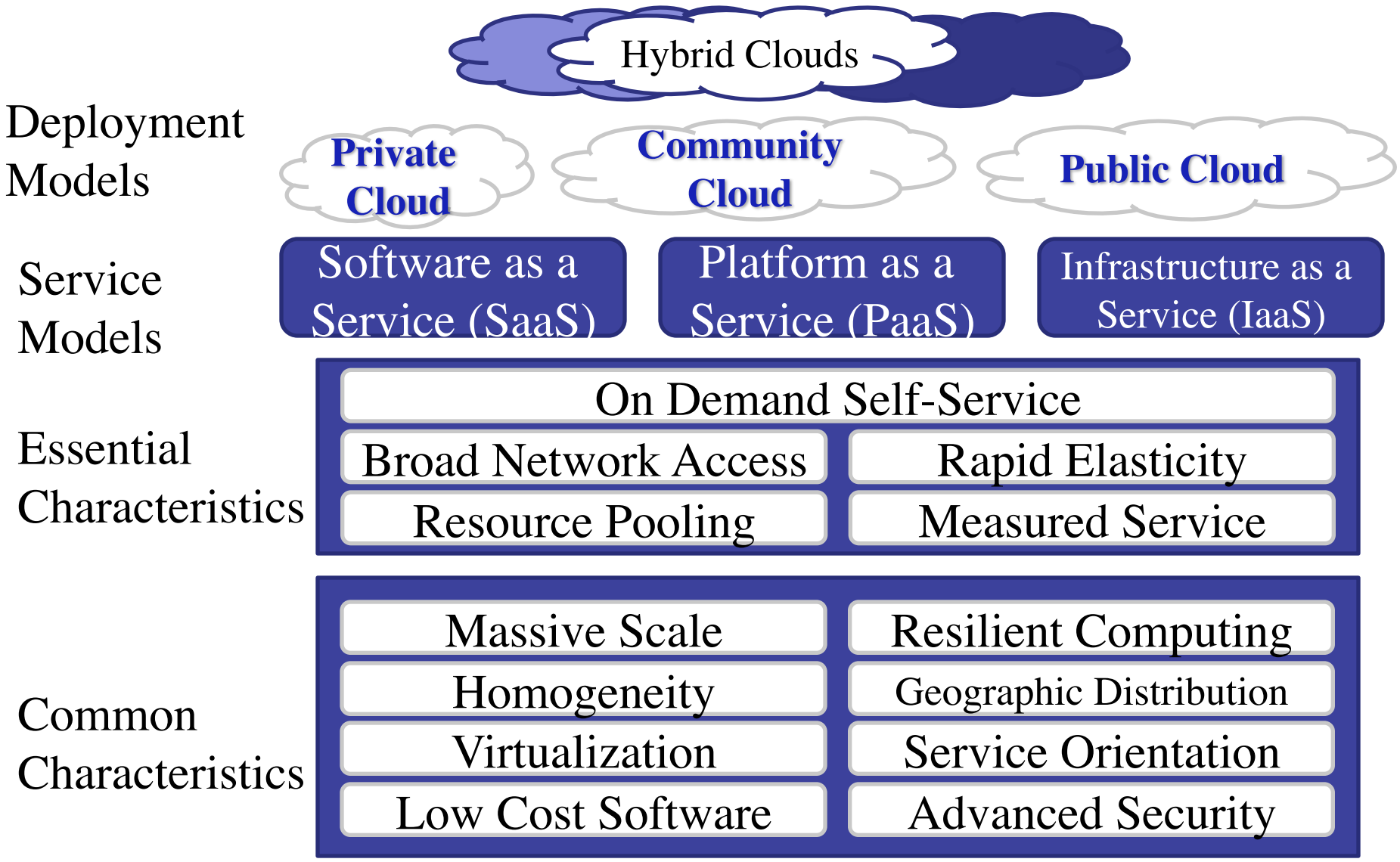
http://csrc.nist.gov/groups/SNS/cloud-computing/

- Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

- This cloud model promotes availability and is composed of five essential **characteristics,** three **service models**, and four **deployment models**.

The NIST Cloud Definition Framework

http://csrc.nist.gov/groups/SNS/cloud-computing/



Is Cloud Computing Analogous to an Electric Grid?

Consumers use electric energy on-demand according to their needs and pay based on their consumption.	**Cloud computing users use resources on demand according to their needs and pay based on their consumption.**
Power utilities have to be able to dynamically determine how to match demand and supply.	**Cloud computing providers have to be able to dynamically determine how to match demand and supply.**
The product delivered by the power grid is homogeneous (e.g., 110 V of alternating current at 60 Hz).	Clouds offer a variety of resources on demand.
One can plug any appliance to the power grid and it will work seamlessly as long as it conforms to a very simple specification of voltage and frequency.	The APIs offered by cloud providers are not standardized and may be very complicated in many cases.

