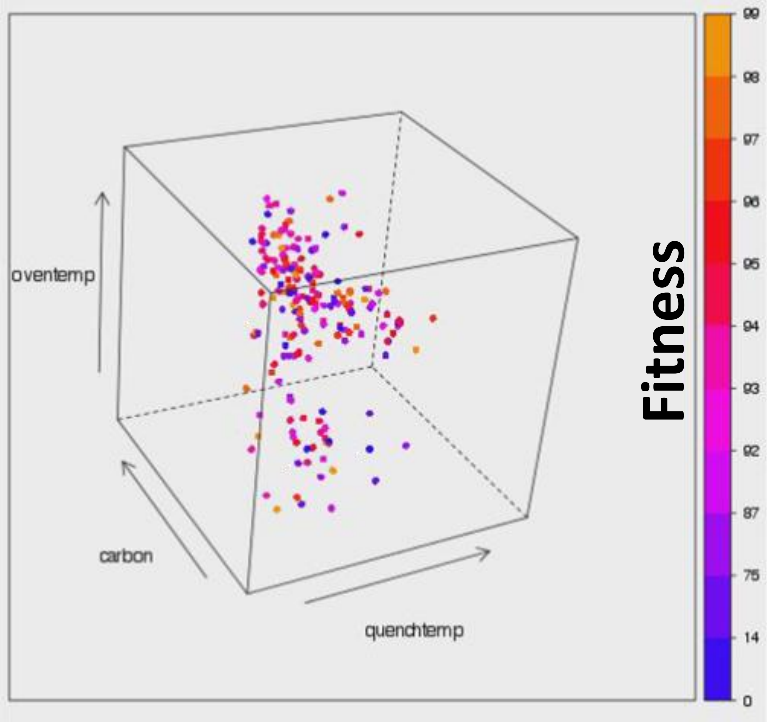
Improving Cloud Reliability

(and reliability for complex information systems in general)

June 5-7, 2012 NIST Cloud Forum **Kevin Mills**, Jim Filliben and Chris Dabrowski

Image shows one frame from a 5-Dimensional animation of a Genetic Algorithm (GA) searching for an optimal combination of oven temperature, quench temperature and carbon concentration in a production process, where fitness is measured as the percentage of non-defective springs produced.



Koala Information Visualizations by Sandy Ressler

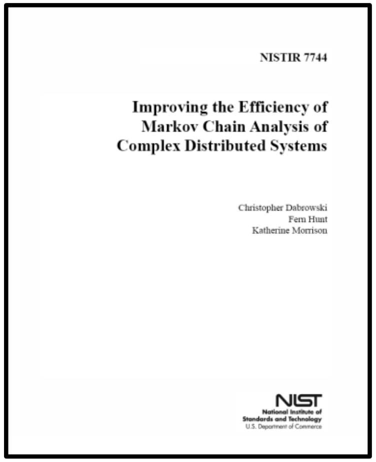
(see http://math.nist.gov/~SRessler/cloudviz.html for animations and more)

The Big Picture

- **Ongoing & Planned ITL Research**: How can we help to increase the reliability of complex information systems?

- **Research Goals**: (1) develop **design-time methods** that system engineers can use to detect existence and causes of costly failure regimes prior to system deployment and (2) develop **run-time methods** that system managers can use to detect onset of costly failure regimes in deployed systems, prior to collapse.

- **Ongoing**: investigating two 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. **Sensitivity Analysis + Anti-Optimization + Genetic Algorithm** – **subject of this talk**


NISTIR 7744

Improving the Efficiency of
Markov Chain Analysis of
Complex Distributed Systems

Christopher Dabrowski
Fern Hunt
Katherine Morrison

NIST

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).

