NIST Cloud Computing Program

NIST Cloud Computing Forum and Workshop VIII

Predicting Global Failure Regimes in Complex Information Systems

Kevin Mills, NIST

July 9, 2015

Project Research Goals

- Develop ***design-time methods*** that system engineers can use to detect existence and causes of costly failure regimes prior to deployment
- Develop ***run-time methods*** that system managers can use to detect onset of costly failure regimes in deployed systems, prior to collapse

Topics

- Some past results on design-time methods
- Example→ Applying one design-time method to seek failure scenarios in a cloud system
- Ongoing work on run-time methods
- Where to find more information

Some Past Results

State-space reduction techniques and their application to clouds

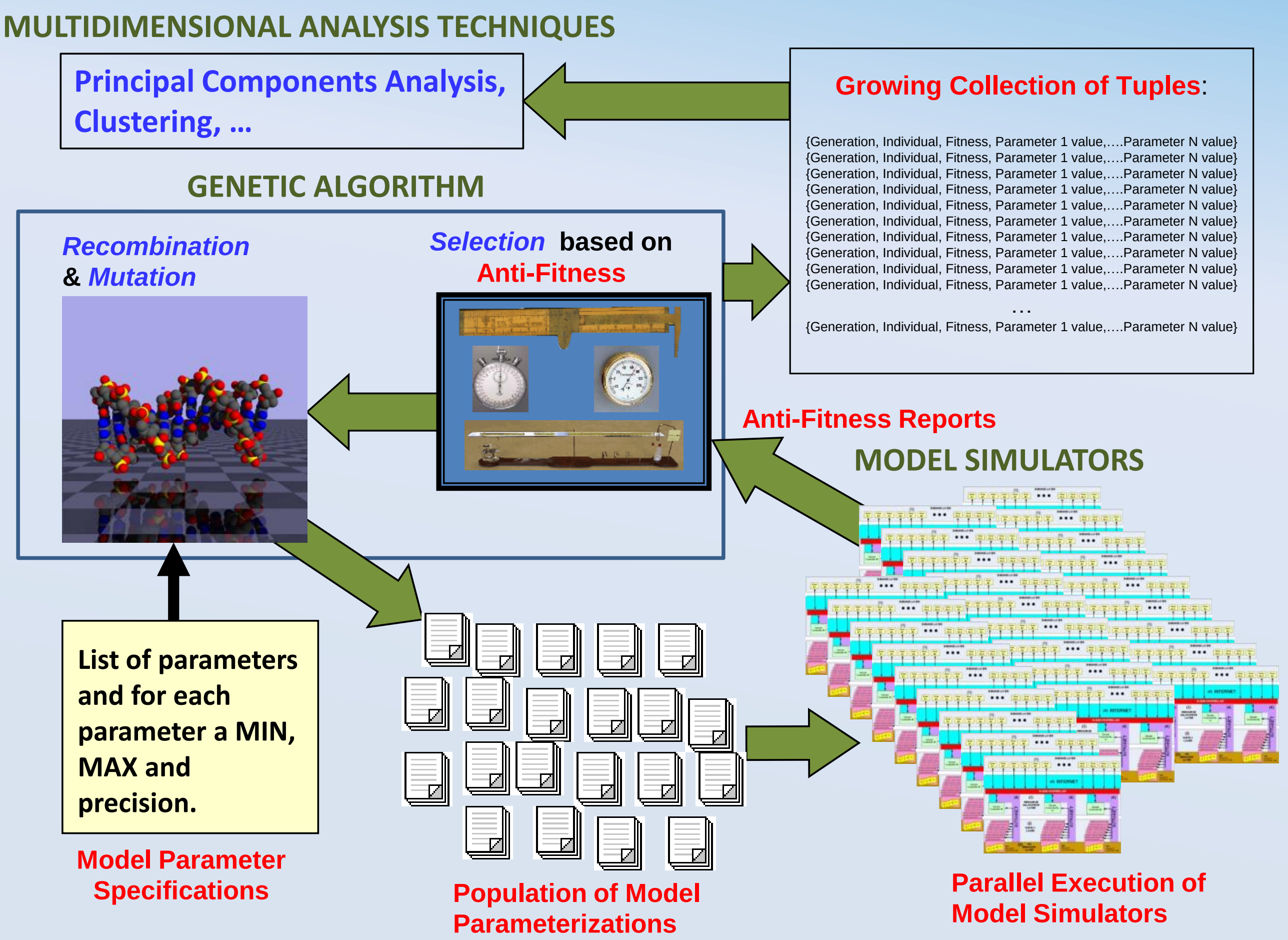
- Modeling Complex Systems with Few Input Parameters
- Identifying Key Combinations of Parameters & Values to Compare
- Identifying Essential Responses to Measure and Compare
- Applying Techniques to Assess Effects of Asymmetries, Dynamics and Failures on a Cloud Simulator

