Workload Generation and Emulation Environments

Master 2R SL module MD

Jean-Marc Vincent and Arnaud Legrand

Laboratory ID-IMAG
MESCAL Project
Universities of Grenoble
{Jean-Marc.Vincent,Arnaud.Legrand}@imag.fr

February 15, 2007

Outline

Outline

1. Simulation Environments
   - Introduction
   - Operating System Emulation
   - Network Emulation
   - Grid Emulation
   - World-wide Platform "Emulation"

2. Generating Synthetic Platforms
   - Topology
   - Compute resources
   - Background conditions

Outline

Reminder on Simulation

Simulation = model implementation

more abstract ↑	Mathematical Simulation	Based solely on equations
	Discrete-Event Simulation	Abstraction of system as a set of dependant actions and events (fine- or coarse- grain)
less abstract ↓	Emulation	Trapping and virtualization of low-level application/system actions

Simulation = model implementation

		Network
more abstract ↑	Mathematical Simulation	Macroscopic: flows in a pipe (coarse-grain d.e simulation + math. simulation)
	Discrete-Event Simulation	Microscopic: packet-level (fine-grain d.e. simulation)
less abstract ↓	Emulation	Actual flows go through some network

Reminder on Simulation

Simulation = model implementation

		CPU
more abstract ↑	Mathematical Simulation	Macroscopic: flows in a pipe (coarse-grain d.e simulation + math. simulation)
	Discrete-Event Simulation	Microscopic: Cycle-accurate simulation (fine-grain d.e. simulation)
less abstract ↓	Emulation	Virtualization via another CPU/virtual machine

Reminder on Simulation

Simulation = model implementation

		Application
more abstract ↑	Mathematical Simulation	Macroscopic: application = analytical "flow"
	Discrete-Event Simulation	Less Macroscopic: set of abstract tasks with resource needs and dependancies
less abstract ↓	Emulation	Virtualization (emulation of actual code with trapping of application generated events)

Reminder on Simulation

Simulation = model implementation

			Application
more abstract	↑	Mathematical Simulation	Macroscopic: application = analytical "flow"
		Discrete-Event Simulation	Less Macroscopic: set of abstract tasks with resource needs and dependancies
less abstract	↓	Emulation	Virtualization (emulation of actual code with trapping of application generated events)

Boundaries above are more than blurred.

Reminder on Simulation

Simulation = model implementation

		Application
more abstract ↑	Mathematical Simulation	Macroscopic: application = analytical "flow"
	Discrete-Event Simulation	Less Macroscopic: set of abstract tasks with resource needs and dependancies
less abstract ↓	Emulation	Virtualization (emulation of actual code with trapping of application generated events)

Boundaries above are more than blurred.

Simulation can combine all paradigms at different levels. Today, we will mainly talk about emulation though.

- Xen, User Mode Linux
- ModelNet, Emulab/DummyNet
- MicroGrid
- PlatetLab

Outline

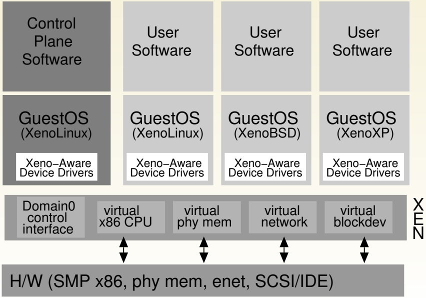
Goal run many instances of operating systems on the same physical machine.

Difficulties

- Performance Isolation (scheduling priority, memory demand, network traffic, disk accesses).
- OS compatibility (full/partial virtualization)
- Performance overhead

Common Tools User Mode Linux, Vserver, VMware, Xen

An example: Xen



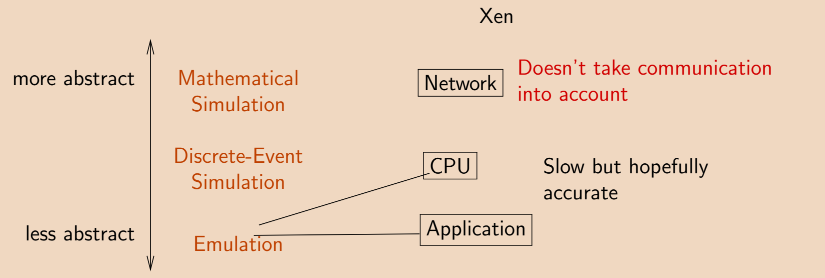
Operating System Emulation

Goal run many instances of operating systems on the same physical machine.