National Institute of Standards and Technology

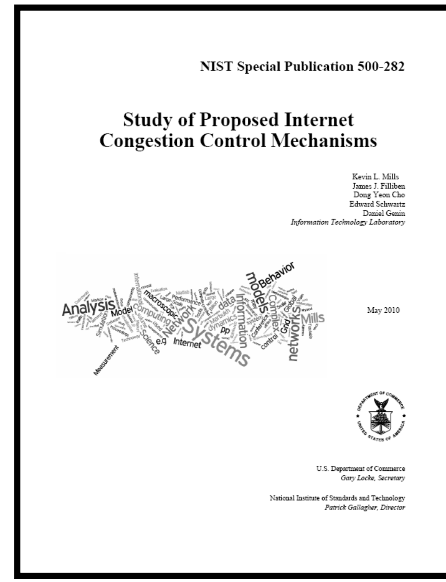
The Bigger Picture

NIST National Institute of Standards and Technology

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

- 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).
- 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


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

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

Today's Topics

- What is the problem and why is it hard?
- How might Sensitivity Analysis + **Anti-Optimization + Genetic Algorithm** address the problem?
- What is the current state of the project?
- What progress is expected over the next six months?
- How might your organization benefit from collaborating with us?
- What other actions might help to improve cloud reliability?

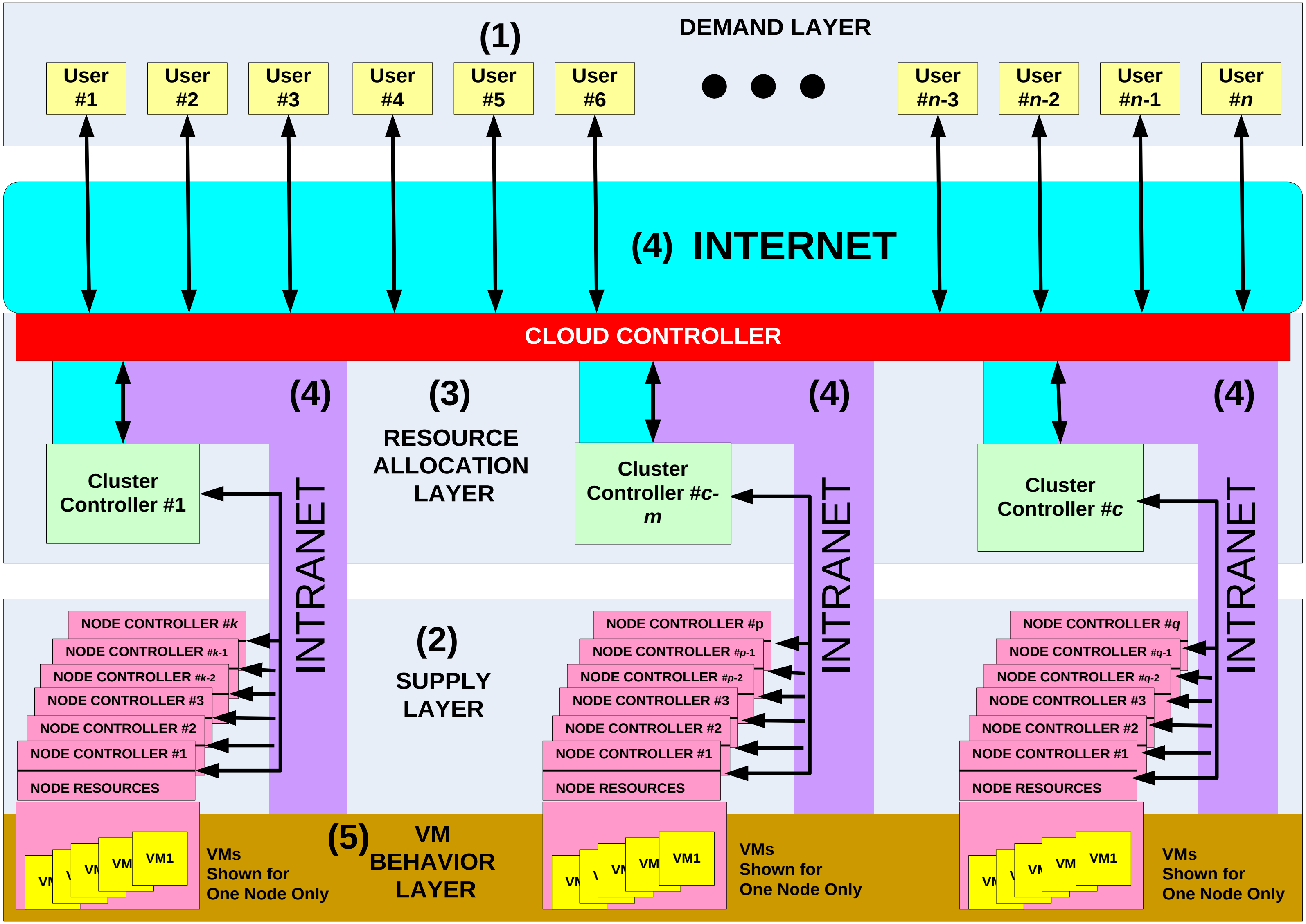
What is the Problem? Why is it Hard?