Directed and self-directed search techniques and their application to clouds

- Identifying Failure Scenarios in Complex Systems by Perturbing Markov Chains
- Using Spectral Methods to Streamline Search for Failure Scenarios
- Combining GAs & Simulations to Search for Failure Scenarios in System Models

←EXAMPLE (next 5 slides)

Method: Genetic Algorithm (GA) steers a population of simulators to search for parameter combinations that lead to system failure



In our following **example**, we use the *Koala* cloud simulator, and we define *anti-fitness* as the proportion of users not served, and we use differential probability analysis on the collection of tuples.

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

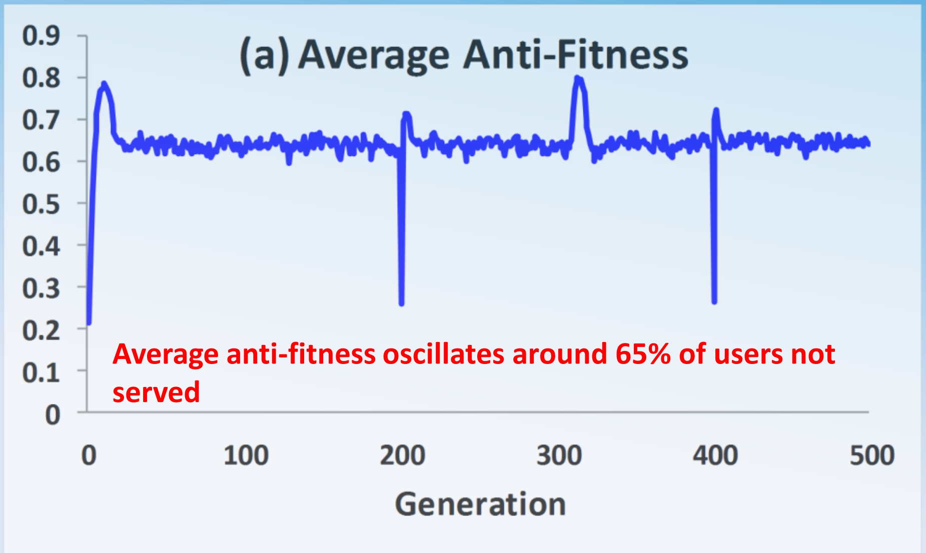
Average # values per parameter is about 6, so search space is ≈ 6^{129}
i.e., ≈ 10^{100} scenarios are possible