Advantages of Cloud Computing

- *Pay as you go.*
- *No need to provision for peak loads.*
- *Time to market.*
- *Consistent performance and availability.*

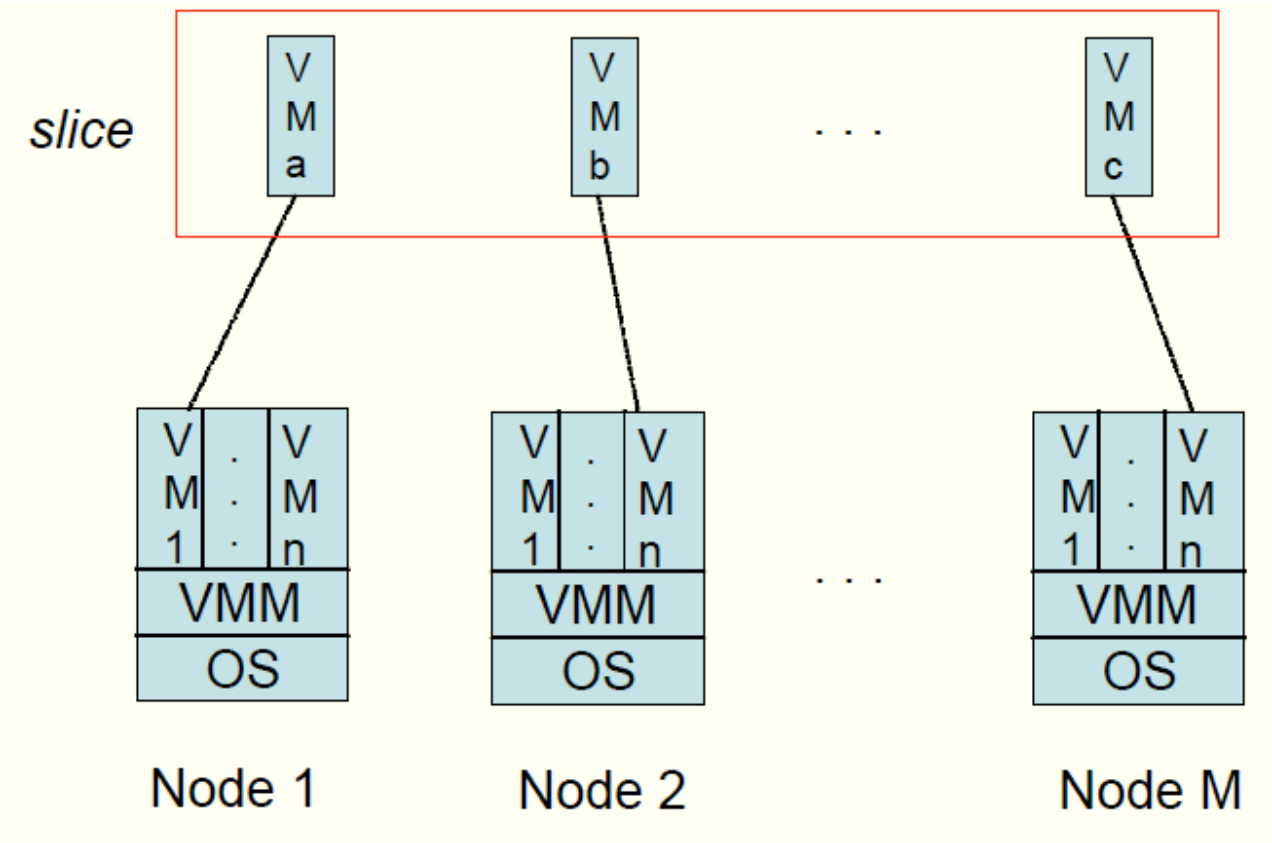
GEORGE MASON UNIVERSITY

Potential Drawbacks of Cloud Computing

- *Privacy and security*.
- *External dependency for mission critical applications*.
- *Disaster recovery*.
- *Monitoring and Enforcement of SLAs*.

GEORGE MASON UNIVERSITY

Examples: PlanetLab



Examples: PlanetLab

- 425 active sites with 985 nodes scattered over 40 countries.
- Shared resources include CPU cycles, storage, and memory.
- PlanetLab Central API allow users to create automated scripts to monitor node availability. See http://www.planet-lab.org/doc/plc_api

Examples: Amazon's Elastic Computing Cloud (EC2)

- Virtual site farm
- Users request the number and type of compute *instances* they need:
  - Standard
  - High-memory
  - High-CPU.
- Payment: by instance-hour
- One EC2 compute unit provides the equivalent of the CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

GEORGE MASON UNIVERSITY

Examples: Amazon's Elastic Computing Cloud (EC2)

- EC2's Auto Scaling allows users to determine when to scale up or down their EC2 usage.
- EC2's CloudWatch aggregates and reports metrics for CPU utilization, data transfer, and disk usage and activity for each EC2 instance.

GEORGE MASON UNIVERSITY

Amazon's EC2 Compute Instances

Standard Instances

Instances of this family are well suited for most applications.

Small Instance (default)*

1.7 GB memory
1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)
160 GB instance storage (150 GB plus 10 GB root partition)
32-bit platform
I/O Performance: Moderate

Large Instance

7.5 GB memory
4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)
850 GB instance storage (2×420 GB plus 10 GB root partition)
64-bit platform
I/O Performance: High

Extra Large Instance

15 GB memory
8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)
1,690 GB instance storage (4×420 GB plus 10 GB root partition)
64-bit platform
I/O Performance: High

http://aws.amazon.com/ec2/instance-types/

Amazon's EC2 Compute Instances

High-Memory Instances

Instances of this family offer large memory sizes for high throughput applications, including database and memory caching applications.