Difficulties

- Performance Isolation (scheduling priority, memory demand, network traffic, disk accesses).
- OS compatibility (full/partial virtualization)
- Performance overhead

Xen in a Nutshell



H/W (SMP x86, phy mem, enet, SCSI/IDE)

Outline

ModelNet

ModelNet is a UCSD/Duke project [VYW+02] lead by Amin Vahdat.

Applications are supported by emulation and virtualization: Actual application code is executed on "virtualized" resources

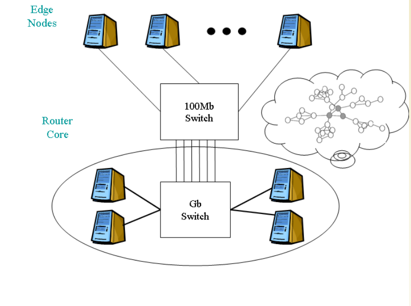
A key tradeoff in ModelNet is scalability versus accuracy.

ModelNet accounts for network but not for CPU:

Resource gethostnames, sockets are wrapped

CPU Plain mapping, slowdown is not taken into account

Network one emulator running FreeBSD, a gigabit LAN, some host machines with IP aliasing for the virtual nodes ⇝ emulation of heterogeneous links

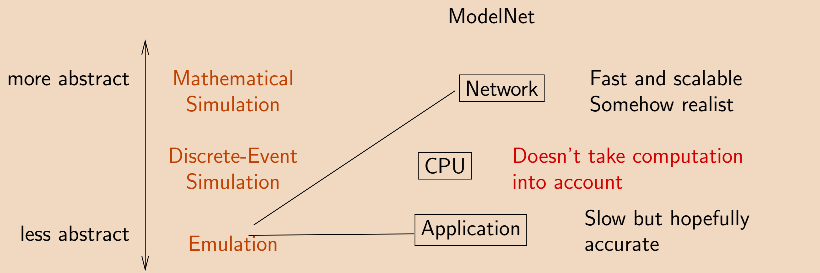


Similar ideas have been used in other projects (Emulab [WLS+02], DummyNet, Panda [KBM+02], . . .)

ModelNet is a UCSD/Duke project [VYW+02] lead by Amin Vahdat.

Applications are supported by emulation and virtualization: Actual application code is executed on "virtualized" resources

ModelNet in a Nutshell



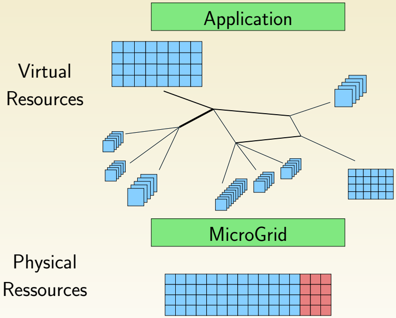
for the virtual nodes ⇝ emulation of heterogeneous links

Similar ideas have been used in other projects (Emulab [WLS+02], DummyNet, Panda [KBM+02], . . .)