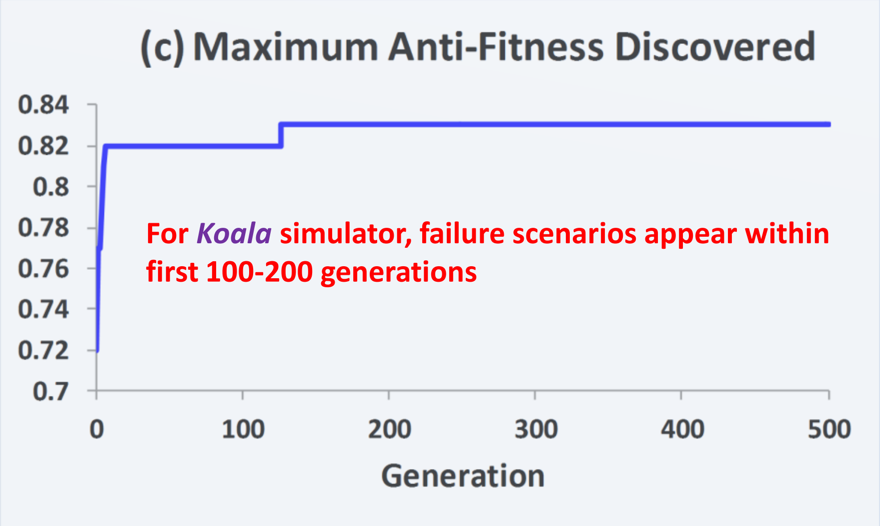
- adapted 125-parameter *Koala* IaaS simulator to be GA controllable
- added 4 *Koala* parameters to turn on/off logic to control (a) creation orphans, (b) termination orphans, (c) relocation orphans and (d) administrator actions

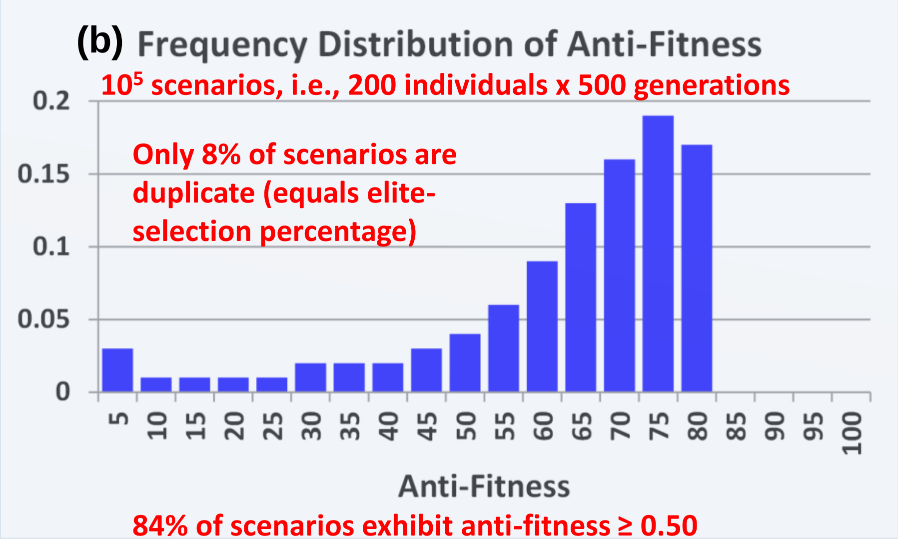
Koala GA Search over 500 Generations

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







Differential Probability Analysis

Let $\boldsymbol{C}$ be the set of collected tuples, each containing a vector of parameter value (PV) pairs and a corresponding anti-fitness value, $\boldsymbol{f}$

Segment $\boldsymbol{C}$ into high-pass ($\boldsymbol{H}$) and low-pass ($\boldsymbol{L}$) subsets, where:

$$\boldsymbol{H} = \{\boldsymbol{x} \in \boldsymbol{C} \mid \boldsymbol{f_x} > 0.70\} \text{ and } \boldsymbol{L} = \{\boldsymbol{x} \in C \mid \boldsymbol{f_x} < 0.15\}$$