High-Memory Double Extra Large Instance

34.2 GB of memory
13 EC2 Compute Units (4 virtual cores with 3.25 EC2 Compute Units each)
850 GB of instance storage
64-bit platform
I/O Performance: High

High-Memory Quadruple Extra Large Instance

68.4 GB of memory
26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)
1690 GB of instance storage
64-bit platform
I/O Performance: High

http://aws.amazon.com/ec2/instance-types/

Amazon's EC2 Compute Instances

High-CPU Instances

Instances of this family have proportionally more CPU resources than memory (RAM) and are well suited for compute-intensive applications.

High-CPU Medium Instance

1.7 GB of memory
5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each)
350 GB of instance storage
32-bit platform
I/O Performance: Moderate

High-CPU Extra Large Instance

7 GB of memory
20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each)
1690 GB of instance storage
64-bit platform
I/O Performance: High

http://aws.amazon.com/ec2/instance-types/

Examples: Google's App Engine

- Web applications can be deployed on Google's infrastructures.
- Applications can run in Java or Python run-time environments.
- Free startup: all applications can use up to 500 MB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month for free.
  - After that, pay according to resource usage.

Examples: Google's App Engine

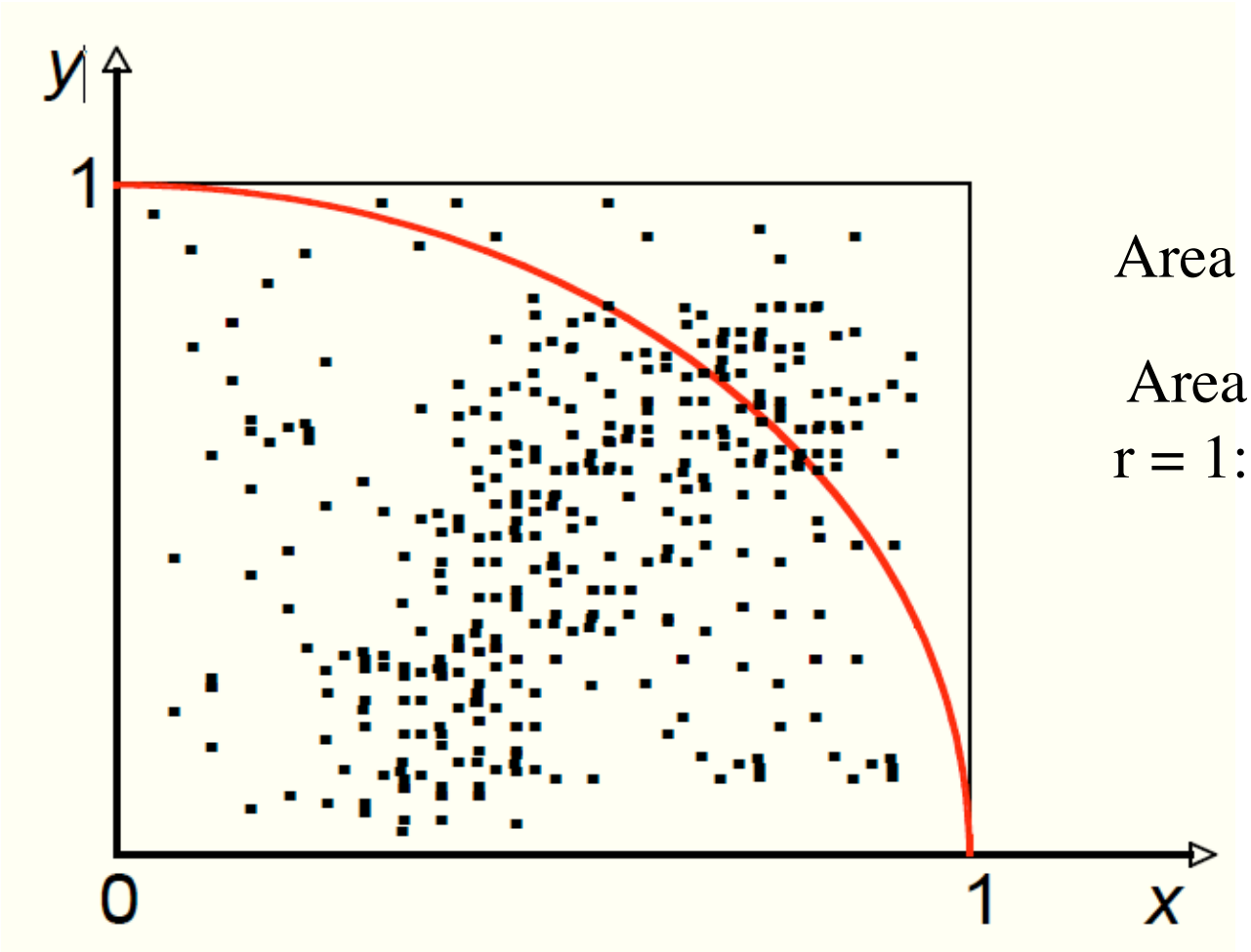
- App Engine provides a powerful distributed data storage service that features a query engine and transactions.
- Applications may include:
  - dynamic web serving
  - persistent storage
  - automatic scaling and load balancing
  - user authentication
  - task queues
  - scheduled tasks