- **Problem**: Given a detailed simulation model of a complex information system, how can one identify rarely occurring combinations of conditions that could cause global system behavior to degenerate, leading to costly system outages?

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

For example, the NIST ***Koala*** simulator of IaaS Clouds has about $n = 125$ parameters with average $k = 6.6$ values each, which leads to a model **parameter space** of **~10^{100}** (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 $m \approx 42$).

Schematic of *Koala* IaaS Cloud Computing Model



Summary of *Koala* Parameters*

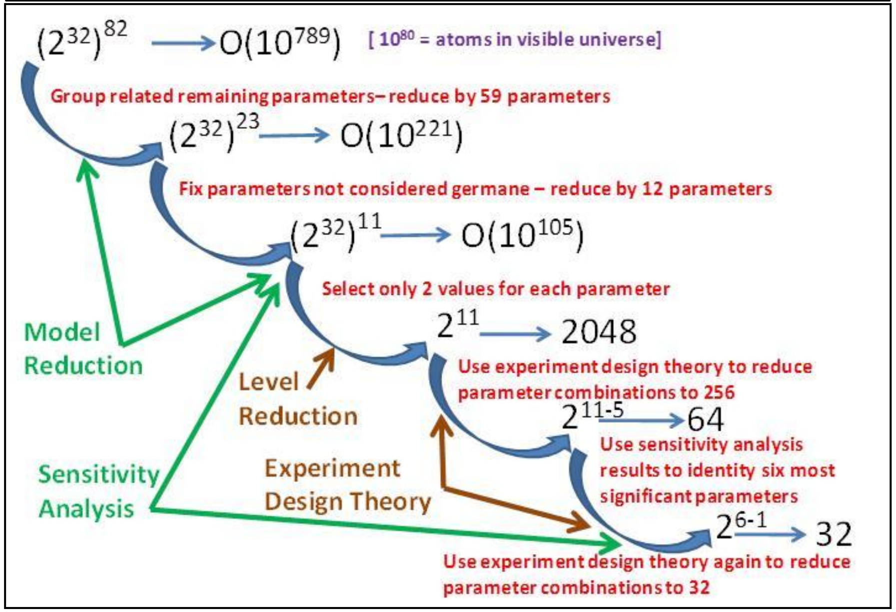
Model Element	Parameters by Category				
	Structure	Dynamics	Failures	Asymmetries	Total
Users	**2**	**30**	**0**	**3**	**35**
Cloud Controller	**3**	**18**	**0**	**2**	**23**
Cluster Controllers	**5**	**15**	**5**	**1**	**26**
Nodes	**1**	**5**	**14**	**0**	**20**
Intra-Net/ Inter-Net	**13**	**4**	**6**	**0**	**23**
Simulation Control	**8**	**2**	**0**	**0**	**10**

**Koala* continues to evolve so these parameter counts represent a temporal snapshot

How might Sensitivity Analysis help address the problem?