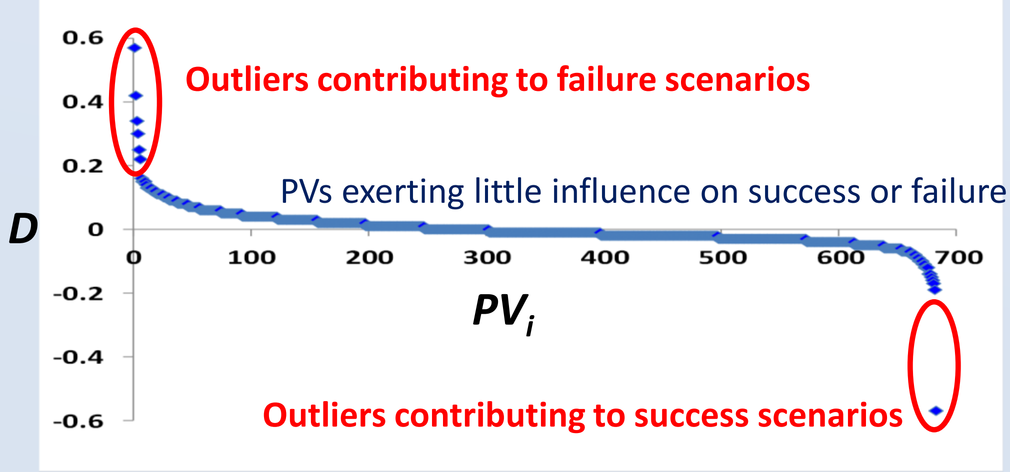
For each PV estimate the probability of occurrence in $\boldsymbol{H}$ and $\boldsymbol{L}$:

$$\mathrm{P}(\mathrm{PV}_i \,|\, \boldsymbol{f} > 0.70 = |\mathrm{PV}_i \in \boldsymbol{H}| / |\boldsymbol{H}| \text{ and } \mathrm{P}(\mathrm{PV}_i \,|\, \boldsymbol{f} > 0.15) = |\mathrm{PV}_i \in \boldsymbol{L}| / |\boldsymbol{L}|$$

Then compute the estimated differential probability:

$$\boldsymbol{D} = \mathrm{P}(\mathrm{PV}_i \,|\, \boldsymbol{f} > 0.70) - \mathrm{P}(\mathrm{PV}_i \,|\, \boldsymbol{f} < 0.15)$$