Other Examples of Cloud Computing

- Microsoft's Azure
- Eucalyptus (http://www.eucalyptus.com/) - open source
- NSF's Cloud Computing Research Initiative (research)

GEORGE MASON UNIVERSITY

Experiments with PlanetLab: Parallel Computation of π



Area of a circle: πr^2

Area of a quadrant when $r = 1$: $\pi/4$

Experiments with PlanetLab: Parallel Computation of π

1. (At the master node). Send m and n to all n slave nodes.

2. (At each slave node i, $i = 1, \cdots, n$). $\text{NumPointsInQuadrant}_i \leftarrow 0$.

3. (At each slave node i, $i = 1, \cdots, n$). Repeat (a) and (b) m/n times.

   (a) Randomly select a point (x, y) such that x and y are random numbers uniformly distributed in [0,1].

   (b) If $\sqrt{x^2 + y^2} \leq 1$ then $\text{NumPointsInQuadrant}_i \leftarrow \text{NumPointsInQuadrant}_i + 1$.

4. (At each slave node $i, i = 1, \cdots, n$). Send $\text{NumberPointsInQuadrant}_i$ to the master node.

5. (At the master node). $\pi \leftarrow 4 \times \sum_{i=1}^{n} \text{NumPointsInQuadrant}_i / m$.

Experiments with PlanetLab: Parallel Computation of π

1. (At the master node). Send m and n to all n slave nodes.

2. (At each slave node i, $i = 1, \cdots, n$). $\text{NumPointsInQuadrant}_i \leftarrow 0$.

3. (At each slave node i, $i = 1, \cdots, n$). Repeat (a) and (b) m/n times.

   (a) Randomly select a point (x, y) such that x and y are random numbers uniformly distributed in [0,1].

   (b) If $\sqrt{x^2 + y^2} \leq 1$ then $\text{NumPointsInQuadrant}_i \leftarrow \text{NumPointsInQuadrant}_i + 1$.

4. (At each slave node $i, i = 1, \cdots, n$). Send $\text{NumberPointsInQuadrant}_i$ to the master node.

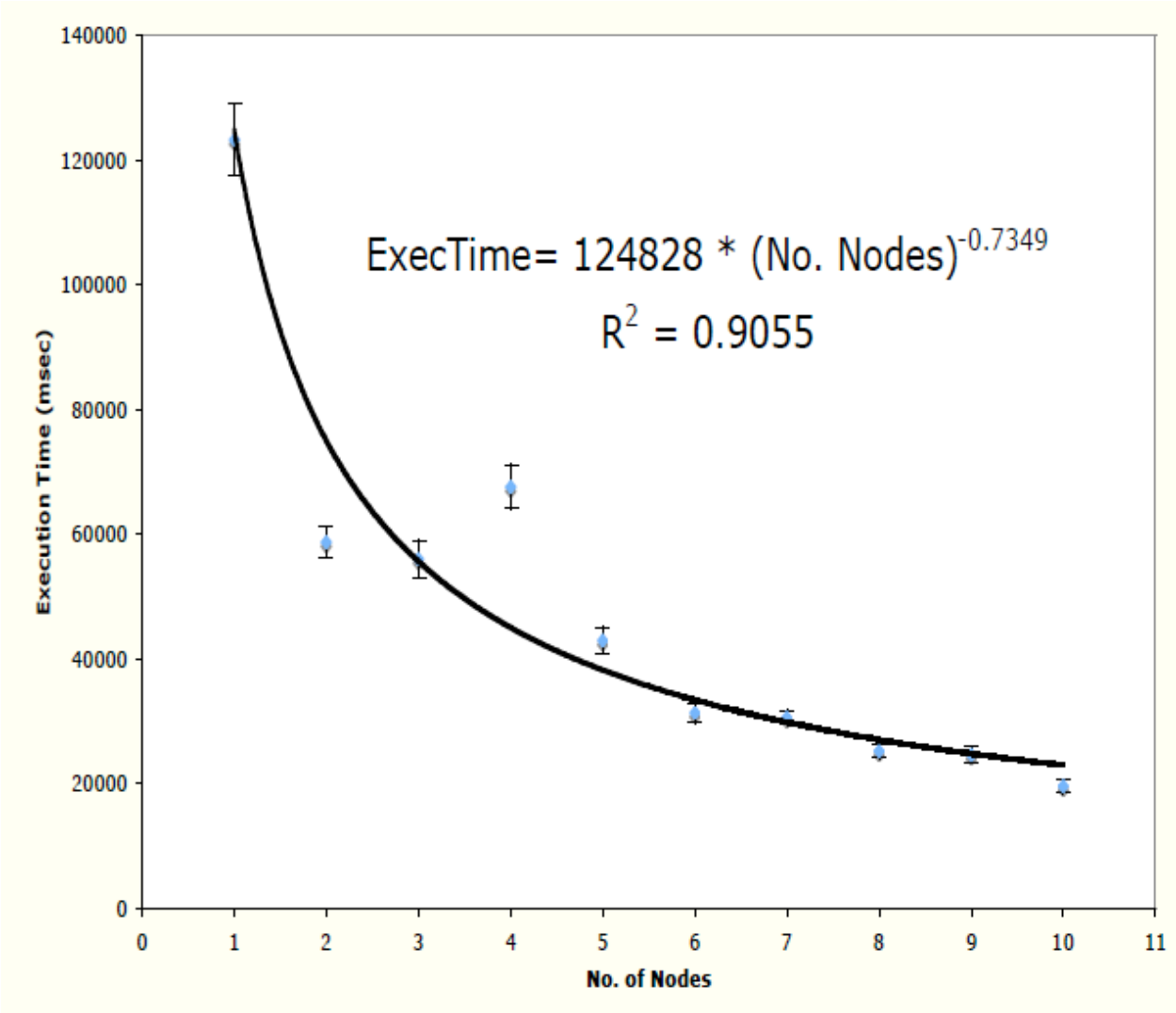
5. (At the master node). $\pi \leftarrow 4 \times \sum_{i=1}^{n} \text{NumPointsInQuadrant}_i / m$.

