Understanding Behavior and Improving Reliability in Complex Information Systems

Kevin Mills

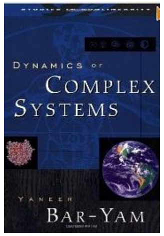
NIST

May 9, 2013

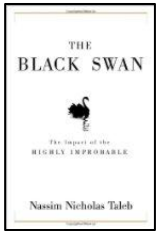
Joint work with a long list of collaborators, including statistician *Jim Filliben*, computer scientist *Chris Dabrowski*, visualization expert *Sandy Ressler*, data mining expert *Dong-Yeon Cho*, simulation expert *Jim Henriksen*, electrical engineer *Jian Yuan* and mathematicians *Fern Hunt* & *Dan Genin*, as well as NSF SURF students *Edward Schwartz* (incipient PhD from CMU), *Andrea Haines* and *Brittany Devine*.

Motivation

Background: Information Systems, increasingly central to the nation's economic well-being and security, are: large, distributed, continuously evolving, unpredictable, fragile and interdependent – in a word, Complex

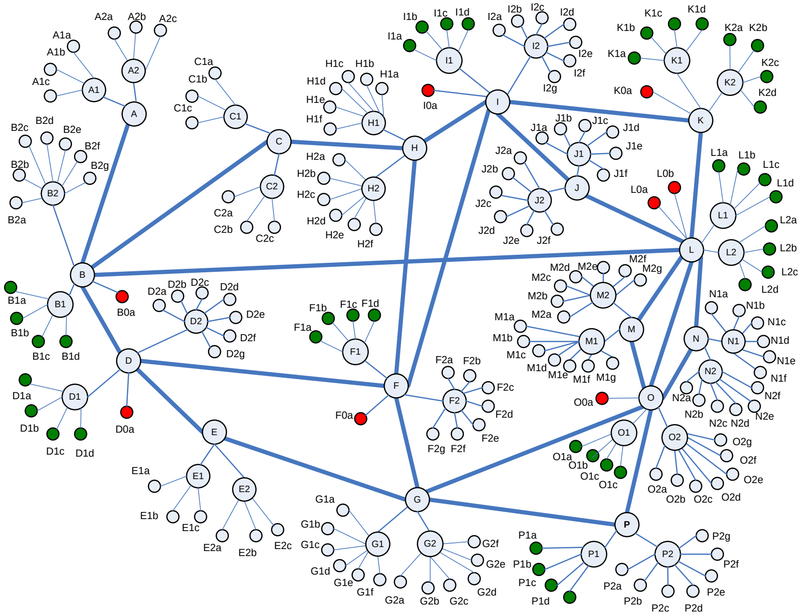


Problem I: How can we predict the effects on macroscopic behavior and user experience when new or revised components are injected into complex information systems?



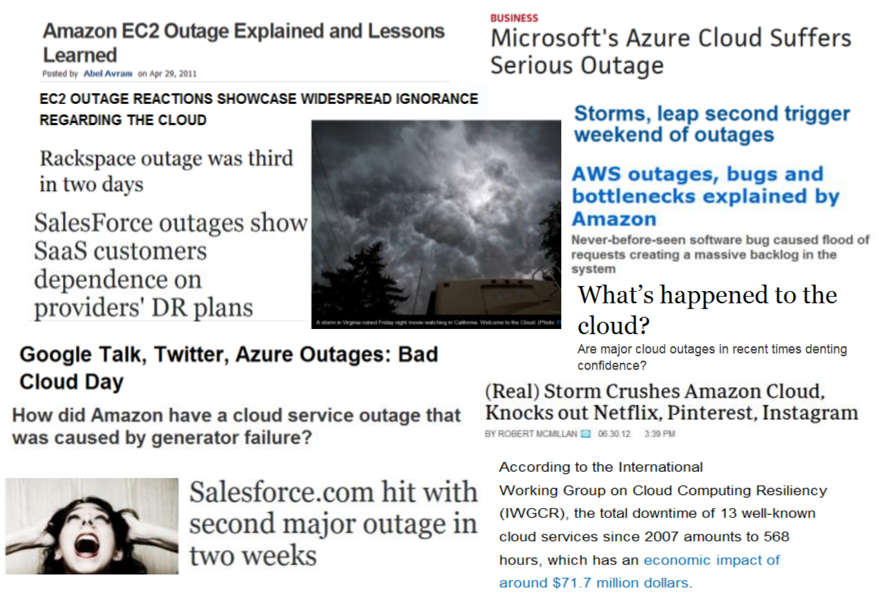
Problem II: How can we identify low-probability combinations of conditions in complex information systems that will drive macroscopic behavior into extremely costly failure regimes?

"It is now common knowledge that BGP routing policies can interact to produce unexpected routing anomalies such as protocol oscillation. We introduce a new class of anomalies, where routing is *wedged* into a local optimum that is very difficult to change." Tim Griffin, Cambridge University



"The number of websites that would now break if Amazon were to go down, and the growing pervasiveness of Amazon behind the scenes, is really quite impressive." Craig Labovitz, DeepField, quoted in *WIRED ENTERPRISE*.

Hard Problems



Amazon EC2 Outage Explained and Lessons Learned

Posted by Abel Avram on Apr 29, 2011

EC2 OUTAGE REACTIONS SHOWCASE WIDESPREAD IGNORANCE REGARDING THE CLOUD

Rackspace outage was third in two days

SalesForce outages show SaaS customers dependence on providers' DR plans

Google Talk, Twitter, Azure Outages: Bad Cloud Day

How did Amazon have a cloud service outage that was caused by generator failure?

Salesforce.com hit with second major outage in two weeks

BUSINESS

Microsoft's Azure Cloud Suffers Serious Outage

Storms, leap second trigger weekend of outages

AWS outages, bugs and bottlenecks explained by Amazon

Never-before-seen software bug caused flood of requests creating a massive backlog in the system

What's happened to the cloud?

Are major cloud outages in recent times denting confidence?

(Real) Storm Crushes Amazon Cloud, Knocks out Netflix, Pinterest, Instagram

BY ROBERT MCMILLAN 06.30.12 3:39 PM

According to the International Working Group on Cloud Computing Resiliency (IWGCR), the total downtime of 13 well-known cloud services since 2007 amounts to 568 hours, which has an economic impact of around $71.7 million dollars.

Why is it difficult to understand & predict behavior in complex information systems?

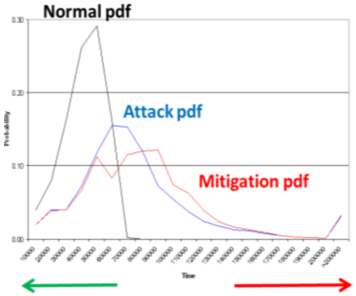
Reason #1: System state space is immense!!

$$\underbrace{y_1, \ldots, y_m}_{\text{Model Response Space}} \quad = \quad \underbrace{f(\, x_{1\,|\,[1,\ldots,k]}, \ldots, x_{n\,|\,[1,\ldots,k]} \,)}_{\text{Model Parameter Space}}$$

For example, the NIST ***Koala*** simulator of IaaS Clouds has about $n = 130$ parameters with average $k = 6$ values each, which leads to a model **parameter space** of **~10^{101}** (note that the visible universe has ~10^{80} atoms) and the ***Koala*** response space ranges from $m = 8$ to $m = 200$, depending on the specific responses chosen for analysis (typically $m \approx 45$).

Why is it difficult to understand & predict behavior in complex information systems?

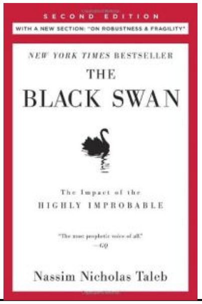
Reason #2: Emergent behaviors are difficult to predict!!



For example, deploying new client software with a reasonable approach to mitigate domain-name spoofing attacks in a grid system resulted in worse performance than ignoring the attacks, because mitigating the attacks shifted the global schedule of job executions.

Why is it difficult to understand & predict behavior in complex information systems?

Reason #3: Highly improbable events are more probable than we expect!!



Gaussian and Poissonian assumptions do not hold in complex systems. Instead, the probability landscape is better represented by heavy-tailed distributions, which means that highly improbably events occur more frequently than we assume. Such improbable events often lead to very expensive system-wide performance degradation or collapse.

Past Research 2006-2011

How can we understand the influence of distributed control algorithms on global system behavior and user experience?

INTERNET

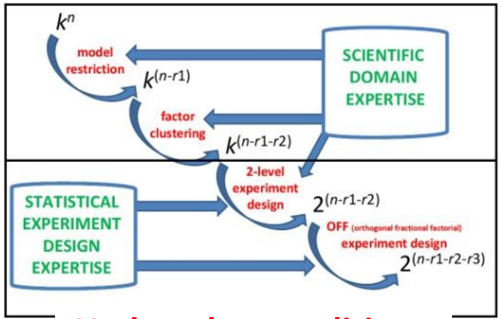
- Mills, Filliben, Cho, Schwartz and Genin, **Study of Proposed Internet Congestion Control Mechanisms**, NIST SP 500-282 (2010).
- Mills and Filliben, "**Comparison of Two Dimension-Reduction Methods for Network Simulation Models**", *Journal of NIST Research* 116-5, 771-783 (2011).
- Mills, Schwartz and Yuan, "**How to Model a TCP/IP Network using only 20 Parameters**", *Proceedings of the Winter Simulation Conference* (2010).
- Mills, Filliben, Cho and Schwartz, "**Predicting Macroscopic Dynamics in Large Distributed Systems**", *Proceedings of ASME* (2011).

IaaS CLOUDS

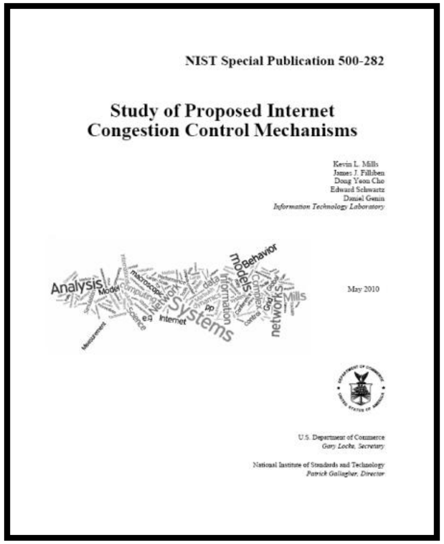
- Mills, Filliben and Dabrowski, "**An Efficient Sensitivity Analysis Method for Large Cloud Simulations**", *Proceedings of the 4th International Cloud Computing Conference*, IEEE (2011).
- Mills, Filliben and Dabrowski, "**Comparing VM-Placement Algorithms for On-Demand Clouds**", *Proceedings of IEEE CloudCom*, 91-98 (2011).