- **Sensitivity Analysis**: Determine which parameters most significantly influence model behavior and what response dimensions the model exhibits. Allows reduction parameter search space and identifies model responses that must be analyzed. **Can be helpful to reduce GA search space, but not essential.**

Use 2-level, orthogonal fractional factorial (OFF) experiment design to identify the most significant parameters of your model



Use correlation analysis and clustering to identify unique behavior dimensions of your model

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**

See: **Mills, Filliben and Dabrowski, "An Efficient Sensitivity Analysis Method for Large Cloud Simulations",** ***Proceedings of the 4th International Cloud Computing Conference*, IEEE (2011).**

How might **Genetic Algorithms** help address the problem?

MULTIDIMENSIONAL ANALYSIS TECHNIQUES

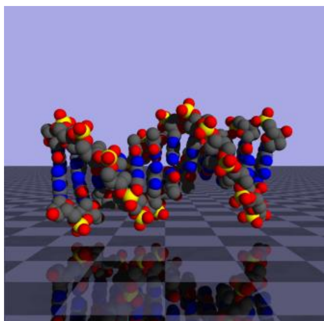
Principal Components Analysis, Clustering, …

Growing Collection of Tuples:

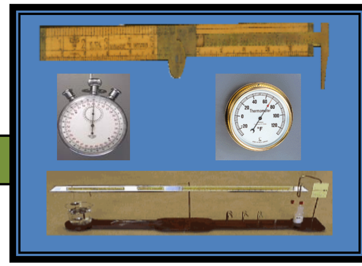
{Generation, Individual, Fitness, Parameter 1 value,….Parameter N value}
{Generation, Individual, Fitness, Parameter 1 value,….Parameter N value}
{Generation, Individual, Fitness, Parameter 1 value,….Parameter N value}
{Generation, Individual, Fitness, Parameter 1 value,….Parameter N value}
{Generation,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

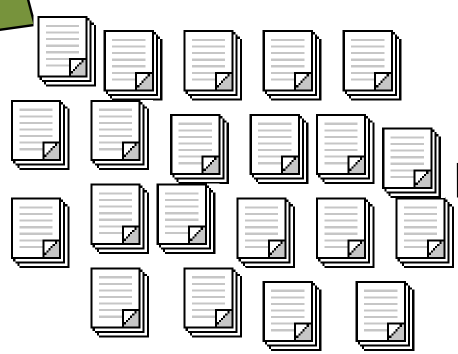


Anti-Fitness Reports

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

Sample Chromosome Specification for *Koala* Simulator

PARAMETER	MIN	MAX	PRECISION	#VALUES	LOW_BIT	HIGH_BIT	#BITS
P_CreationOrphanControlOn	0	1	1	2	1	1	1
P_TerminationOrphanControlOn	0	1	1	2	2	2	1
P_RelocationOrphanControlOn	0	1	1	2	3	3	1
P_AdministratorActive	0	1	1	2	4	4	1
P_clusterAllocationAlgorithm	0	5	1	6	5	7	3
P_describeResourcesInterval	600	3600	600	6	8	10	3
P_nodeResponseTimeout	30	90	30	3	11	12	2
P_FailedInstancesRetry	2	10	2	5	13	15	3
P_TerminatedInstancesBackOffThreshold	3	6	1	4	16	17	2
P_TerminationBackOffInterval	180	360	60	4	18	19	2
P_TerminationRetryPeriod	600	1200	300	3	20	21	2
P_StaleShadowAllocationPurgeInterval	3600	10800	1800	5	22	24	3
P_cloudAllocationCriteria	0	3	1	4	25	26	2
P_clusterShadowPurgeLimit	1	21	5	5	27	29	3
P_instancePurgeDelay	180	600	60	8	30	32	3
P_clusterEvaluationResponseTimeout	60	120	30	3	33	34	2
P_MaxPendingRequests	1	10	1	10	35	38	4
P_CloudTerminatedInstancesBackOffThresh	3	6	1	4	39	40	2
P_CloudTerminationBackOffInterval	180	360	60	4	41	42	2
P_CloudTerminationRetryPeriod	3600	10800	1800	5	43	45	3
P_ClusterShutdownGracePeriod	86400	2.59E+05	43200	5	46	48	3
●	●	●	●	●	●	●	●
P_probabilityInterSiteWSmessageLost	0.000001	0.1	0.01	10	299	302	4
P_probabilityIntraSiteWSmessageLost	0.000001	0.1	0.01	10	303	306	4
P_probabilityClusterCommunicationCut	0	0.2	0.05	5	307	309	3
P_maxTimeToClusterCommunicationCut	86400	2.59E+05	43200	5	310	312	3
P_minClusterCommunicationCutDuration	3600	14400	3600	4	313	314	2
P_modeClusterCommunicationCutDuration	21600	36000	7200	3	315	316	2
P_maxClusterCommunicationCutDuration	64800	2.16E+05	28800	6	317	319	3