Experiments with PlanetLab: Parallel Computation of π

Node Name	Location	Participation
UP1	Univ. Pennsylvania	1-10
UP2	Univ. Pennsylvania	2-10
GT1	Georgetown University	3-10
GT2	Georgetown University	4-10
GMU3	George Mason University	5-10
GMU4	George Mason University	6-10
CT1	Caltech	7-10
CT2	Caltech	8-10
CN4	Cornell University	9-10
VT	Virginia Tech	10

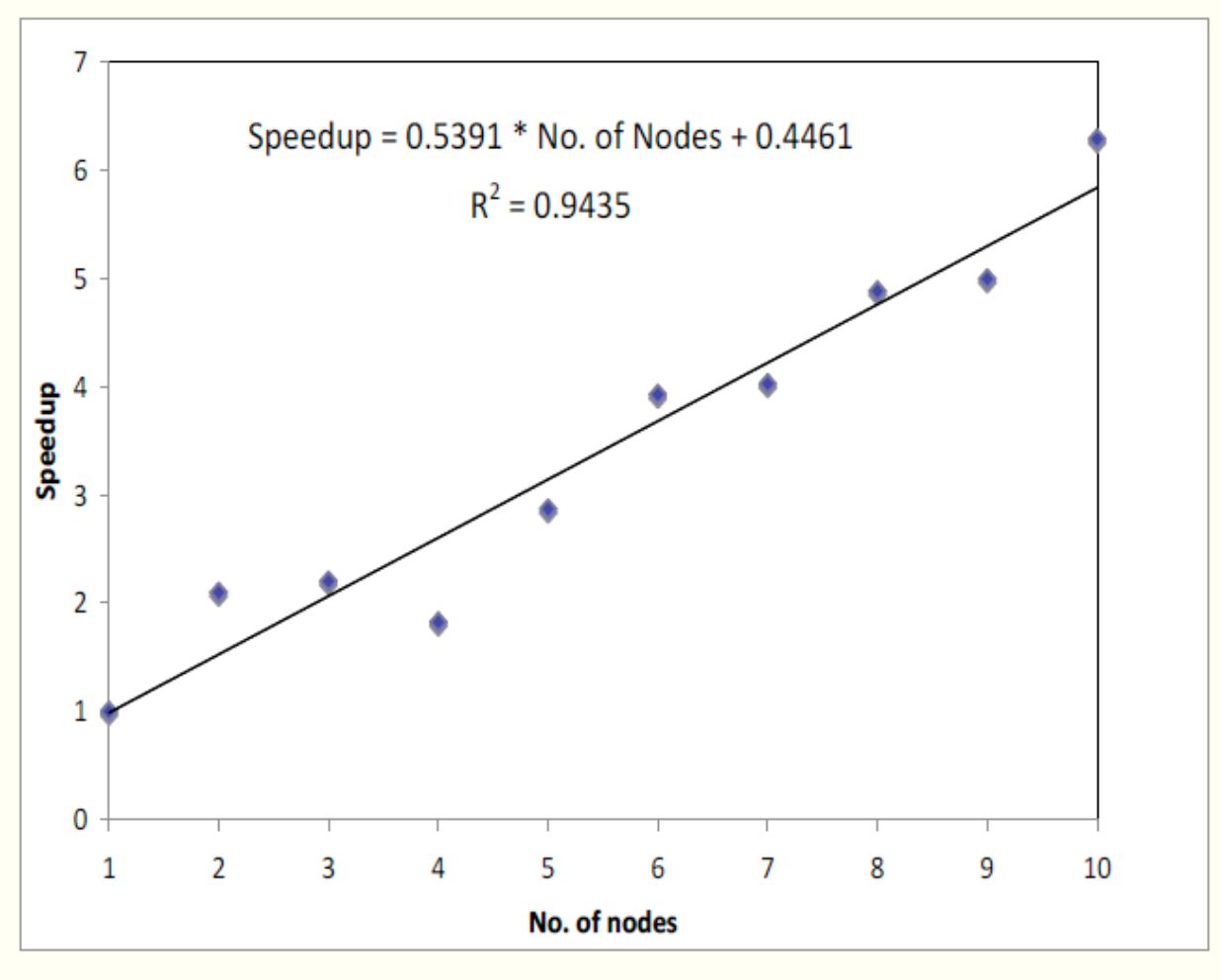
All nodes: 2 Intel Core 2 Duo E6550 Processor @ 2.44 GHz with 3.44GB memory.