For more see: http://www.nist.gov/itl/antd/emergent_behavior.cfm

What to measure



Under what conditions


NIST Special Publication 500-282

Study of Proposed Internet
Congestion Control Mechanisms

Kevin L. Mills
James J. Filliben
Doug Yeon Cho
Edward Schwartz
Daniel Genin
Information Technology Laboratory

May 2010

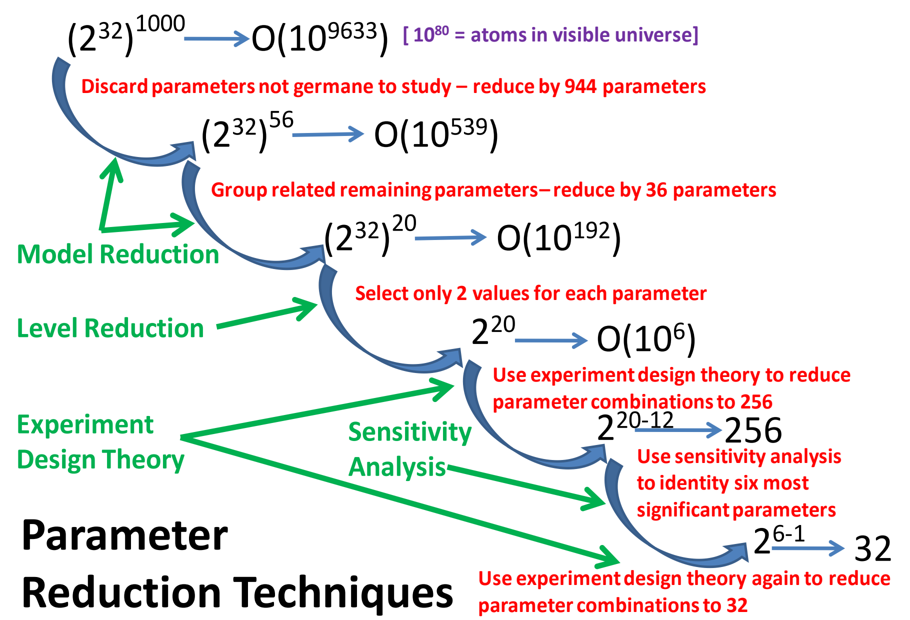
U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Patrick Gallagher, Director

http://www.nist.gov/itl/antd/Congestion_Control_Study.cfm

At an affordable cost

Sample Techniques



$(2^{32})^{1000} \rightarrow O(10^{9633})$ [10^{80} = atoms in visible universe]

Discard parameters not germane to study – reduce by 944 parameters

$(2^{32})^{56} \rightarrow O(10^{539})$

Group related remaining parameters– reduce by 36 parameters

$(2^{32})^{20} \rightarrow O(10^{192})$

Select only 2 values for each parameter

$2^{20} \rightarrow O(10^{6})$

Use experiment design theory to reduce parameter combinations to 256

$2^{20-12} \rightarrow 256$

Use sensitivity analysis to identity six most significant parameters

$2^{6-1} \rightarrow 32$

Use experiment design theory again to reduce parameter combinations to 32

Parameter Reduction Techniques

Response Reduction Techniques

We identified an **8-dimensional response space** within the 40 responses

Compute correlation coefficient (*r*) for all response pairs

Examine frequency distribution for all |*r*| to determine threshold for correlation pairs to retain; |*r*| > 0.65, here

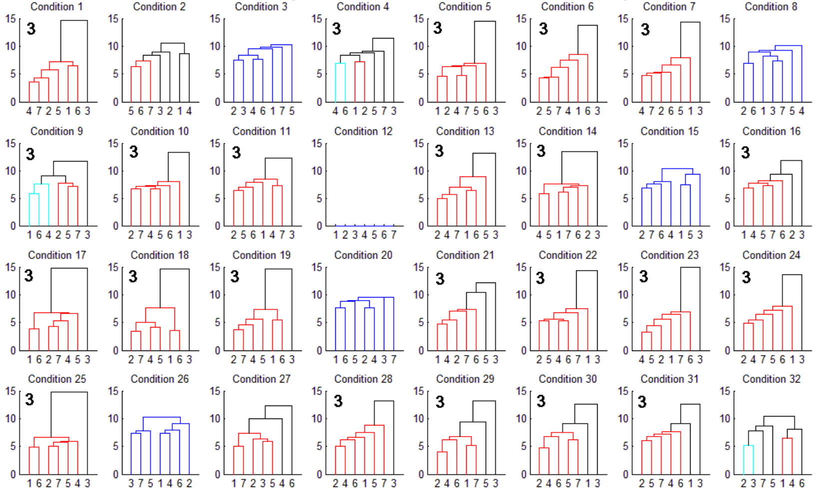
Create clusters of mutually correlated pairs; each cluster represents one dimension

Select one response from each cluster to represent the dimension; we selected response with largest mean correlation that was not in another cluster*

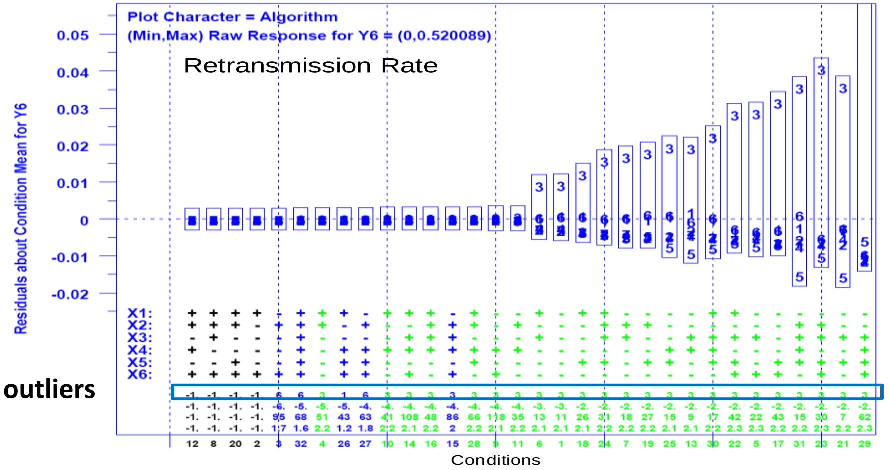
Response Dimension	SA1-small (9 dimensions)	SA1-large (8 dimensions)	SA2-small (10 dimensions)	SA2-large (9 dimensions)
Cloud-wide Demand/Supply Ratio	y1, y2, **y3**, y5, y6, y8, y9, y10, y13, y23, y24, y25, y29, y30, y32, y34, y36, y38	y1, y2, **y3**, y5, y6, y7, y8, y9, y10, y13, y23, y34, y25, y29, y30, y32, y33, y34, y36, y38	y1, **y2**, y3, y5, y6, y8, y9, y10, y11, y13, y14, y15, y23, y24, y25, y38	y1, y2, y3, y5, y6, y8, y9, **y23**, y24, y25, y38
Cloud-wide Resource Usage	y10, y11, y12, y13, y14, **y15**	y10, y11, y12, y13, y14, **y15**	**y10**, y11, y12, y13, y14, y15	**y10**, y11, y12, y13, y14, y15
Variance in Cluster Load	y16, y17, y18, y19,y20, y21, **y26**, y27	y16, y17, y18, y19,y20, y21, **y26**, y27	y16, y18, y19, y20, y21, y26, **y27** **y17** (Mem. Util)	y16, y17, y18, **y19**,y20, y21, y26, y27
Mix of VM Types	y34, **y35** (WS) **y31** (MS)	**y31** (MS)	y12, y14, y15, y30, y31, y33, y34, y35, **y36**	y14, y15, y30, **y31**, y33, y34, y35 y15, **y36** (DS)
Number of VMs	y29, **y37**	**y37**	y29, **y37**	**y29**
User Arrival Rate	**y4**	**y4**	**y4**	**y4**, y37
Reallocation Rate	**y7**, y22	y7, **y22**	**y7** (cluster) **y22** (node)	y7, **y22**
Variance in Choice of Cluster	**y28**	**y28**	**y28**	**y28**

*Not possible for cloud-wide resource usage in SA2-small, so we selected response with highest mean correlation.

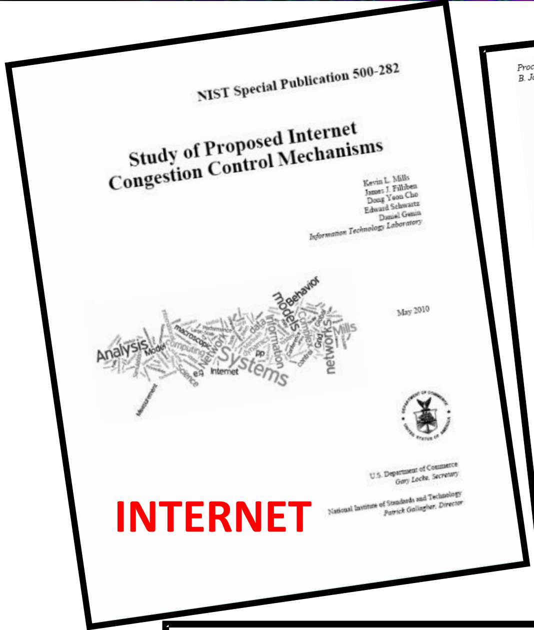
Cluster Analyses Over All Responses



Sorted Residual Analyses to Reveal Causality



Sample Results



NIST Special Publication 500-282

Study of Proposed Internet Congestion Control Mechanisms

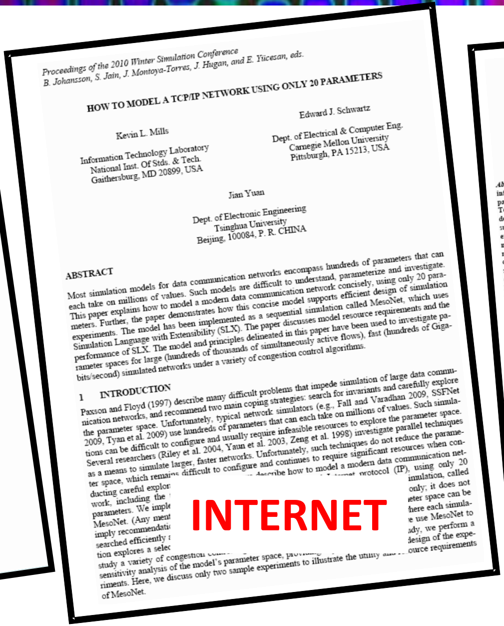
Kevin L. Mills
James J. Filliben
Dong Yeon Cho
Edward Schwartz
Daniel Genin
Information Technology Laboratory

May 2010

U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Patrick Gallagher, Director

INTERNET



Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

HOW TO MODEL A TCP/IP NETWORK USING ONLY 20 PARAMETERS

Kevin L. Mills
Information Technology Laboratory
National Inst. Of Stds. & Tech.
Gaithersburg, MD 20899, USA

Edward J. Schwartz
Dept. of Electrical & Computer Eng.
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Jian Yuan
Dept. of Electronic Engineering
Tsinghua University
Beijing, 100084, P. R. CHINA

ABSTRACT

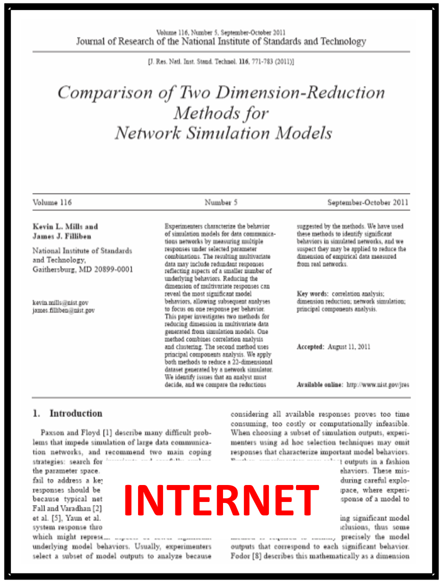
Most simulation models for data communication networks encompass hundreds of parameters that can each take on millions of values. Such models are difficult to understand, parameterize and investigate. This paper explains how to model a modern data communication network concisely, using only 20 parameters. Further, the paper demonstrates how this concise model supports efficient design of simulation experiments. The model has been implemented as a sequential simulation called MesoNet, which uses Simulation Language with Extensibility (SLX). The paper discusses model resource requirements and the performance of SLX. The model and principles delineated in this paper have been used to investigate parameter spaces for large (hundreds of thousands of simultaneously active flows), fast (hundreds of Gigabits/second) simulated networks under a variety of congestion control algorithms.