Outline

MicroGrid is a UCSD project [SLJ+00] lead by Andrew Chien.
Applications are supported by emulation and virtualization: Actual application code is executed on "virtualized" resources
Microgrid accounts for CPU and network

- **Resource** gethostnames, sockets, GIS, MDS, NWS are wrapped
- **CPU** Direct execution on a fraction of physical CPU: find a good mapping
- **Network** Packet-level simulation (parallel version of MaSSF)
- **Time** Synchronize real time and virtual time: find the good execution rate

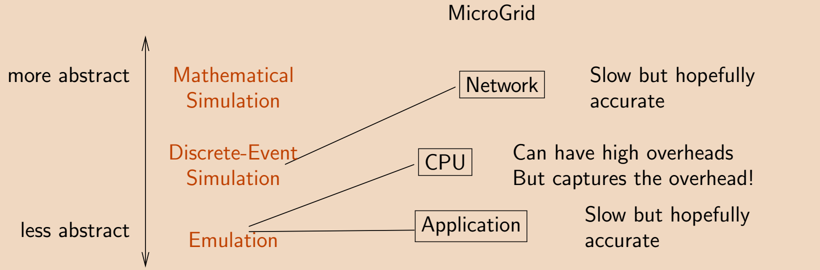


MicroGrid is a UCSD project [SLJ+00] lead by Andrew Chien.

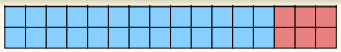
Applications are supported by emulation and virtualization: Actual application code is executed on "virtualized" resources

Microgrid accounts for CPU and network

MicroGrid in a Nutshell



virtual time: find the good execution rate



Outline

PlanetLab

PlanetLab is an open platform for developping, deploying, and accessing planetary-scale services.

Planetary-scale 350 machines, 140 sites, 20 countries

Distribution Virtualization each user can get a slice of the platform.

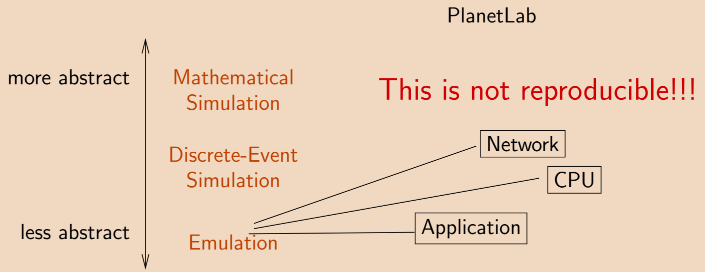
Unbundled Management

- OS defines only local (per-node) behavior (global (network-wide) behavior implemented by services)
- multiple competing services running in parallel (shared, unprivileged interfaces)

Unstable like the real world Convenient to try P2P applications or demonstrate the feasability of a middleware.

PlanetLab is an open platform for developping, deploying, and accessing planetary-scale services.

PlanetLab in a Nutshell



demonstrate the feasability of a middleware.

Outline

Why Synthetic Platforms?

Two goals of simulations:

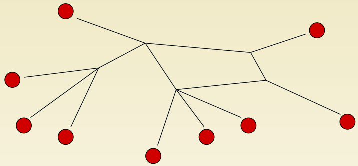
- Simulate platforms beyond the ones at hand
- Perform sensitivity analyses

Need: Synthetic platforms

- Examine real platforms
- Discover principles
- Implement "platform generators"

