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)
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 –
   b. Sensitivity Analysis + Anti-Optimization + Genetic Algorithm – subject of this talk

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).

June 5-7, 2012 NIST Cloud Workshop
Past ITL Research: How can we understand the influence of distributed control algorithms on global system behavior and user experience?


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

June 5-7, 2012 NIST Cloud Workshop
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?

\[
y_1, \ldots, y_m = f(x_1|[1,\ldots,k], \ldots, x_n|[1,\ldots,k])
\]

Model Response Space  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 \( \sim 10^{100} \) (note that the visible universe has \( \sim 10^{80} \) 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

**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

**INTERNET**

**SUPPLY LAYER**

- Cluster Controller #1
- Cluster Controller #c-m
- Cluster Controller #c

**RESOURCE ALLOCATION LAYER**

- Node Controller #k
- Node Controller #k-1
- Node Controller #k-2
- Node Controller #k-3
- Node Controller #p
- Node Controller #p-1
- Node Controller #p-2
- Node Controller #p-3
- Node Controller #q
- Node Controller #q-1
- Node Controller #q-2

**VM BEHAVIOR LAYER**

- VM1
- VM2
- VM3

**NODE RESOURCES**

- Node Resources

*VMs Shown for One Node Only*

June 5-7, 2012 NIST Cloud Workshop
## Summary of Koala Parameters*

<table>
<thead>
<tr>
<th>Model Element</th>
<th>Structure</th>
<th>Dynamics</th>
<th>Failures</th>
<th>Asymmetries</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>2</td>
<td>30</td>
<td>0</td>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>Cloud Controller</td>
<td>3</td>
<td>18</td>
<td>0</td>
<td>2</td>
<td>23</td>
</tr>
<tr>
<td>Cluster Controllers</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>1</td>
<td>26</td>
</tr>
<tr>
<td>Nodes</td>
<td>1</td>
<td>5</td>
<td>14</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>Intra-Net/Inter-Net</td>
<td>13</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>Simulation Control</td>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

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

June 5-7, 2012 NIST Cloud Workshop
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:

\[
(2^{32})^{82} \rightarrow O(10^{789}) \quad [10^{80} = \text{atoms in the visible universe}]
\]

- Group related remaining parameters—reduce by 59 parameters:
  \[
  (2^{32})^{23} \rightarrow O(10^{221})
  \]
- Fix parameters not considered germane—reduce by 12 parameters:
  \[
  (2^{32})^{11} \rightarrow O(10^{105})
  \]
- Select only 2 values for each parameter:
  \[
  2^{11} \rightarrow 2048
  \]
- Use experiment design theory to reduce parameter combinations to 256:
  \[
  2^{11} \rightarrow 64
  \]
- Use sensitivity analysis results to identify six most significant parameters:
  \[
  2^{26.1} \rightarrow 32
  \]

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

- Compute correlation coefficient \(r\) for all response pairs:
  \[
  \text{Response Dimension: } SA_{1\text{small}}(9 \text{ dimensions})
  \]
- Examine frequency distribution for all \(|r|\) to determine threshold for correlation pairs to retain; \(|r| > 0.65\), here:
  \[
  \text{Cloud-wide Demand/Supply Ratio: } y_7, y_5, y_6, y_8, y_9, y_{10}, y_{12}, y_{18}, y_{20}
  \]
  \[
  \text{Cloud-wide Resource Usage: } y_6, y_{12}, y_{14}, y_{15}
  \]
  \[
  \text{Variance in Cluster Load: } y_{26}, y_{27}
  \]
- Create clusters of mutually correlated pairs; each cluster represents one dimension:
  \[
  \text{Mixed Vm Types: } y_{31} \text{ (VM)}
  \]
  \[
  \text{Number of Vms: } y_{28}, y_{37}
  \]
  \[
  \text{User Arrival Rate: } y_{4}, y_{4}
  \]
  \[
  \text{Realization Rate: } y_{7}, y_{22}
  \]
  \[
  \text{Variance in Choice of Cluster: } y_{22} \text{ (VM)}
  \]

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

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

**MULTIDIMENSIONAL ANALYSIS TECHNIQUES**

- Principal Components Analysis, Clustering, ...

**GENETIC ALGORITHM**

- Recombination & Mutation
- Selection based on Anti-Fitness

**Model Parameter Specifications**

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

**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)

**MODEL SIMULATORS**

- Population of Model Parameterizations
- Parallel Execution of Model Simulators
- Anti-Fitness Reports

---

June 5-7, 2012 NIST Cloud Workshop
Sample Chromosome Specification for *Koala* Simulator

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

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

June 5-7, 2012 NIST Cloud Workshop
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 $\geq 30$ 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.

June 5-7, 2012 NIST Cloud Workshop
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:
{k mil ls, 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