INTERNET

An Efficient Sensitivity Analysis Method for Large Cloud Simulations

K. Mills, J. Filliben and C. Dabrowski
Information Technology Laboratory
NIST
Gaithersburg, MD USA
{kmills, jfilliben, cdabrowski}@nist.gov

CLOUD



Volume 116, Number 5, September-October 2011
Journal of Research of the National Institute of Standards and Technology

[J. Res. Natl. Inst. Stand. Technol. 116, 771-783 (2011)]

Comparison of Two Dimension-Reduction Methods for Network Simulation Models

Kevin L. Mills and James J. Filliben
National Institute of Standards and Technology,
Gaithersburg, MD 20899-0001

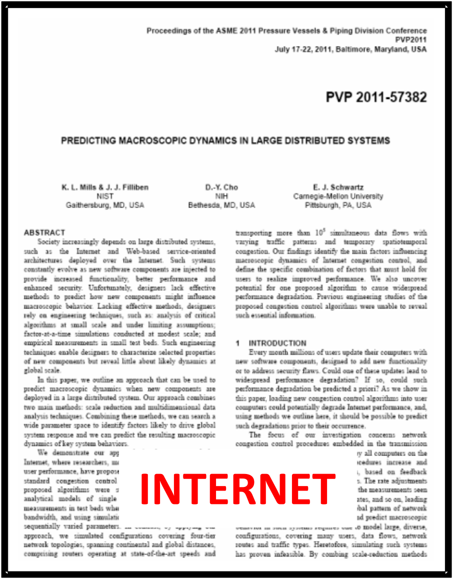
kevin.mills@nist.gov
james.filliben@nist.gov

Key words: correlation analysis; dimension reduction; network simulation; principal components analysis.

Accepted: August 11, 2011

Available online: http://www.nist.gov/jres

INTERNET



Proceedings of the ASME 2011 Pressure Vessels & Piping Division Conference
PVP2011
July 17-22, 2011, Baltimore, Maryland, USA

PVP 2011-57382

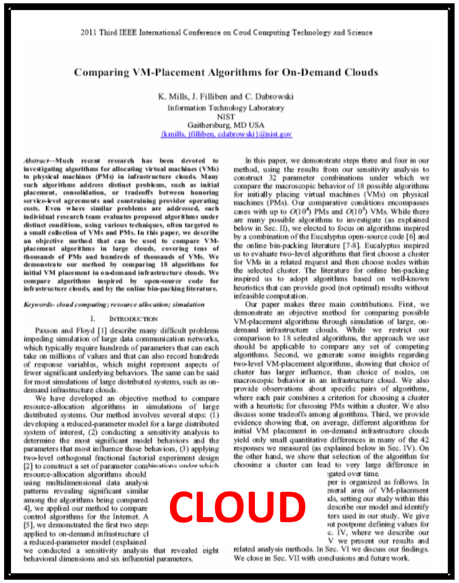
PREDICTING MACROSCOPIC DYNAMICS IN LARGE DISTRIBUTED SYSTEMS

K. L. Mills & J. J. Filliben
NIST
Gaithersburg, MD, USA

D.-Y. Cho
NIH
Bethesda, MD, USA

E. J. Schwartz
Carnegie-Mellon University
Pittsburgh, PA, USA

INTERNET

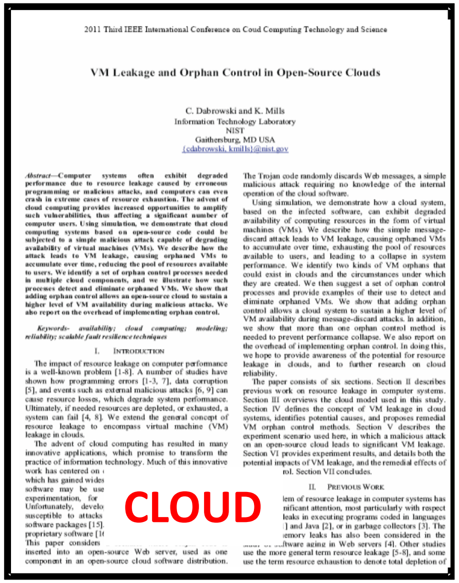


2011 Third IEEE International Conference on Coud Computing Technology and Science

Comparing VM-Placement Algorithms for On-Demand Clouds

K. Mills, J. Filliben, and C. Dabrowski
Information Technology Laboratory
NIST
Gaithersburg, MD USA
{kmills, jfilliben, cdabrowski}@nist.gov

CLOUD



2011 Third IEEE International Conference on Coud Computing Technology and Science

VM Leakage and Orphan Control in Open-Source Clouds

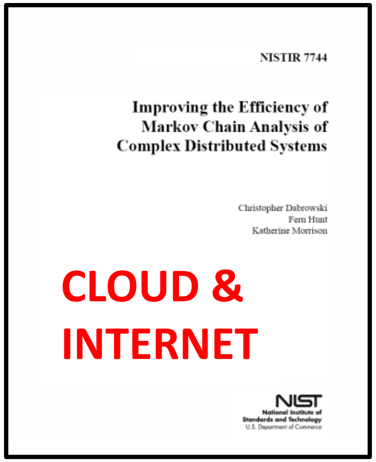
C. Dabrowski and K. Mills
Information Technology Laboratory
NIST
Gaithersburg, MD USA
{cdabrowski, kmills}@nist.gov

CLOUD

Ongoing Research 2011-

How can we increase the reliability of complex information systems?

- **Research Goals**: (1) develop and evaluate **design-time methods** that system engineers can use to detect existence and causes of costly failure regimes prior to system deployment and (2) develop and evaluate **run-time methods** that system managers can use to detect onset of costly failure regimes in deployed systems, prior to collapse.
- **Ongoing**: investigating **design-time methods** –
  a. **Markov Chain Modeling + Cut-Set Analysis + Perturbation Analysis** (e.g., Dabrowski, Hunt and Morrison, “Improving the Efficiency of Markov Chain Analysis of Complex Distributed Systems”, NIST IR 7744, 2010).
  b. **Anti-Optimization (AO) + Genetic Algorithm (GA)** **– example to be presented in some depth**



NISTIR 7744

Improving the Efficiency of
Markov Chain Analysis of
Complex Distributed Systems

Christopher Dabrowski
Fern Hunt
Katherine Morrison

CLOUD & INTERNET

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

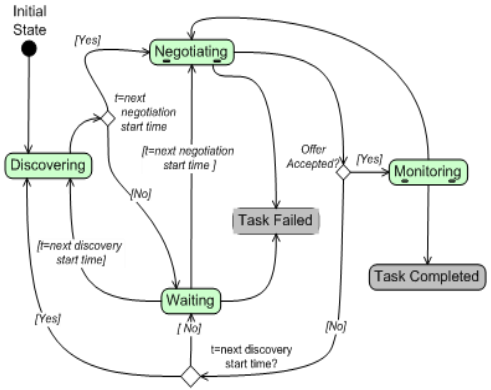
http://www.nist.gov/itl/antd/upload/NISTIR7744.pdf

- **Planned:** investigate **run-time methods** based on approaches that may provide early warning signals for critical transitions in large systems (e.g., Scheffer et al., “Early-warning signals for critical transitions”, *NATURE*, 461, 53-59, 2009).

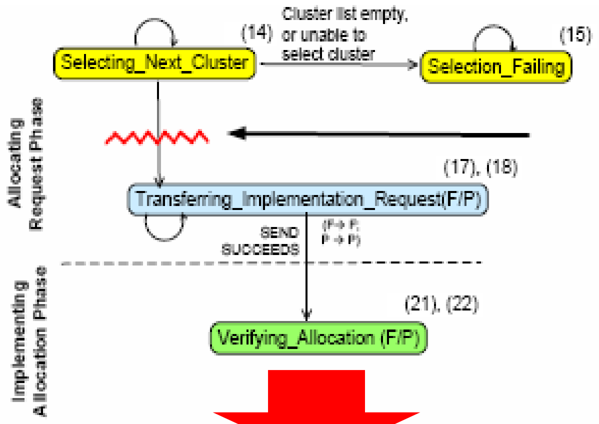
Design-Time Example One

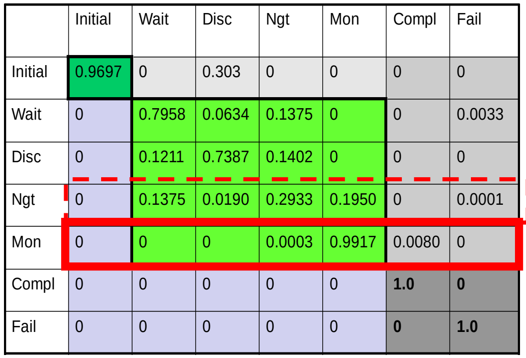
Example Markov Chains + Cut-set Analysis + Perturbation Analysis

EXTRACT FINITE-STATE MACHINE (FSM) FROM SIMULATION MODEL OR SYSTEM



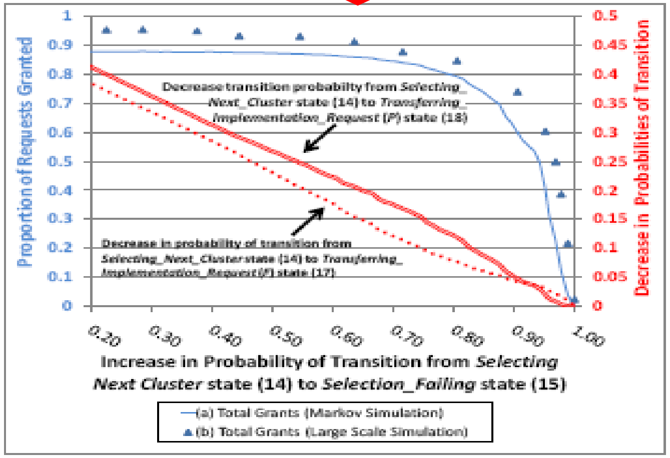
TREAT FSM AS GRAPH AND CONDUCT CUT-SET ANALYSIS





	Initial	Wait	Disc	Ngt	Mon	Compl	Fail
Initial	0.9697	0	0.303	0	0	0	0
Wait	0	0.7958	0.0634	0.1375	0	0	0.0033
Disc	0	0.1211	0.7387	0.1402	0	0	0
Ngt	0	0.1375	0.0190	0.2933	0.1950	0	0.0001
Mon	0	0	0	0.0003	0.9917	0.0080	0
Compl	0	0	0	0	0	**1.0**	**0**
Fail	0	0	0	0	0	**0**	**1.0**