Experiments with PlanetLab: Parallel Computation of π



m = 1 billion

Experiments with PlanetLab: Parallel Computation of π



$$S = \frac{E_1}{E_n}$$

m = 1 billion

Capacity Planning for the Cloud: from the consumer's point of view

- Problems for consumers:
  - How to select SLAs for various QoS metrics in a way that maximizes a utility function for the consumer subject to cost-constraints?

GEORGE MASON UNIVERSITY

Capacity Planning for the Cloud: from the consumer's point of view

- $\mathrm{SLA_r}$: SLA (in seconds) on the average response time per transaction.
- $\mathrm{SLA_x}$: SLA (in tps) on the transaction throughput.
- $\mathrm{SLA_a}$: SLA on the cloud availability.
- $C_r(\mathrm{SLA_r})$: per transaction cost (in cents) when the negotiated response time SLA is $\mathrm{SLA_r}$.
- $C_x(\mathrm{SLA_x})$: per transaction cost (in cents) when the negotiated throughput SLA is $\mathrm{SLA_x}$.
- $C_a(\mathrm{SLA_a})$: per transaction cost (in cents) when the negotiated availability SLA is $\mathrm{SLA_a}$.

GEORGE MASON UNIVERSITY

Capacity Planning for the Cloud: from the consumer's point of view

- U: global utility. The global utility function is composed of terms that represent the utility for various metrics such as response time, throughput, and availability. The utility is a dimensionless number in the [0,1] range.

- w_r, w_x, w_a: weights associated to response time, throughput, and availability, respectively, used to compute the global utility. $w_r + w_x + w_a = 1$.