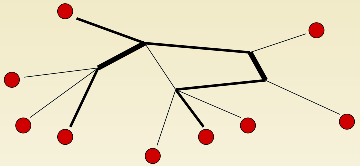
Network Topology

- Graph



Network Topology

- Graph
- Bandwidth and Latencies



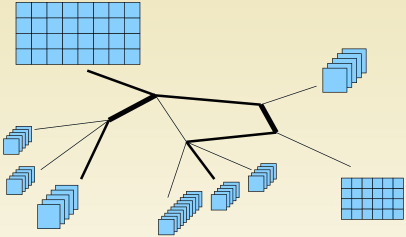
Generation of Synthetic Platforms

Network Topology

- Graph
- Bandwidth and Latencies

Compute Resources And other resources "Background" Conditions

- CPU capacity

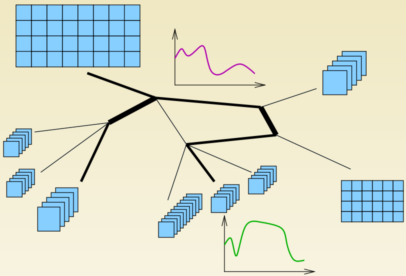


Network Topology

- Graph
- Bandwidth and Latencies

Compute Resources And other resources "Background" Conditions

- CPU capacity
- Load

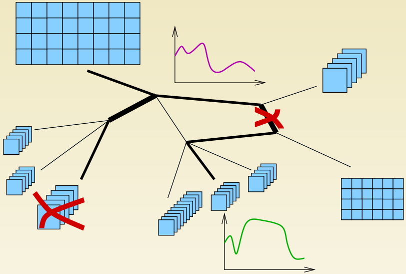


Network Topology

- Graph
- Bandwidth and Latencies

Compute Resources And other resources "Background" Conditions

- CPU capacity
- Load
- Failures



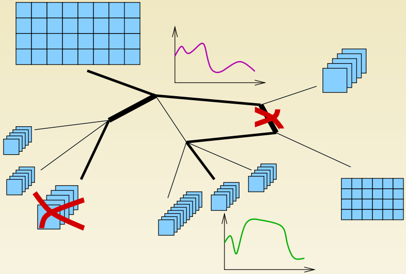
Generation of Synthetic Platforms

Network Topology

- Graph
- Bandwidth and Latencies

Compute Resources And other resources "Background" Conditions

- CPU capacity
- Load
- Failures



What is Representative and Tractable?

Outline

Synthetic Network Topologies

The network community has wondered about the properties of the Internet topology for years

- The Internet grows in a decentralized fashion with seemingly complex rules and incentives
- Could it have a mathematical structure?
- Could we then have generative models?

Three "generations of graph generators"

- "Plain" Random
- Structural
- Degree-based

Flat models

Brain-dead N dots are randomly chosen (using a uniform distribution) in a square. Then they are randomly connected with a uniform probability α.

Flat models

Brain-dead N dots are randomly chosen (using a uniform distribution) in a square. Then they are randomly connected with a uniform probability α.

Waxman [Wax88] Dots are randomly placed on a square of side c and are randomly connected with a probability $P(u, v) = \alpha e^{-d/(\beta L)}$, $0 < \alpha, \beta \leqslant 1$ where d is the Euclidean distance between u and v and $L = c\sqrt{2}$.

Flat models

Brain-dead N dots are randomly chosen (using a uniform distribution) in a square. Then they are randomly connected with a uniform probability α.

Waxman [Wax88] Dots are randomly placed on a square of side c and are randomly connected with a probability $P(u,v) = \alpha e^{-d/(\beta L)}$, $0 < \alpha, \beta \leqslant 1$ where d is the Euclidean distance between u and v and $L = c\sqrt{2}$. The edge number increases with α and the edge length heterogeneity increases with β.

Flat models

Brain-dead N dots are randomly chosen (using a uniform distribution) in a square. Then they are randomly connected with a uniform probability α.

Waxman [Wax88] Dots are randomly placed on a square of side c and are randomly connected with a probability $P(u,v) = \alpha e^{-d/(\beta L)}$, $0 < \alpha, \beta \leqslant 1$ where d is the Euclidean distance between u and v and $L = c\sqrt{2}$. The edge number increases with α and the edge length heterogeneity increases with β.

Exponential Dots are randomly placed and are connected with a probability $P(u,v) = \alpha e^{-d/(L-d)}$.

Flat models

Brain-dead N dots are randomly chosen (using a uniform distribution) in a square. Then they are randomly connected with a uniform probability α.

Waxman [Wax88] Dots are randomly placed on a square of side c and are randomly connected with a probability $P(u,v) = \alpha e^{-d/(\beta L)}$, $0 < \alpha, \beta \leqslant 1$ where d is the Euclidean distance between u and v and $L = c\sqrt{2}$. The edge number increases with α and the edge length heterogeneity increases with β.