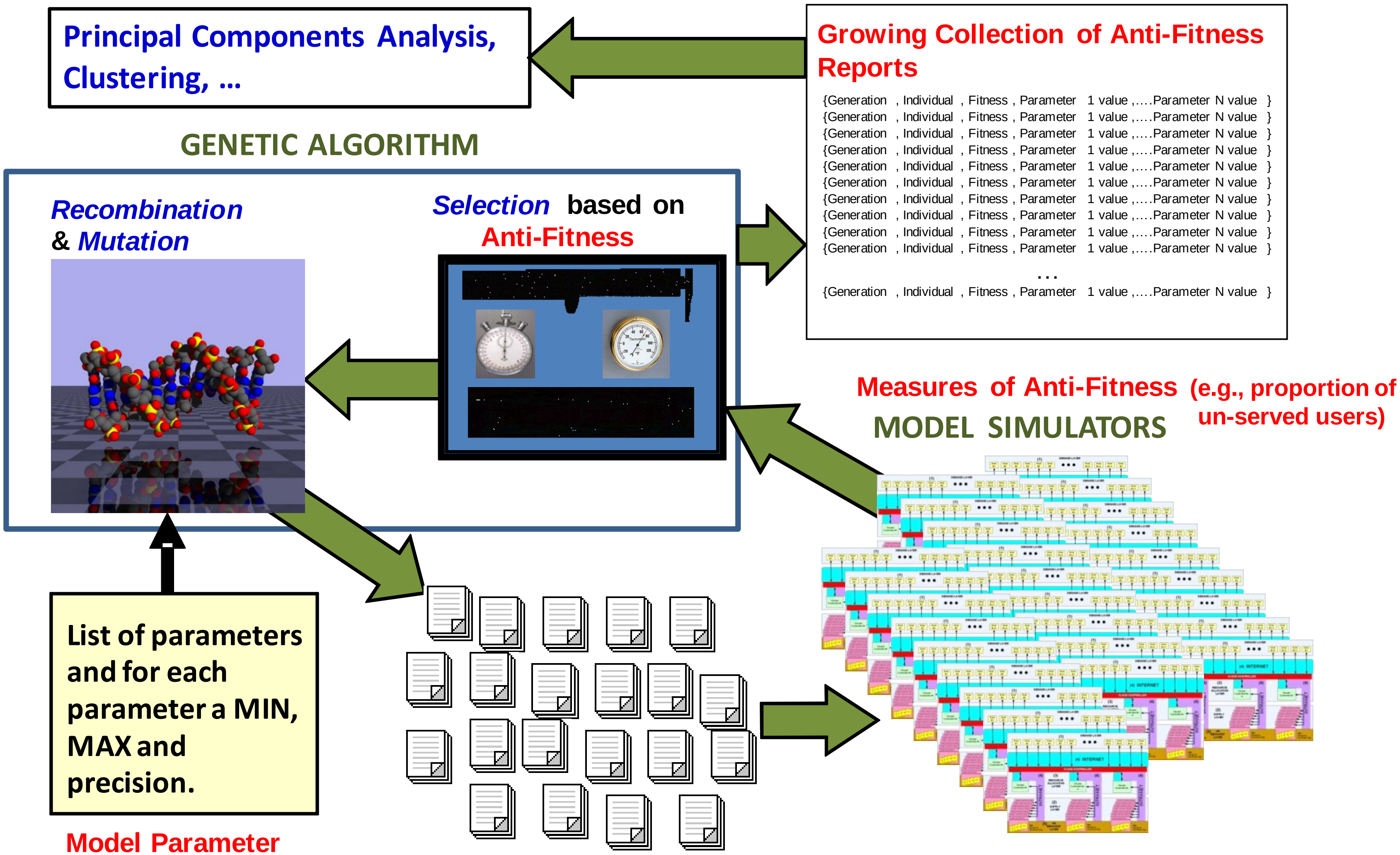
INSTRUMENT MODEL AND BUILD MARKOV CHAIN



PERTURB MARKOV CHAIN AT CUTS

Design-Time Example Two

Example: Anti-Optimization + Genetic Algorithm



MULTIDIMENSIONAL ANALYSIS TECHNIQUES

Principal Components Analysis, Clustering, …

Growing Collection of Anti-Fitness Reports

{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
…
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }

GENETIC ALGORITHM

Recombination **& *Mutation***

Selection **based on Anti-Fitness**

Measures of Anti-Fitness (e.g., proportion of un-served users)

MODEL SIMULATORS

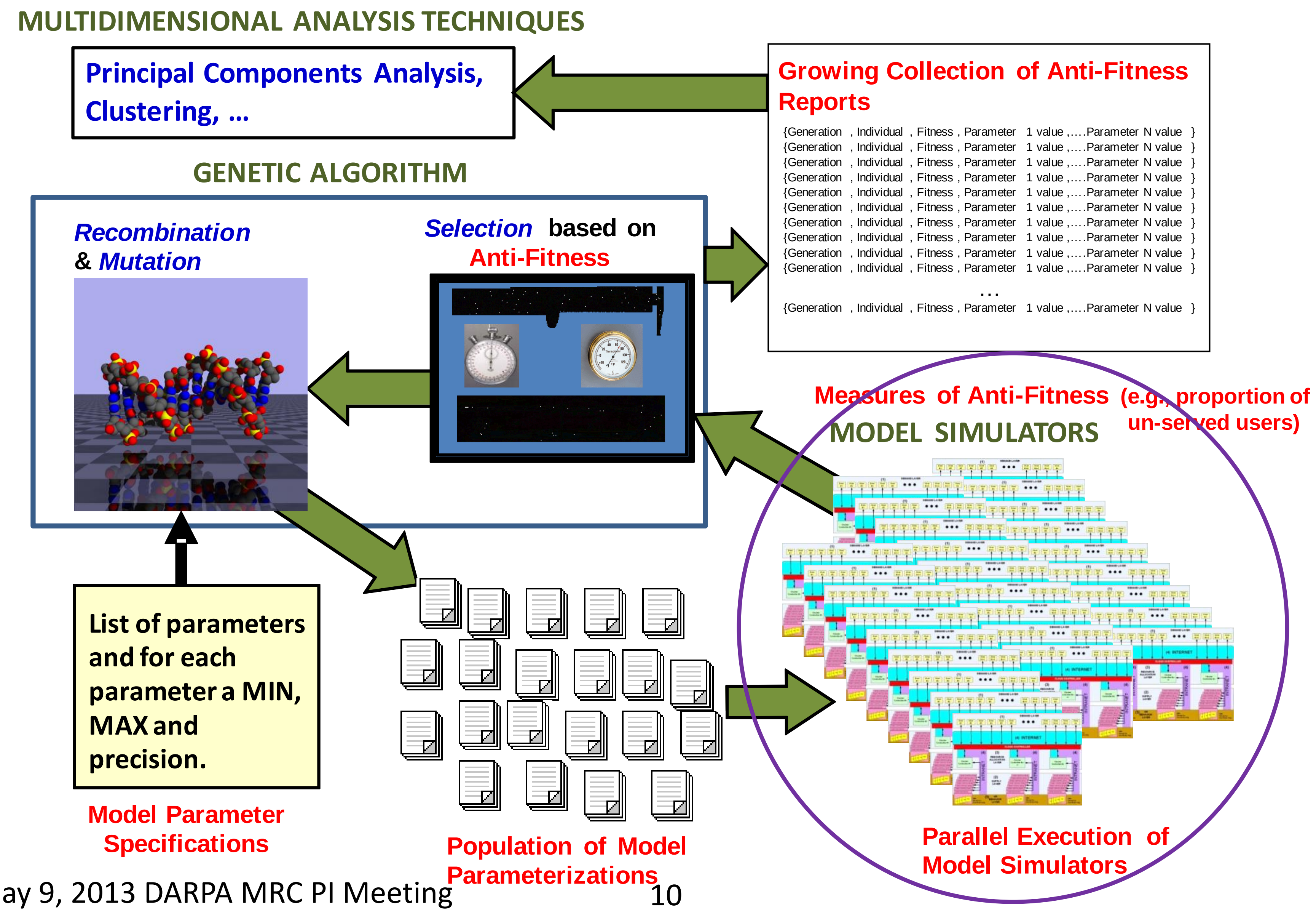
List of parameters and for each parameter a MIN, MAX and precision.

Model Parameter Specifications

Population of Model Parameterizations

Parallel Execution of Model Simulators

Design-Time Example Two



MULTIDIMENSIONAL ANALYSIS TECHNIQUES

Principal Components Analysis, Clustering, …

Growing Collection of Anti-Fitness Reports

{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
…
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }

GENETIC ALGORITHM

Recombination
& ***Mutation***

Selection **based on**
Anti-Fitness

Measures of Anti-Fitness **(e.g., proportion of un-served users)**

MODEL SIMULATORS

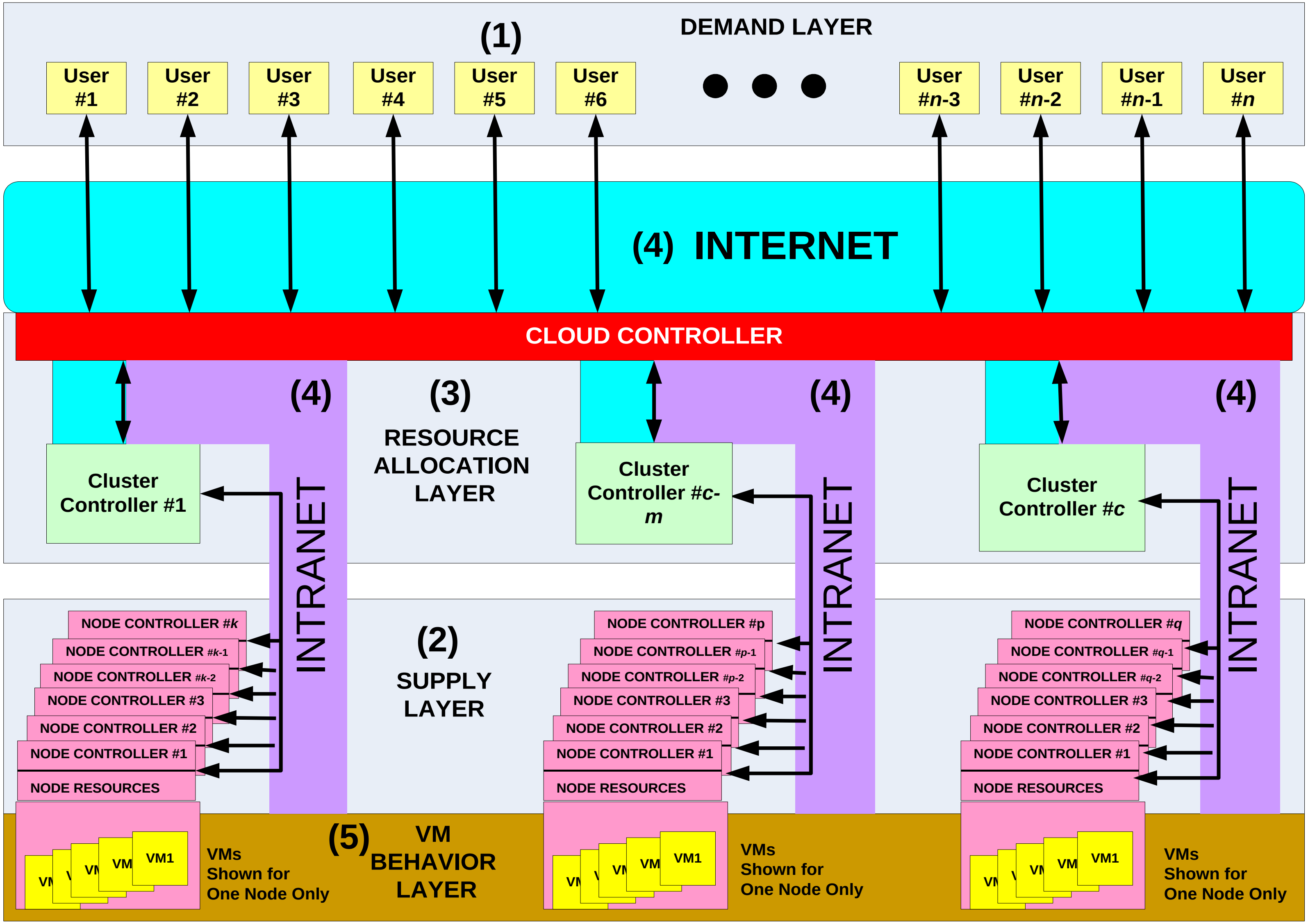
List of parameters and for each parameter a MIN, MAX and precision.