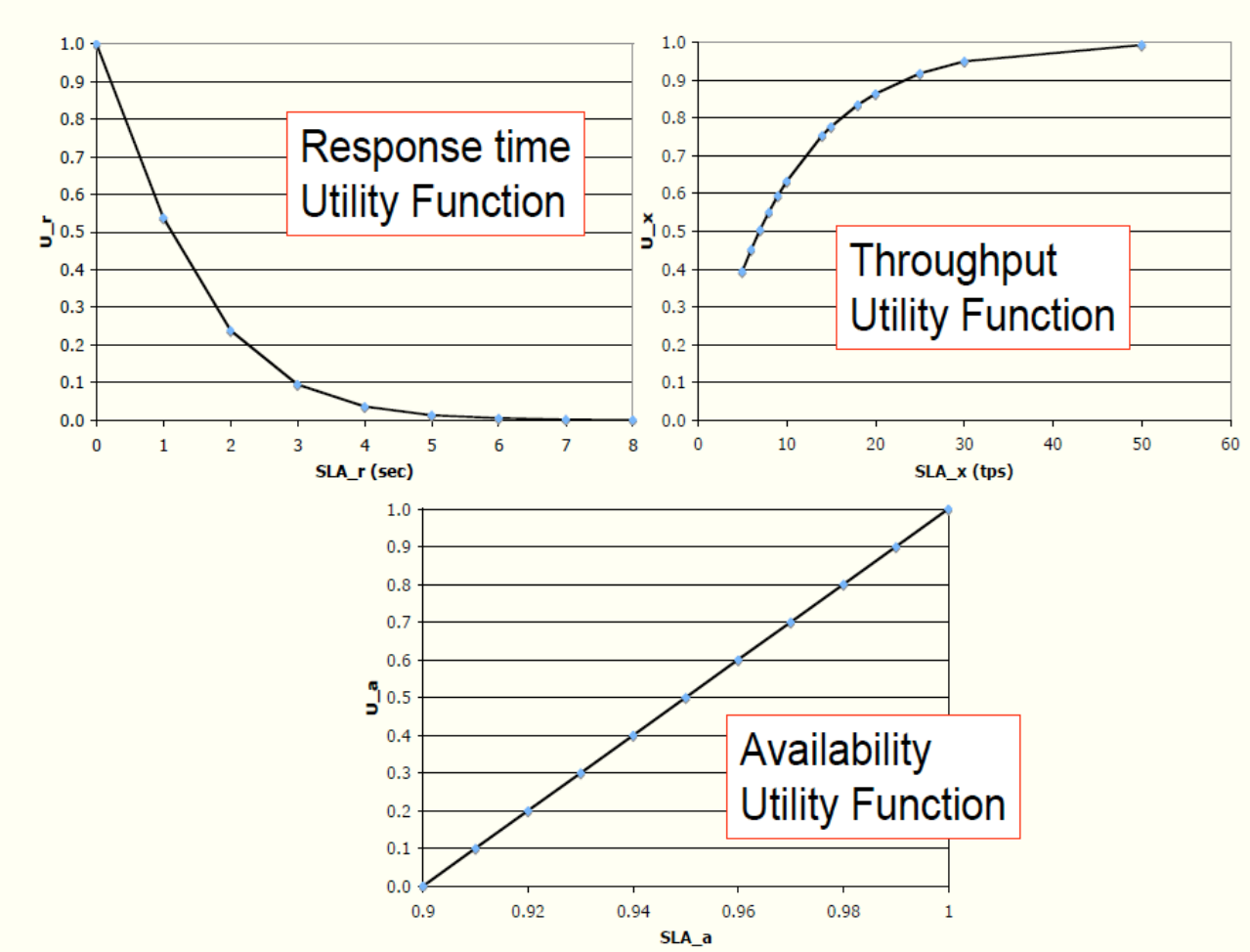
GEORGE MASON UNIVERSITY

Capacity Planning for the Cloud: from the consumer's point of view

$$
\begin{aligned}
C_r(\mathrm{SLA_r}) &= \alpha_r e^{-\beta_r\ \mathrm{SLA_r}} \\
C_x(\mathrm{SLA_x}) &= \alpha_x\ \mathrm{SLA_x} \\
C_a(\mathrm{SLA_a}) &= e^{\beta_a\ \mathrm{SLA_a}} - e^{0.9\beta_a},\ \mathrm{SLA_a} \geq 0.9.
\end{aligned}
$$

$$
\begin{aligned}
U &= w_r\ \frac{2.0\ e^{-\mathrm{SLA_r}}}{1 + e^{-\mathrm{SLA_r}}} + \\
& \quad w_x(1 - e^{-0.1\ \mathrm{SLA_x}}) + w_a(10\ \mathrm{SLA_a} - 9).
\end{aligned}
$$

Capacity Planning for the Cloud: from the consumer's point of view



Capacity Planning for the Cloud: from the consumer's point of view

$$
\begin{aligned}
maximize\ U &= f(\mathrm{SLA_r}, \mathrm{SLA_x}, \mathrm{SLA_a}) \\
subject\ to & \\
\gamma_r^{\min} & \leq \mathrm{SLA_r} \leq \gamma_r^{\max} \\
\gamma_x^{\min} & \leq \mathrm{SLA_x} \leq \gamma_x^{\max} \\
\gamma_a^{\min} & \leq \mathrm{SLA_a} \leq \gamma_a^{\max} \\
C_r(\mathrm{SLA_r}) &+ C_x(\mathrm{SLA_x}) + C_a(\mathrm{SLA_a}) \leq C_{max}
\end{aligned}
$$

Capacity Planning for the Cloud: from the consumer's point of view

$$
\begin{aligned}
maximize\ U &= f(\mathrm{SLA_r}, \mathrm{SLA_x}, \mathrm{SLA_a}) \\
subject\ to & \\
\gamma_r^{\min} & \leq \mathrm{SLA_r} \leq \gamma_r^{\max} \\
\gamma_x^{\min} & \leq \mathrm{SLA_x} \leq \gamma_x^{\max} \\
\gamma_a^{\min} & \leq \mathrm{SLA_a} \leq \gamma_a^{\max} \\
C_r(\mathrm{SLA_r}) &+ C_x(\mathrm{SLA_x}) + C_a(\mathrm{SLA_a}) \leq C_{max}
\end{aligned}
$$

Solvers:

- NEOS: http://neos.mcs.anl.gov
- MS Excel's Solver (see Tools menu)