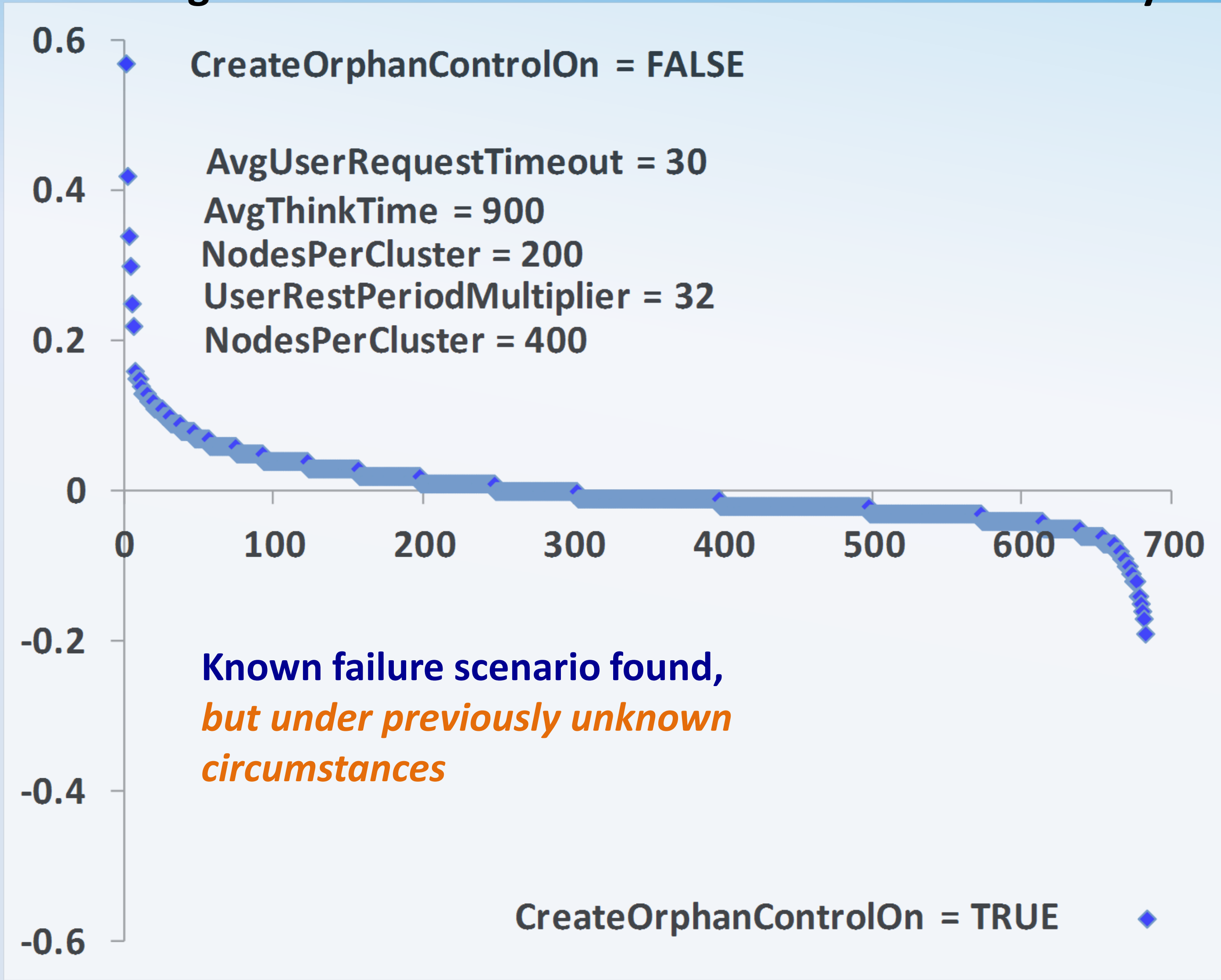
Plot $\boldsymbol{D}$ for each PV pair



PV pairs sorted by decreasing *D*

Analysis of Results from *Koala* GA Search 1 – 500 Generations

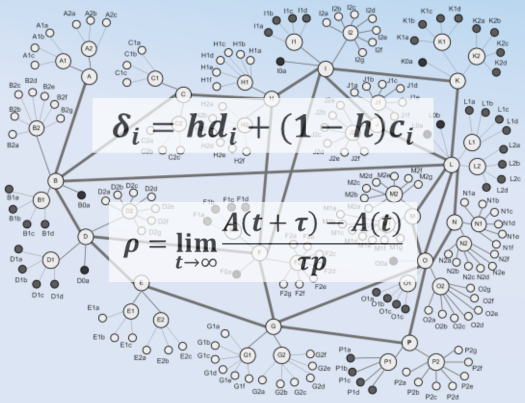
Seeking Known Failure Scenario – search duration 30 days



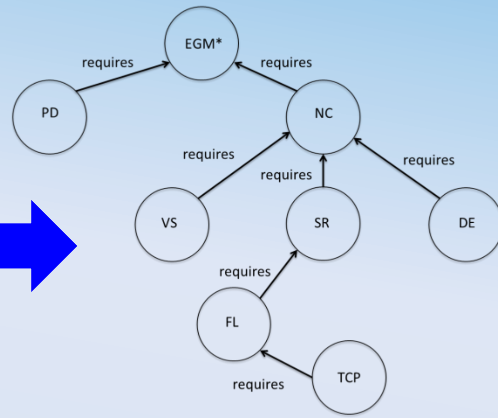
D (y-axis) for 684 PV pairs (x-axis) for first GA search–outlier PV pairs labeled.

Ongoing Work: Do published findings on the spread of congestion hold for realistic network models?