Model Parameter Specifications

Population of Model Parameterizations

Parallel Execution of Model Simulators

Schematic of *Koala* IaaS Cloud Computing Model



(1) DEMAND LAYER

User #1

User #2

User #3

User #4

User #5

User #6

● ● ●

User #*n*-3

User #*n*-2

User #*n*-1

User #*n*

(4) INTERNET

CLOUD CONTROLLER

(4)

(3) RESOURCE ALLOCATION LAYER

Cluster Controller #1

Cluster Controller #*c-m*

Cluster Controller #*c*

INTRANET

(2) SUPPLY LAYER

NODE CONTROLLER #*k*

NODE CONTROLLER #*k*-1

NODE CONTROLLER #*k*-2

NODE CONTROLLER #3

NODE CONTROLLER #2

NODE CONTROLLER #1

NODE RESOURCES

NODE CONTROLLER #*p*

NODE CONTROLLER #*p*-1

NODE CONTROLLER #*p*-2

NODE CONTROLLER #*q*

NODE CONTROLLER #*q*-1

NODE CONTROLLER #*q*-2

(5) VM BEHAVIOR LAYER

VM1

VMs Shown for One Node Only

(2) SUPPLY LAYER

Virtual Machine (VM) Types Simulated in *Koala*

VM Type	Virtual Cores		Virtual Block Devices		# Virtual Network Interfaces	Memory (GB)	Instruct. Arch.
	#	Speed (GHz)	#	Size (GB) of Each			
M1 small	1	1.7	1	160	1	2	32-bit
M1 large	2	2	2	420	2	8	64-bit
M1 xlarge	4	2	4	420	2	16	64-bit
C1 medium	2	2.4	1	340	1	2	32-bit
C1 xlarge	8	2.4	4	420	2	8	64-bit
M2 xlarge	8	3	1	840	2	32	64-bit
M4 xlarge	8	3	2	850	2	64	64-bit

Four of 22 Physical Platform Types Simulated in *Koala*

Platform Type	Physical Cores		Memory (GB)	# Physical Disks by Size				# Network Interfaces	Instruct. Arch.
	#	Speed (GHz)		250 GB	500 GB	750 GB	1000 GB		
C8	2	2.4	32	0	3	0	0	1	64-bit
C14	4	3	64	0	4	0	3	2	64-bit
C18	8	3	128	0	0	4	3	4	64-bit
C22	16	3	256	0	0	0	7	4	64-bit

(1) DEMAND LAYER

Description of User Types Simulated in *Koala*

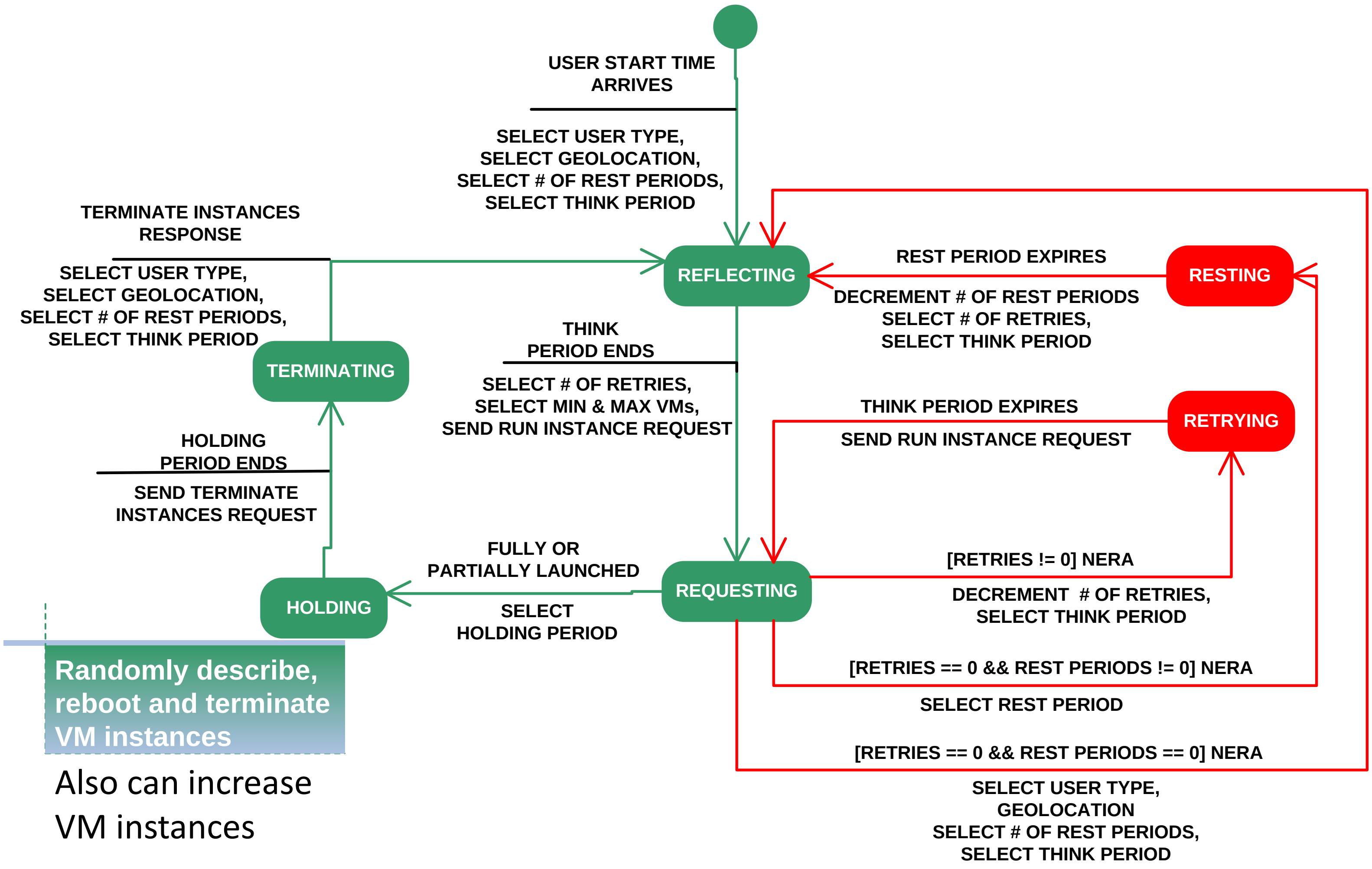
We created different classes of demand, such as processing users (PU), distributed simulation users (MS), peer-to-peer users (PS), Web service users (WS) and data search users (DS)

User Type	VM Type(s)	Max-Min VMs	Max-Max VMs
PU1	M1 small	10	100
PU3		100	500
PU5		500	1000
PU2	M1 large	10	100
PU4		100	500
PU6		500	1000
MS1	M1 xlarge	10	100
MS3		100	500

User Type	VM Type(s)	Max-Min VMs	Max-Max VMs
PS1	C1 medium	3	10
PS2		10	50
PS3		50	100
WS1	M1 large M2 xlarge C1 xlarge	1	3
WS2	M1 large M2 xlarge C1 xlarge	3	9
WS3	M1 large M2 xlarge C1 xlarge	9	12
DS1	M4 xlarge	10	100
DS2		100	500
DS3		500	1000

(1) DEMAND LAYER - USER BEHAVIOR

Finite-State Machine of Simulated User Behavior in *Koala*



USER START TIME ARRIVES

SELECT USER TYPE,
SELECT GEOLOCATION,
SELECT # OF REST PERIODS,
SELECT THINK PERIOD

REFLECTING

REST PERIOD EXPIRES

DECREMENT # OF REST PERIODS
SELECT # OF RETRIES,
SELECT THINK PERIOD

RESTING

THINK PERIOD ENDS

SELECT # OF RETRIES,
SELECT MIN & MAX VMs,
SEND RUN INSTANCE REQUEST

THINK PERIOD EXPIRES

SEND RUN INSTANCE REQUEST

RETRYING

TERMINATE INSTANCES RESPONSE

SELECT USER TYPE,
SELECT GEOLOCATION,
SELECT # OF REST PERIODS,
SELECT THINK PERIOD

TERMINATING

HOLDING PERIOD ENDS

SEND TERMINATE INSTANCES REQUEST

HOLDING

FULLY OR PARTIALLY LAUNCHED

SELECT HOLDING PERIOD

REQUESTING

[RETRIES != 0] NERA

DECREMENT # OF RETRIES,
SELECT THINK PERIOD

[RETRIES == 0 && REST PERIODS != 0] NERA

SELECT REST PERIOD

[RETRIES == 0 && REST PERIODS == 0] NERA

SELECT USER TYPE,
GEOLOCATION
SELECT # OF REST PERIODS,
SELECT THINK PERIOD

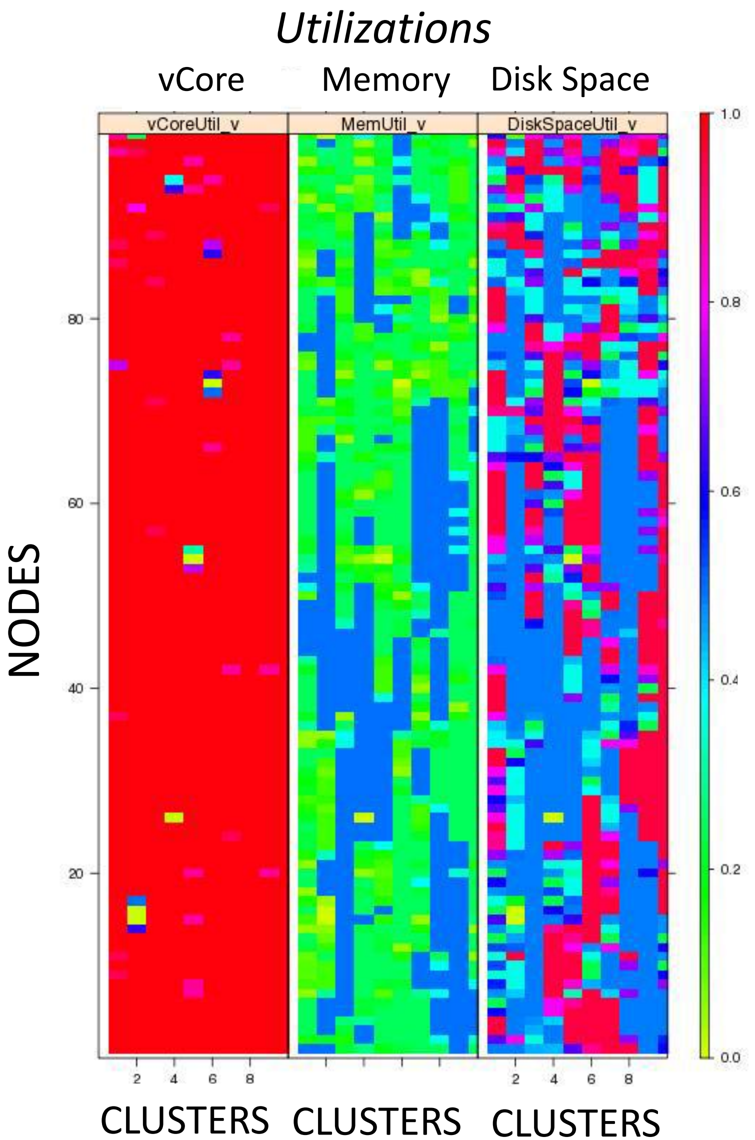
Randomly describe, reboot and terminate VM instances