Capacity Planning for the Cloud: from the consumer's point of view

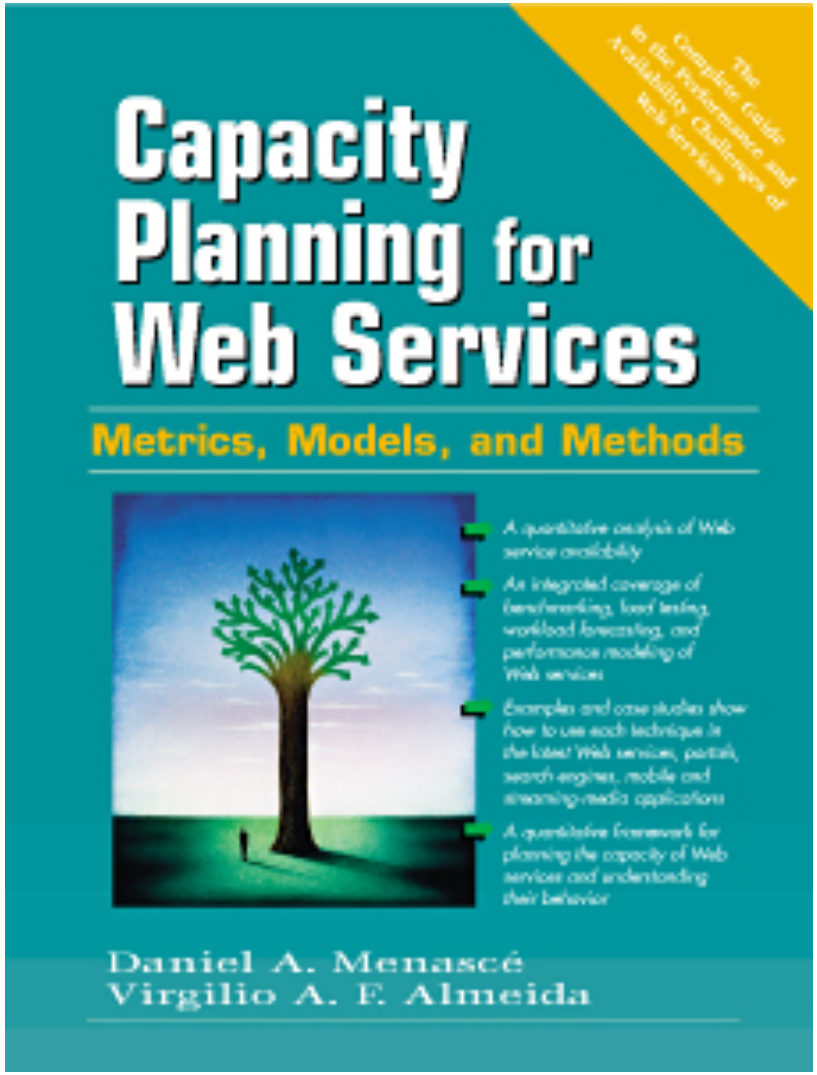
C_{max}	Utility	SLA_r	SLA_x	SLA_a
0.70	0.641	1.000	5.625	0.999
0.60	0.438	3.543	5.000	0.999
0.55	0.366	4.000	5.000	0.978
0.50	0.279	4.000	5.000	0.949

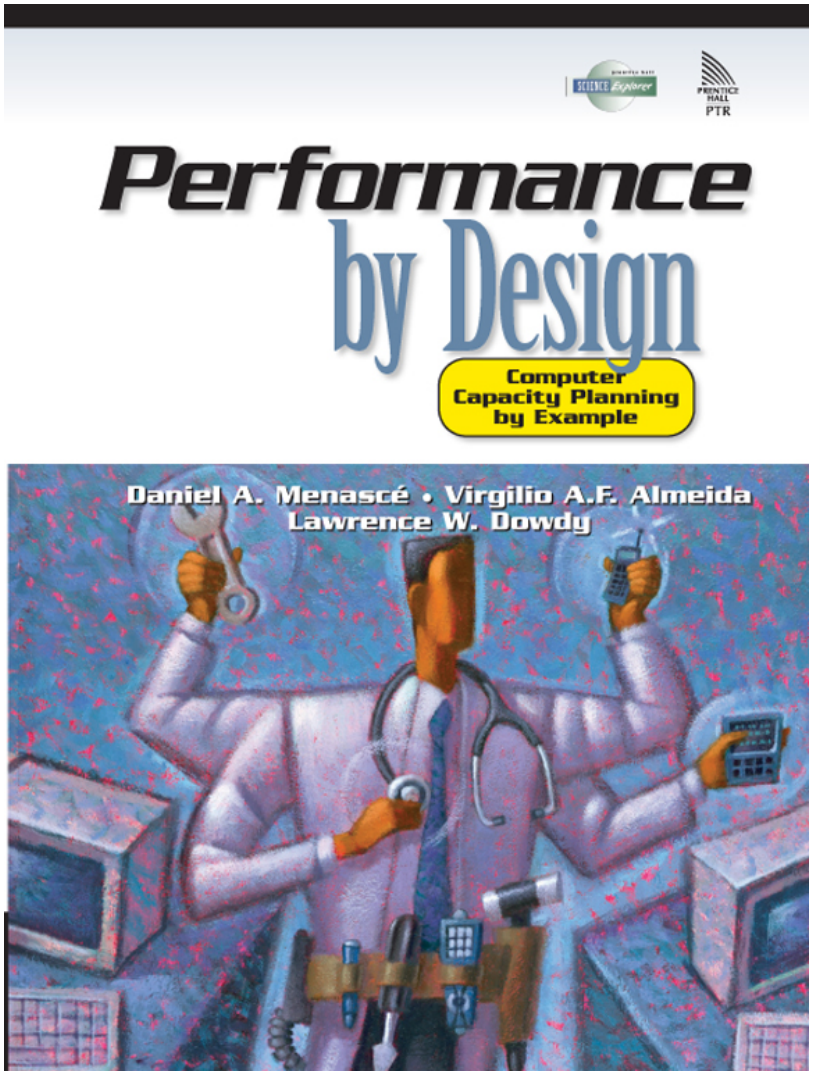
$$w_r = 0.4;\ w_x = 0.3;\ w_a = 0.3$$

Capacity Planning for the Cloud: from the provider's point of view

- Providers have to deal with:
  - Large and complex infrastructures
  - Hard to predict and time-varying workloads
- Providers need to implement autonomic computing techniques that are capable to dynamically shift resources without human intervention to cope with negotiated SLAs.

GEORGE MASON UNIVERSITY





www.cs.gmu.edu/faculty/menasce.html