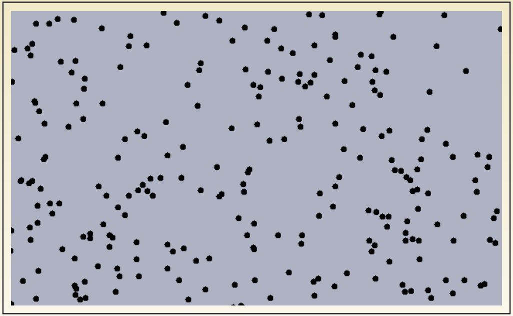
Exponential Dots are randomly placed and are connected with a probability $P(u,v) = \alpha e^{-d/(L-d)}$.

Locality [ZCD97] This model is due to Zegura. Dots are randomly placed and are connected with a probability

$$P(u,v) = \begin{cases} \alpha & \text{if } d < L \times r \\ \beta & \text{if } d \geqslant L \times r \end{cases}.$$

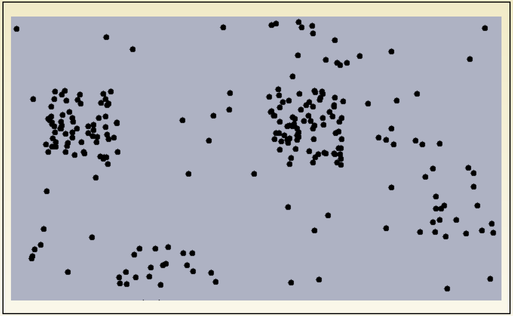
Node placement

Uniform



Node placement

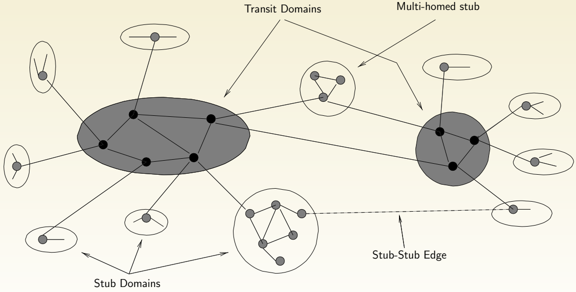
Heavy Tailed



What about hierarchy ?

Top-Down

N-level [ZCD97] Starting from a connected graph, at each step, a node is replaced by another connected graph (Tiers, GT-ITM).

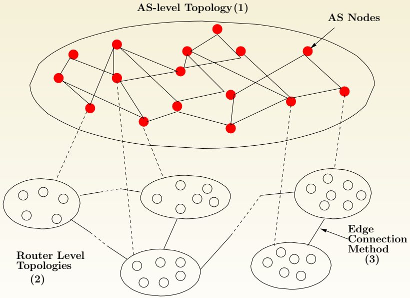


What about hierarchy ?

Top-Down

N-level [ZCD97] Starting from a connected graph, at each step, a node is replaced by another connected graph (Tiers, GT-ITM).

Transit-stub [ZCD97] 2-levels of hierarchy and some additional edges (GT-ITM, BRITE).



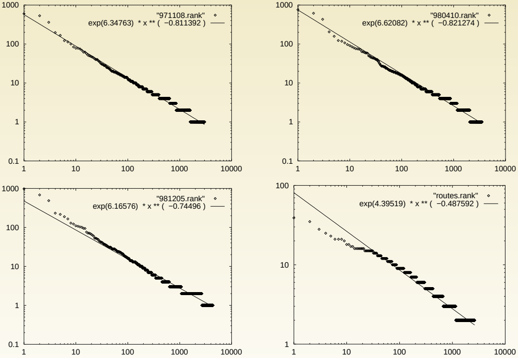
Power-Law : rank exponent

Faloutsos brothers [FFF99] have analyzed the topology at the AS level and have established power-laws describing this topology.

The rank r_v of a note v is its index in the order of decreasing degree.

Power-Law : rank exponent

Faloutsos brothers [FFF99] have analyzed the topology at the AS level and have established power-laws describing this topology.
The rank r_v of a note v is its index in the order of decreasing degree.



Power-Law : rank exponent

Faloutsos brothers [FFF99] have analyzed the topology at the AS level and have established power-laws describing this topology.
The rank r_v of a note v is its index in the order of decreasing degree.

Power Law (rank exponent).