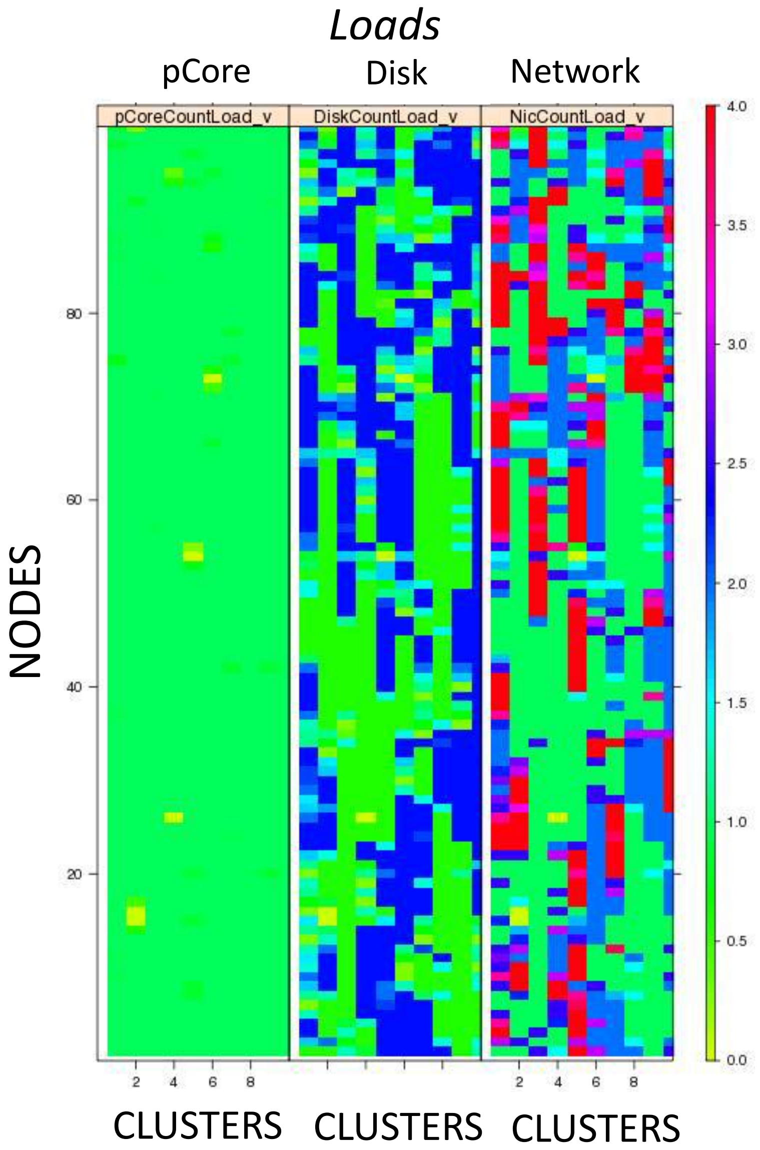
Also can increase VM instances

SYSTEM BEHAVIOR – ENCOMPASSING LAYERS (1) – (4)

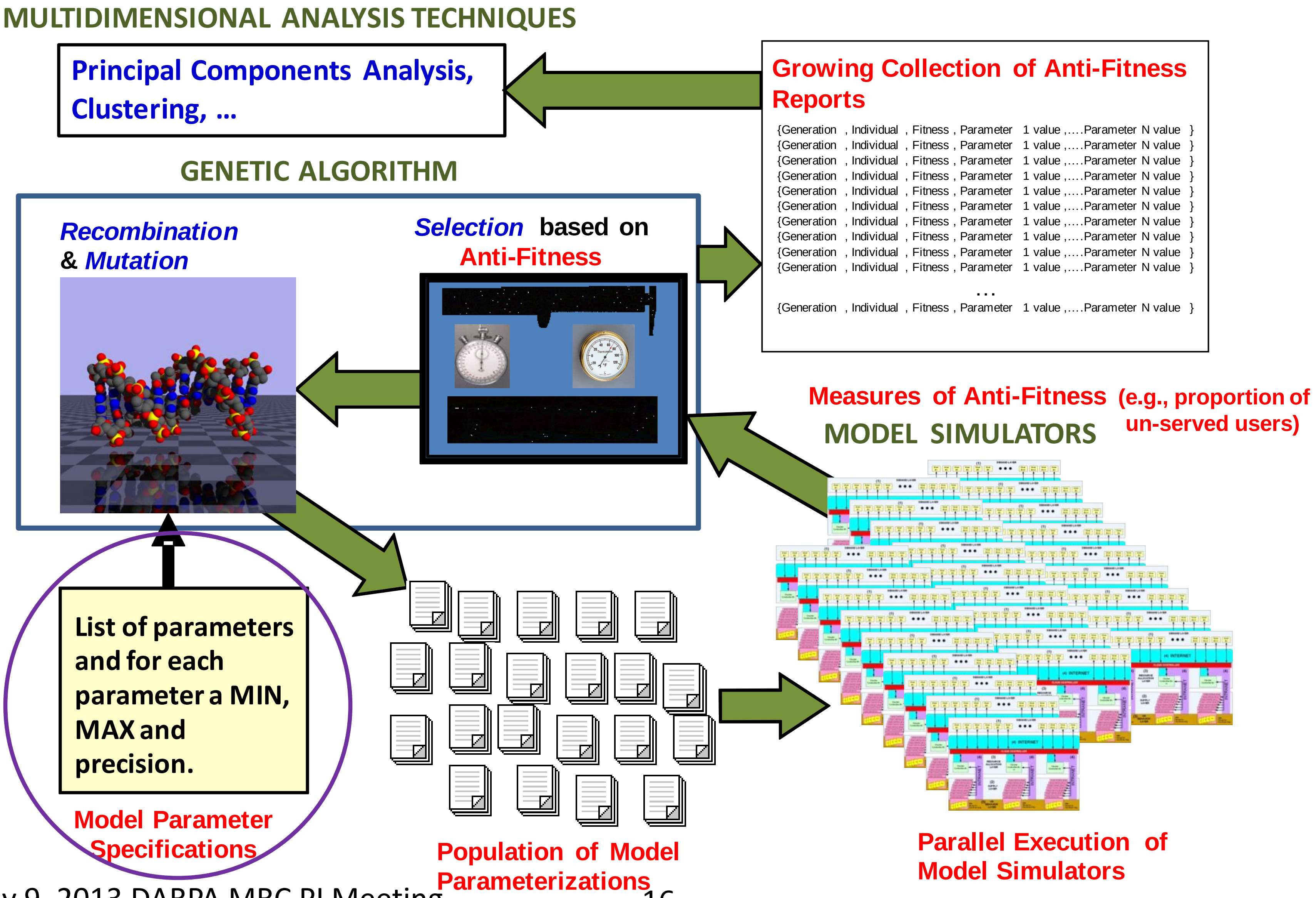
Snapshot of Simulated Cloud State from a *10-D Koala Animation*

(Heat Value for 6 Metrics for each of 10 Clusters x 100 Nodes/Cluster after 90 Hours)





Design-Time Example Two



MULTIDIMENSIONAL ANALYSIS TECHNIQUES

Principal Components Analysis, Clustering, …

Growing Collection of Anti-Fitness Reports

{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
…
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }

GENETIC ALGORITHM

Recombination* & *Mutation

***Selection* based on Anti-Fitness**

Measures of Anti-Fitness (e.g., proportion of un-served users)

MODEL SIMULATORS

List of parameters and for each parameter a MIN, MAX and precision.

Model Parameter Specifications

Population of Model Parameterizations

Parallel Execution of Model Simulators

Summary of ***Koala*** Parameters to Search Over

Test Case – Can GA find VM Leakage due to message loss and lack of orphan control?

Failure scenario found manually by accident and described in C. Dabrowski and K. Mills, "VM Leakage and Orphan Control in Open-Source Clouds", *Proceedings of IEEE CloudCom 2011*, Nov. 29-Dec. 1, Athens, Greece, pp. 554-559.

Model Element	Parameter Category				
	Behavior	**Structure**	**Failure**	**Asymmetry**	**Total**
User	28	2	0	4	34
Cloud Controller	21	4	0	5	30
Cluster Controllers	11	5	0	3	19
Nodes	6	0	14	0	20
Intra-Net/Inter-Net	4	11	9	2	26
Totals	70	22	23	14	129

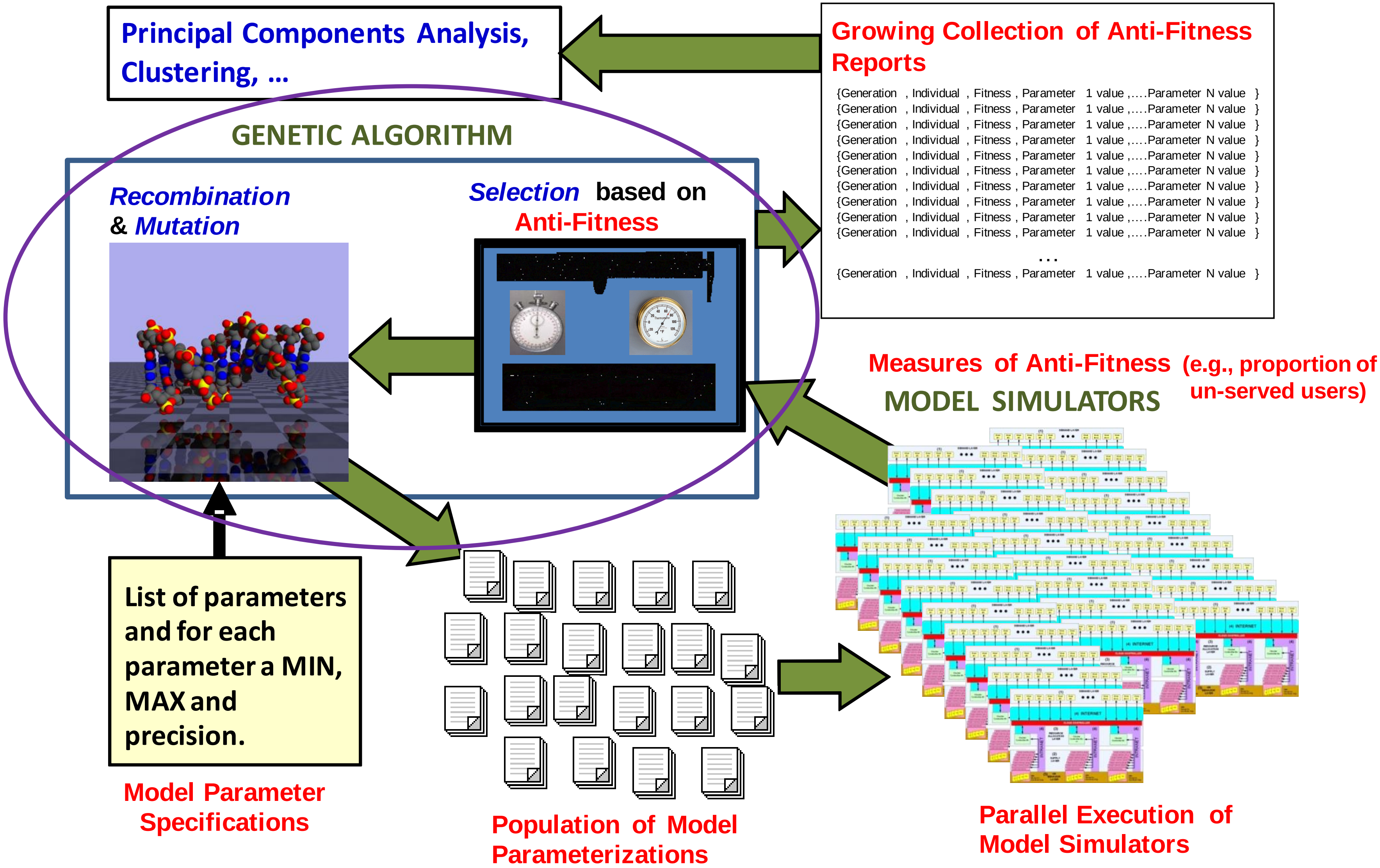
Average # values per parameter is about 6, so search space is ≈ 6^{129}
i.e., ≈ 10^{100} scenarios are possible