Abstract Network Model



Add 7 Realism Factors



34 Valid Combinations

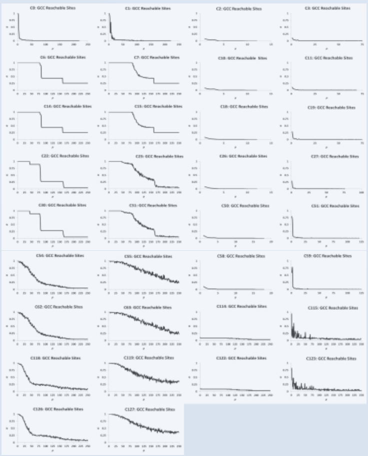
Most Abstract

Config	TCP	FL	SR	DE	VS	NC	PD
C0	0	0	0	0	0	0	0
C1	0	0	0	0	0	0	1
C2	0	0	0	0	0	1	0
C3	0	0	0	0	0	1	1
C6	0	0	0	0	1	1	0
C7	0	0	0	0	1	1	1
C10	0	0	0	1	0	1	0
C11	0	0	0	1	0	1	1
C14	0	0	0	1	1	1	0
C15	0	0	0	1	1	1	1
C18	0	0	1	0	0	1	0
C19	0	0	1	0	0	1	1
C22	0	0	1	0	1	1	0
C23	0	0	1	0	1	1	1
C26	0	0	1	1	0	1	0
C27	0	0	1	1	0	1	1
C30	0	0	1	1	1	1	0
C31	0	0	1	1	1	1	1
C50	0	1	1	0	0	1	0
C51	0	1	1	0	0	1	1
C54	0	1	1	0	1	1	0
C55	0	1	1	0	1	1	1
C58	0	1	1	1	0	1	0
C59	0	1	1	1	0	1	1
C62	0	1	1	1	1	1	0
C63	0	1	1	1	1	1	1
C114	1	1	1	0	0	1	0
C115	1	1	1	0	0	1	1
C118	1	1	1	0	1	1	0
C119	1	1	1	0	1	1	1
C122	1	1	1	1	0	1	0
C123	1	1	1	1	0	1	1
C126	1	1	1	1	1	1	0
C127	1	1	1	1	1	1	1

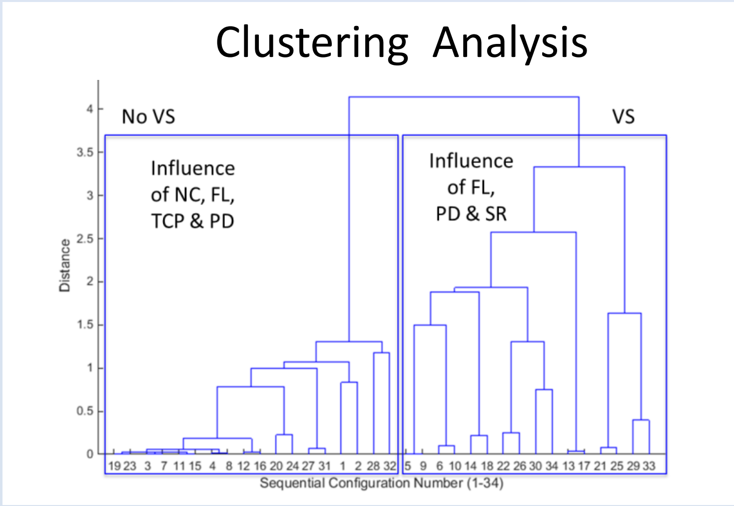
Most Realistic

Y: Network Disruption

vs.



X: Increasing Load



To Learn More

Project Team (the core four)

- **Kevin Mills,** computer scientist – kmills@nist.gov
- **Chris Dabrowski,** computer scientist – cdabrowski@nist.gov
- **Jim Filliben,** statistician – jfilliben@nist.gov
- **Sandy Ressler,** information visualization specialist – sressler@nist.gov

Project Page

- **http://www.nist.gov/itl/antd/emergent_behavior.cfm**