Given a graph, the degree d_v of a node v is proportional to the rank of the node r_v to the power of a constant $\mathcal{R}$.

$$d_v \propto r_v^{\mathcal{R}}$$

	Nov. 97	Apr. 98	Dec. 98	Router 95
$\mathcal{R}$	0,81	0,82	0,74	0,48

What about Power Laws ?

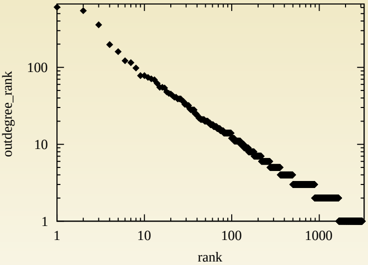
Incremental growth and affinity lead to Power Laws [BA99].

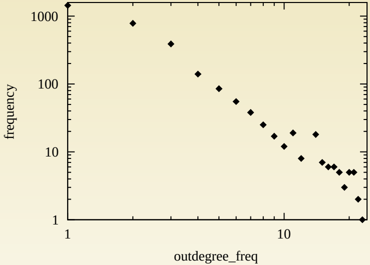
What about Power Laws ?

Incremental growth and affinity lead to Power Laws [BA99].
Nodes are incrementally added. The probability that v is connected to u depends on d_u:

$$P(u,v) = \frac{d_u}{\sum_k d_k}$$

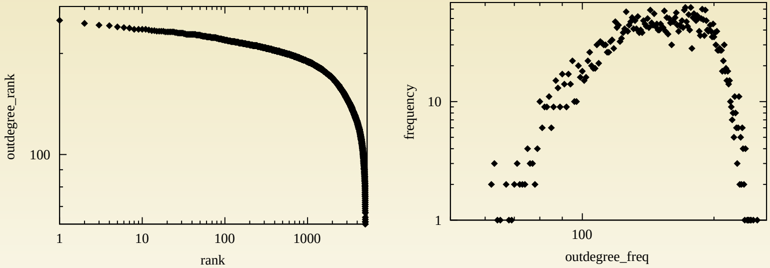
Let us check two Power-Laws





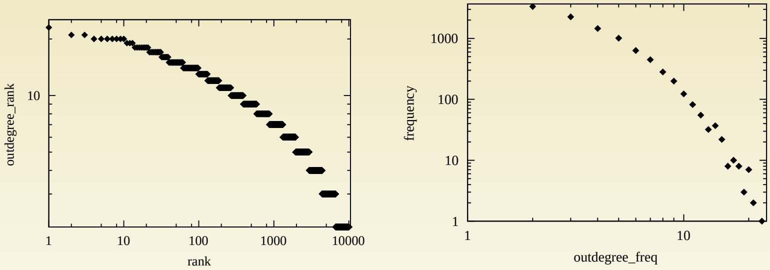
Interdomain November 1997

Let us check two Power-Laws



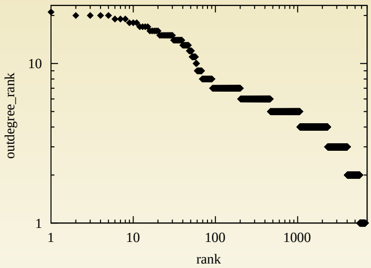
GT-ITM flat ?

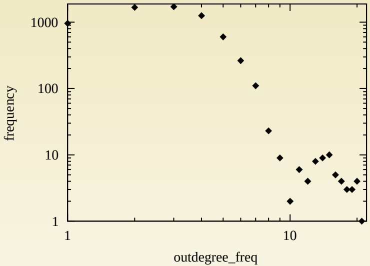
Let us check two Power-Laws



Waxman (BRITE)

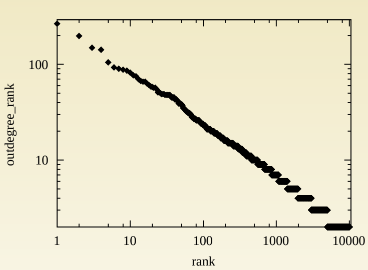
Let us check two Power-Laws





Transit-Stub (GT-ITM)

Let us check two Power-Laws



Barábasi Albert (BRITE)

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated.