- **adapted 125-parameter Koala IaaS simulator to be GA controllable**
- **added 4 *Koala* parameters to turn on/off logic to control (a) creation orphans, (b) termination orphans, (c) relocation orphans and (d) administrator actions**

Sample Chromosome Specification

	Koala Parameter Space (Size = 10^{100})			Genetic Algorithm Computed Chromosome Map (Size = 2^{334})			
PARAMETER	MIN	MAX	PRECISION	#VALUES	LOW_BIT	HIGH_BIT	#BITS
P_CreateOrphanControlOn	0	1	1	2	36	36	1
P_TerminationOrphanControlOn	0	1	1	2	58	58	1
P_RelocationOrphanControlOn	0	1	1	2	11	11	1
P_AdministratorActive	0	1	1	2	330	330	1
P_clusterAllocationAlgorithm	0	5	1	6	31	33	3
P_describeResourcesInterval	600	3600	600	6	81	83	3
P_nodeResponseTimeout	30	90	30	3	210	211	2
P_TerminatedInstancesBackOffThreshold	3	6	1	4	56	57	2
P_TerminationBackOffInterval	180	360	60	4	88	89	2
P_TerminationRetryPeriod	600	1200	300	3	316	317	2
P_StaleShadowAllocationPurgeInterval	600	3600	600	6	242	244	3
P_cloudAllocationCriteria	0	3	1	4	321	322	2
P_clusterShadowPurgeLimit	1	21	5	5	290	292	3
P_instancePurgeDelay	180	600	60	8	98	100	3
P_clusterEvaluationResponseTimeout	60	120	30	3	14	15	2
P_MaxPendingRequests	1	10	1	10	72	75	4
P_CloudTerminatedInstancesBackOffThreshold	3	6	1	4	169	170	2
P_CloudTerminationBackOffInterval	180	360	60	4	40	41	2
P_CloudTerminationRetryPeriod	3600	10800	1800	5	297	299	3
P_ClusterShutdownGracePeriod	86400	2.59E+05	43200	5	147	149	3
●	●	●	●	●	●	●	●
P_RequestEvaluatorTimeoutWaitProportion	0.1	0.4	0.1	4	145	146	2
P_RequestEvaluatorClusterMinimumResponse	0.6	0.9	0.1	3	269	270	2
P_MaxRelocationDuratonProportion	0.65	0.95	0.1	4	90	91	2
P_MaximumRelocateDescribeRetries	4	16	2	7	254	256	3
P_AverageCloudAdministratorAttentionLatency	28800	86400	14400	5	308	310	3
P_AverageCloudAdministratorShutdownDelay	300	900	300	3	45	46	2
P_avgTimeToClusterCommunicationCut	2.88E+06	2.88E+07	2.88E+06	10	217	220	4

Design-Time Example Two



MULTIDIMENSIONAL ANALYSIS TECHNIQUES

Principal Components Analysis, Clustering, …

Growing Collection of Anti-Fitness Reports

{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }
…
{Generation , Individual , Fitness , Parameter 1 value ,….Parameter N value }

GENETIC ALGORITHM

Recombination & ***Mutation***

Selection **based on** **Anti-Fitness**

Measures of Anti-Fitness **(e.g., proportion of un-served users)**

MODEL SIMULATORS

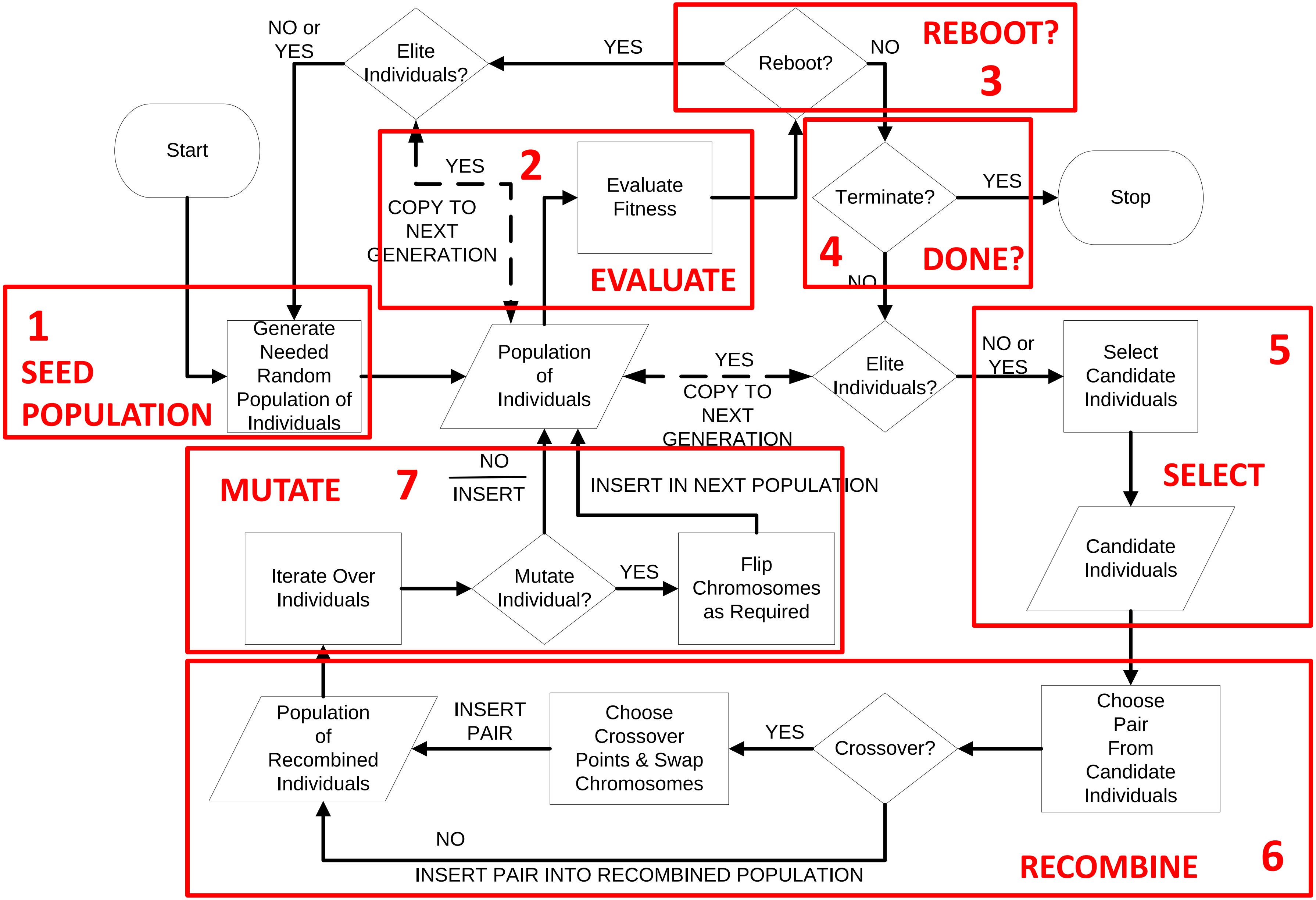
List of parameters and for each parameter a MIN, MAX and precision.

Model Parameter Specifications

Population of Model Parameterizations

Parallel Execution of Model Simulators

Genetic Algorithm Flow Chart



NO or YES

Elite Individuals?

YES

Reboot?

NO

REBOOT?

3

Start

YES

2

COPY TO NEXT GENERATION

Evaluate Fitness

EVALUATE

Terminate?

YES

Stop

4

DONE?

NO

1

SEED POPULATION

Generate Needed Random Population of Individuals

Population of Individuals

YES

COPY TO NEXT GENERATION

Elite Individuals?

NO or YES

Select Candidate Individuals

5

SELECT

Candidate Individuals

MUTATE

7

NO

INSERT

INSERT IN NEXT POPULATION

Iterate Over Individuals

Mutate Individual?

YES

Flip Chromosomes as Required

Population of Recombined Individuals

INSERT PAIR

Choose Crossover Points & Swap Chromosomes

YES

Crossover?