Chromosome Size = 2^{319}

Parameter Space = 10^{96}

What is the current state of the project?

COMPLETED (since project inception in October 2011)

- *Koala* extended to include increased dynamics, failures and asymmetries
- Genetic Algorithm (GA) implemented, with various control parameterizations available (e.g., selection methods, crossover specifications, mutation specifications, optional population rebooting, optional scaling of parameter precisions, and optional elitism)
- Parameter specifications completed for *Koala*
- GA can generate populations of *Koala* parameterizations, control parallel execution of population of *Koala* simulators, and collect results tuples

ONGOING

- *Koala* sensitivity analysis underway
- Latent memory leaks being removed from *Koala* code
- Investigation of suitable multidimensional analysis techniques in progress
- Summer student (Andrea Haines) conducting sensitivity analysis of GA in order to determine best parameterizations to use for *Koala* exploration

What progress is expected over the next six months?

PLANNED DELIVERABLES

1. Paper characterizing the influence of failures, dynamics and asymmetries on IaaS clouds

2. Paper describing Anti-Optimization + Genetic Algorithm combination as a method to search system models to identify potential for global system collapses (and related causes), and demonstrating its application to *Koala*

3. Summer University Research Fellowship (SURF) presentation on sensitivity analysis of a GA, robust over ≥ 30 numeric optimization problems

4. Paper describing sensitivity analysis of a classic GA, and characterizing the influence of control parameters on GA effectiveness

How might your organization benefit from collaborating with us?

- **IF** your organization designs and deploys Clouds (or other large distributed systems) **AND**
  - You wish to improve the reliability of your system **AND**
  - You have a model of your system **OR**
  - You are willing to share sufficient information for us to construct a model **AND** you are willing to help us ensure our model suitability represents your system

- **THEN** working together we could help you improve the reliability of your system (or specific aspects of your system) by:
  - Applying our design-time methods to search the design space for potential collapse scenarios (and iterating on any proposed design revisions you create to mitigate collapse scenarios) **AND/OR**
  - Exploring run-time monitoring and measurement approaches that could signal incipient onset of collapse scenarios that were not detected using our design-time methods

WIN-WIN: ***we would gain additional evaluation and refinement of our methods and you could gain a transfer of our technology to enhance your design process.***

What other actions might help to improve cloud reliability?

- Formulate and publish **best common practices** (BCP) for achieving cloud reliability

- Develop a consensus **process to measure and report industry-wide cloud reliability** information to assess current and future cloud reliability, and to allow evolving measures of community progress

- **Research design-time methods** and tools (in addition to those we discussed today) to identify failure vulnerabilities and **research run-time methods** for measurement and monitoring to predict onset of catastrophic failures

…… YOUR IDEAS GO HERE ……

Questions?

Suggestions?

Ideas?

Contact information about studying Complex Information Systems:
{kmills, jfilliben, cdabrowski@nist.gov}

Contact information about Information Visualization:
sressler@nist.gov

Contact information about NIST Cloud Computing Program:
dawn.leaf@nist.gov

For more information see: http://www.nist.gov/itl/antd/emergent_behavior.cfm
and/or http://www.nist.gov/itl/cloud/index.cfm