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias.

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion
- Excentricity distribution

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion
- Excentricity distribution
- Eigenvalues distribution

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion
- Excentricity distribution
- Eigenvalues distribution
- Set cover size, . . .

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion
- Excentricity distribution
- Eigenvalues distribution
- Set cover size, ...

Observation [TGJ+02]:

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion
- Excentricity distribution
- Eigenvalues distribution
- Set cover size, ...

Observation [TGJ+02]:

- AS-level and router-level have similar characteristics

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion
- Excentricity distribution
- Eigenvalues distribution
- Set cover size, ...

Observation [TGJ+02]:

- AS-level and router-level have similar characteristics
- Degree-based generators are significantly better at representing large scale properties of the Internet than structural ones.

Are these measurements meaningful?

We know network have structure AND power laws. Some projects try to combine both (GridG [LD03]).
However, some of the power laws observed by the Faloutsos brothers are correlated. Power laws could be a measurement bias. What kind of measurements can be used ?

- Expension
- Resilience
- Distortion
- Excentricity distribution
- Eigenvalues distribution
- Set cover size, ...

Observation [TGJ+02]:

- AS-level and router-level have similar characteristics
- Degree-based generators are significantly better at representing large scale properties of the Internet than structural ones.
- Hierarchy seem to arise from degree-based generators.

I'm lost. . . what should I do then?

- For a 10 000 nodes platform, degree-based generators seem to give good results.

I'm lost. . . what should I do then?

- For a $10\,000$ nodes platform, degree-based generators seem to give good results.
- For a 100 node platform, power laws do not make sense and structural generators may be more appropriate.

I'm lost... what should I do then?

- For a $10\,000$ nodes platform, degree-based generators seem to give good results.
- For a 100 node platform, power laws do not make sense and structural generators may be more appropriate.

We need some additional informations
(e.g. routing, bandwidth, latency, sharing capacity, ...).

Bandwidths, latencies, traffic, etc.

Topology generators only produce a graph: We need link characteristics as well

1. Model physical characteristics.
   - Some "models" in topology generators
   - Need to simulate background traffic
   - No accepted model for generating background traffic
   - Simulation can be very costly
2. Model end-to-end performance. Models ([LS01]) or Measurements (NWS, ...). Go from path modeling to link modeling?

Turns out to be a difficult question (DARPA workshop on network modeling).

Bandwidths, latencies, traffic, etc.

Maybe none of this matters?

- Fiber inside the network mostly unused
- Communication bottleneck is the local link
- Appropriate tuning of TCP or better protocols should saturate the local link
- Don't care about topology at all!
- Maybe none of this matters for my application (no network contention)

Outline

Compute Resources

What resources do we put at the end- points?

"Ad-hoc" generalization Look at the TeraGrid. Generate new sites based on existing sites

Statistical modeling

- Examing many production resources
- Identify key statistical characteristics
- Come up with a generative/predictive model

Synthetic Clusters?

Many Grid resources are clusters. What is the "typical" distribution of clusters?

"Commodity Cluster synthesizer" [KCC04a]

- Examined 114 production clusters (10K+ procs)
- Came up with statistical models
  - Linear fit between clock-rate and release-year within a processor family
  - Quadratic fraction of processors released on a given year
- Validated model against a set of 191 clusters (10K+ procs)
- Models allow "extrapolation" for future configurations
- Models implemented in a resource generator

Outline

Probabilistic models

- Naive: experimental distributed availability and unavailability intervals
- Sophisticated: Weibull distributions [NBW05]

Traces NWS, Desktop Grid resources [KCC04b]

Workload models e.g. batch schedulers

- Traces
- Models [LF03]: job inter-arrival times (Gamma), amount of work requested (Hyper-Gamma), number of processors requested: Compounded (2^p, 1, ...)
- Adversary ?

A Sample Synthetic Grid ?