Choose Pair From Candidate Individuals

NO

INSERT PAIR INTO RECOMBINED POPULATION

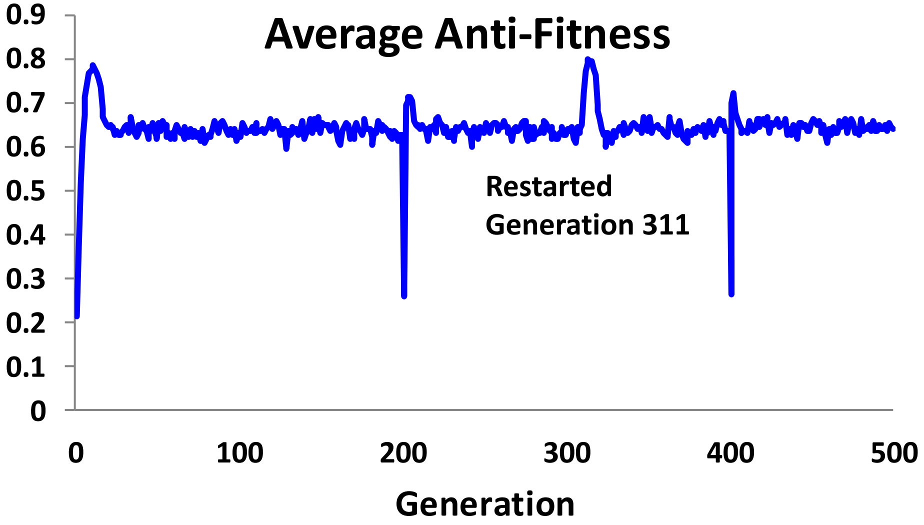
RECOMBINE

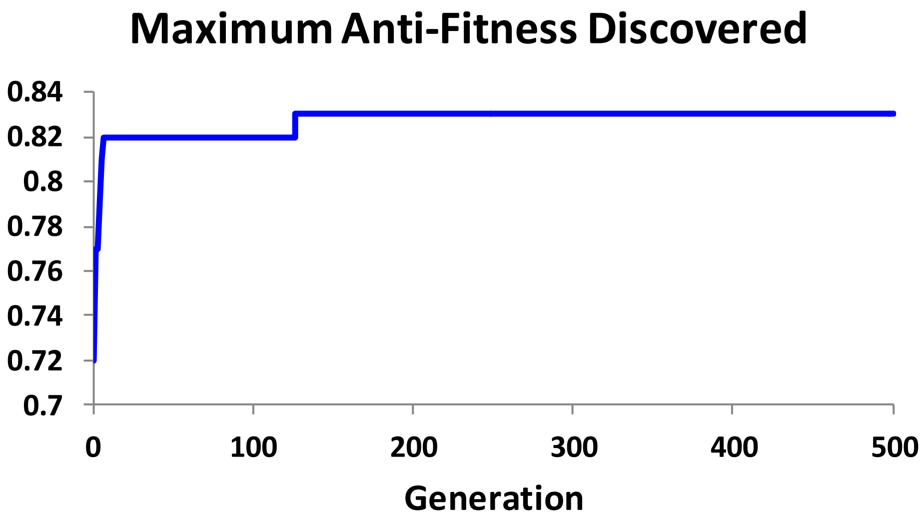
6

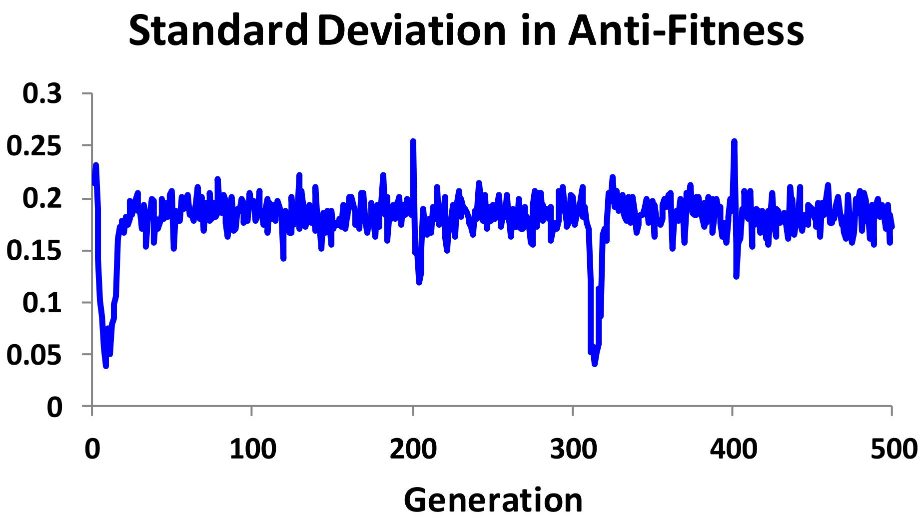
Dynamics of GA's Search

GENETIC ALGORITHM CONTROL PARAMETERS

Generations	500
Population Size	200 Individuals
Elite Per Generation	16 Individuals
Reboot After	200 Generations
Selection Method	Stochastic Uniform Sampling
# Crossover Points	3
Mutation Rate	0.001 ≤ Adaptive ≤ 0.01

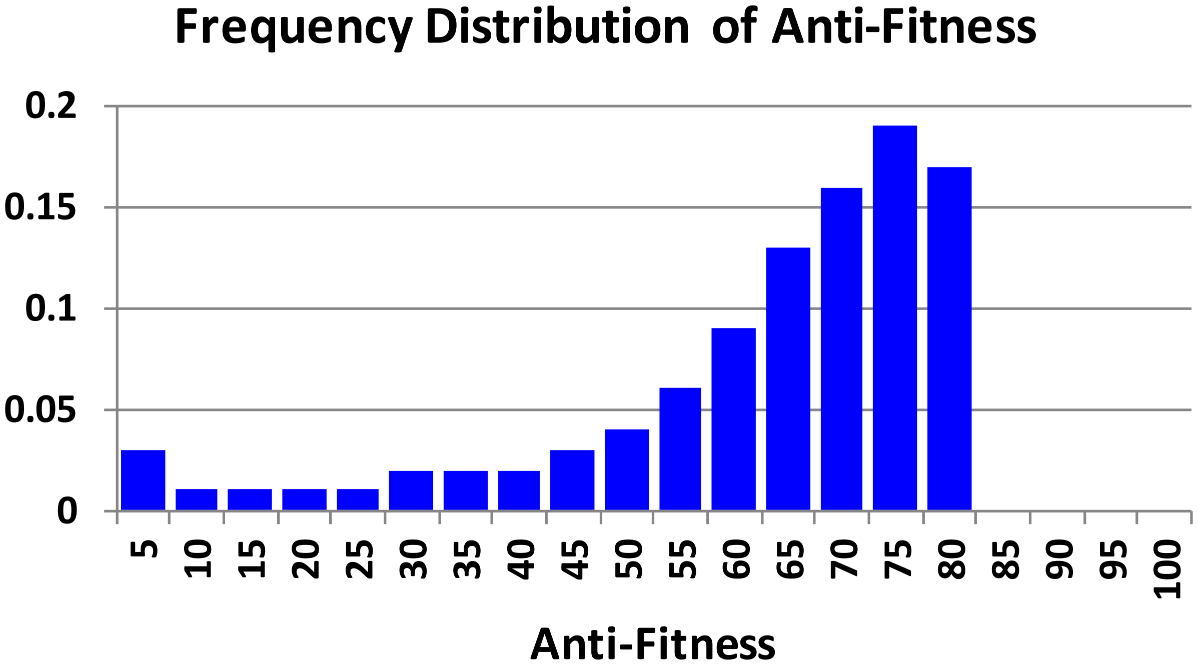






Assessment of Search Conducted by GA

(based on 10^5 scenarios, i.e., 200 individuals x 500 generations)

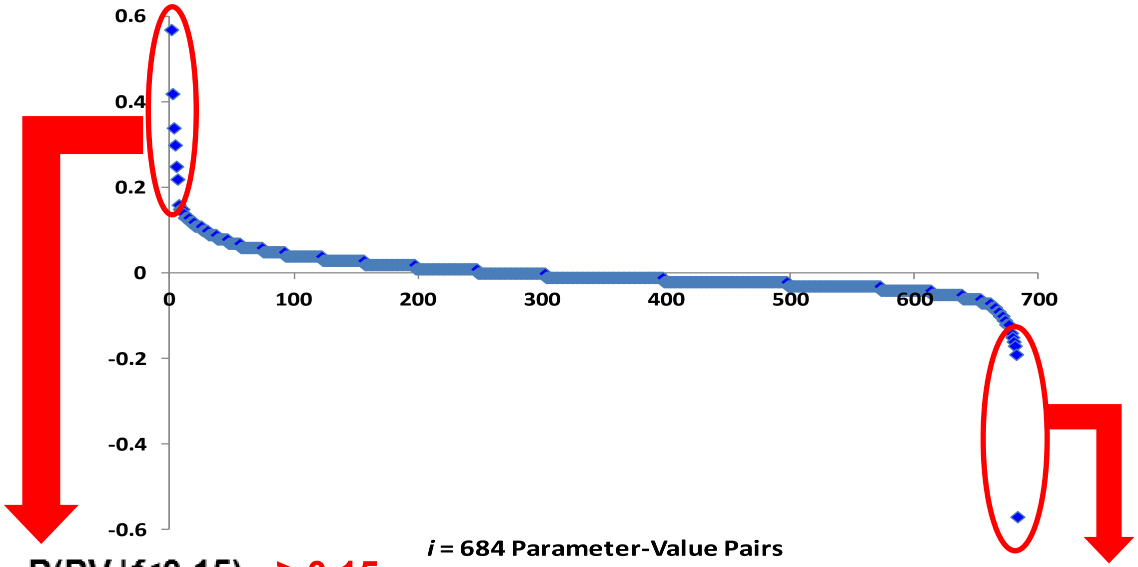


- 84% of scenarios exhibit anti-fitness ≥ 0.50
- Only 8% of scenarios are duplicate (equals elite-selection percentage)

Conclusion: GA is searching scenarios with high anti-fitness and the scenarios searched are overwhelmingly unique

Failure Scenarios Discovered by GA

$P(PV_i|f>0.70) - P(PV_i|f<0.15)$



$P(PV_i|f>0.70) - P(PV_i|f<0.15)$ **≥ 0.15**

Parameter		Value	Difference
P_CreateOrphanControlOn	**Statistically Significant**	0	0.58
P_averageUserRequestTimeout		30	0.42
P_averageThinkTime		900	0.33
P_nodesPerCluster		200	0.31
P_userRestPeriodMultiplier		32	0.25
P_nodesPerCluster		400	0.2
P_MinMaxInstanceLoad	**Not Statistically Significant**	0	0.16
P_maximumRestPeriods		6	0.15
P_minimumReservationRetries		2	0.15

$P(PV_i|f>0.70) - P(PV_i|f<0.15)$ **≤ -0.15**

P_averageThinkTime	1500	-0.15
P_MinMaxInstanceLoad	1	-0.16
P_clusterAllocationAlgorithm	2	-0.18
P_averageUserRequestTimeout	120	-0.19
P_CreateOrphanControlOn	1	-0.58

GA also found that an overload problem arises when clusters are too small

GA discovered that lack of orphan control leads to system failure, conditioned on user request timeouts being too short, which causes virtual message losses

Costs of Search

- Pre-search work required **significant programming effort** to
  - Increase cloud simulator robustness
  - Create robust distributed management system for GA and simulations running a cluster
- **Computing resources** used
  - Generation one: **200 cores** on a local cluster
  - Subsequent generations: 184 cores on a local cluster (16 cores acting as warm standbys to take over for failed simulations)
- **Search latency** about **30 days** (as designed) for **500 Generations**
- **Failure scenarios** are **evident** within **100 Generations,** which requires about **6 days**

Run-Time Research Goals

Problem: Catastrophic events manifest over extended space & time, e.g., congestion, attacks, cascading failures, disconnections

State-of-the-Art:

	Academia	Industry
Pro	**Models + Theory**	**Monitoring for Real Networks**
Con	**Abstract Models**	**Reactive, No Models/No Theory**

New Ideas: (1) Identify precursor S/T patterns in our realistic net models
(2) Assess detection techniques for applicability to real nets
(3) Apply thermodynamic models & theory to explain catastrophic events

Impact, If Successful:

Iraj Saniee, Bell Labs: "…the proposed research would help fill a vacuum in commercial network control and management systems…"

Craig Lee, Aerospace: "This line of work must be pursued, and its results used to shape satellite ground systems of the future."

David Lambert, Internet 2: "…will create a strong foundation of system measurement that has not existed before that is likely to help avoid potentially debilitating real-life network failures and their scientific and economic consequences."

Take Home Messages

If you want to investigate the robustness of your innovative MRC algorithms, NIST would be interested in collaborating to apply and evaluate techniques to identify design-time failure scenarios.

If you are interested in exploring run-time methods to predict incipient systemic failures, NIST has interest in collaborating on that topic.

If you want to evaluate your MRC algorithms in large simulated deployment scenarios, NIST has expertise in relevant techniques – and would be willing to help apply them.

Contact Information

Additional Questions?

Kevin Mills, NIST
Principal Investigator
Measurement Science for Complex Information Systems

Email: kmills@nist.gov
Web: http://www.nist.gov/itl/antd/emergent_behavior.cfm