- Generate a 5,000 node graph with BRITE
- Annotate latency according to BRITE's Euclidian distance method (scaling to obtain the desired network diameter)
- Annotate bandwidth based on a set of end-to-end NWS measurements
- Pick 30% of the end-points
- Generate a cluster at each end-point according to Kee's synthesizer for Year 2006
- Model cluster load with Feitelson's model with a range of parameters for the random distributions
- Model resource failures based on Inca measurements on TeraGrid

A.L. Barabási and R. Albert.
Emergence of scaling in random networks.
Science, 59:509–512, 1999.
Available at http://www.nd.edu/~networks/Papers/science.pdf.

Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos.
On power-law relationships of the internet topology.
In *SIGCOMM*, pages 251–262, 1999.
Available at http://citeseer.nj.nec.com/faloutsos99powerlaw.html.

R. K. Jain.
The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.
John Wiley & Sons Canada, Ltd., 1 edition, 1991.

Thilo Kielmann, Henri E. Bal, Jason Maassen, Rob van Nieuwpoort, Lionel Eyraud, Rutger Hofman, and Kees Verstoep.

Programming environments for high-performance grid computing: the albatross project.
Future Generation Computer Systems, 2002.
Available at http://www.cs.vu.nl/~kielmann/papers/fgcs01.ps.gz.

Yang-Suk Kee, Henri Casanova, and Andrew A. Chien.
Realistic modeling and synthesis of resources for computational grids.
In *Proceedings of Supercomputing 2004*. IEEE Computer Society Press, 2004.

Derrick Kondo, Andrew A. Chien, and Henri Casanova.
Resource management for rapid application turnaround on enterprise desktop grids.
In *Proceedings of Supercomputing 2004*. IEEE Computer Society Press, 2004.

D. Lu and P. Dinda.
GridG: Generating realistic computational grids.
Performance Evaluation Review, 30(4), March 2003.

Available at http://www.cs.nwu.edu/~pdinda/Papers/lu-dinda-per.pdf.

Uri Lublin and Dror G. Feitelson.
The workload on parallel supercomputers: modeling the characteristics of rigid jobs.
Journal of Parallel and Distributed Computing, 63(11):1105–1122, 2003.

David J. Lilja.
Measuring Computer Performance: A Practitioner's Guide - David J. Lilja - Hardcover.
Cambridge University Press, 2000.

C. Lee and J. Stepanek.
On future global grid communication performance.
In *10th Heterogeneous Computing Workshop (HCW'2001)*. IEEE Computer Society Press, 2001.

Daniel Nurmi, John Brevik, , and Rich Wolski.

Modeling machine availability in enterprise and wide-area distributed computing environments.
In *Proceedings of EuroPar 2005*, 2005.

H. J. Song, X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, Kenjiro Taura, and Andrew A. Chien.
The microgrid: a scientific tool for modeling computational grids.
In *Supercomputing*, 2000.
Available at http://www.sc2000.org/techpapr/papers/pap.pap286.pdf.

H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger.
Network topology generators: Degree-based vs structural.
In *Proceedings of SIGCOMM'02*. ACM Press, August 2002.
Available at http://citeseer.nj.nec.com/tangmunarunkit02network.html.

Amin Vahdat, Ken Yocum, Kesvin Walsh, Priya Mahadevan, Dejan Kosti0̆107, Jeffrey Chase, and David Becker.

Scalability and accuracy in a largescale network emulator.
In *Proceedings of the 5th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI)*, 2002.

Bernard M. Waxman.
Routing of Multipoint Connections.
IEEE Journal on Selected Areas in Communications, 6(9):1617–1622, December 1988.

Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, and Abhijeet Joglekar.
An integrated experimental environment for distributed systems and networks.
pages 255–270, Boston, MA, December 2002.

Ellen W. Zegura, Kenneth L. Calvert, and Michael J. Donahoo.
A quantitative comparison of graph-based models for Internet topology.
IEEE/ACM Transactions on Networking, 5(6):770–783, 1997.

Available at http://citeseer.nj.nec.com/zegura97quantitative.html.