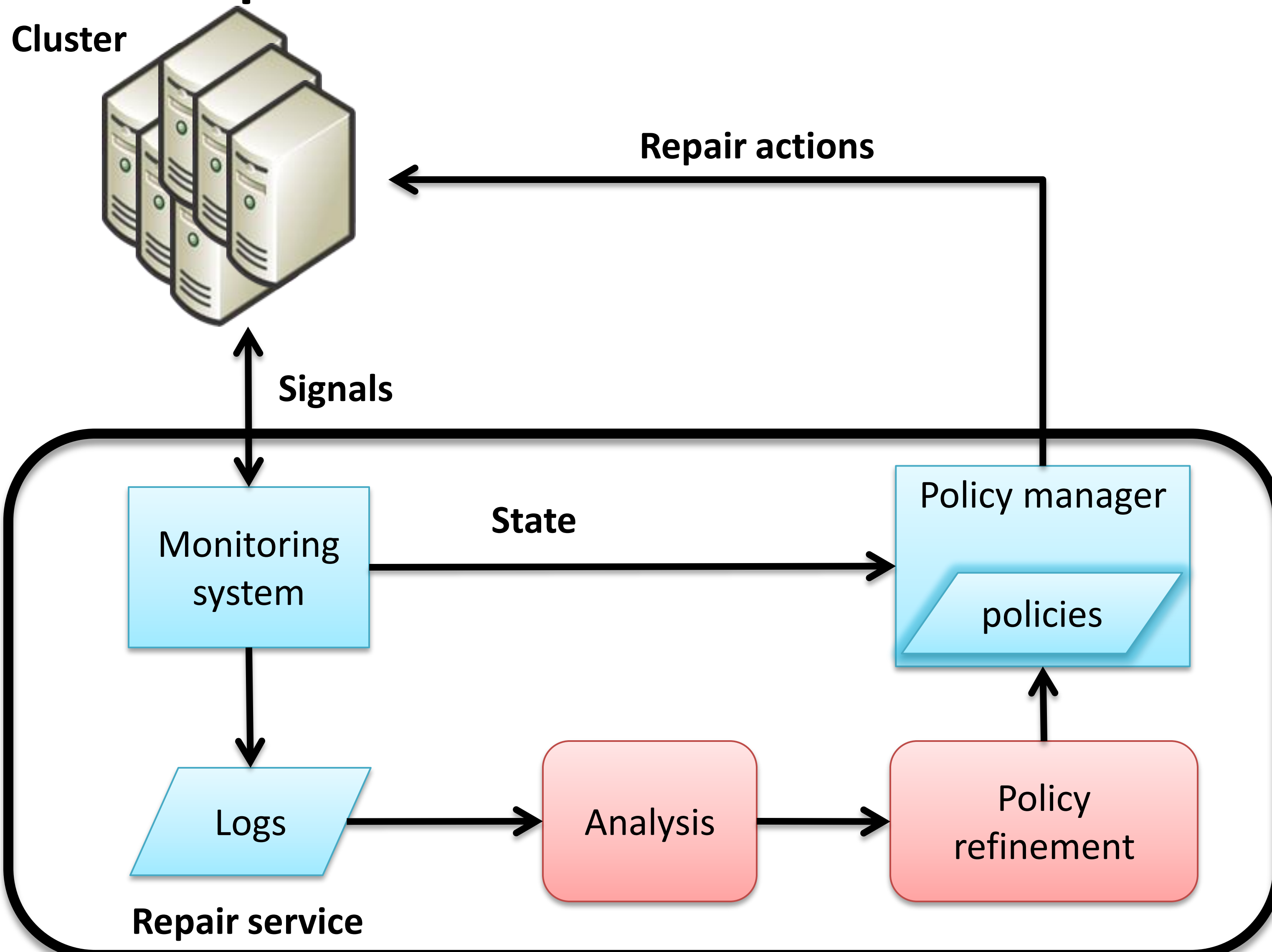
Toward Automatic Policy Refinement in Repair Services for Large Distributed Systems

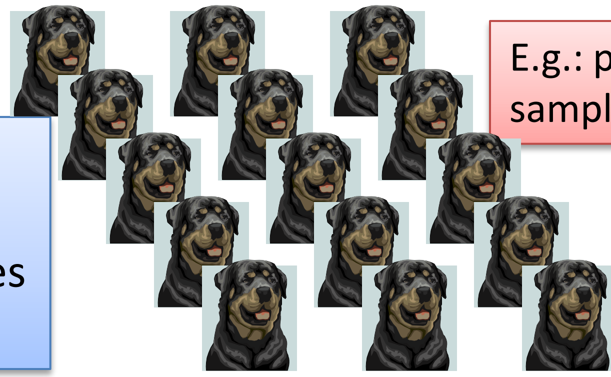
M. Goldszmidt, M. Budiu, Y. Zhang, M. Pechuk

Microsoft

The problem we are addressing



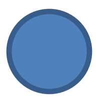
The repair service



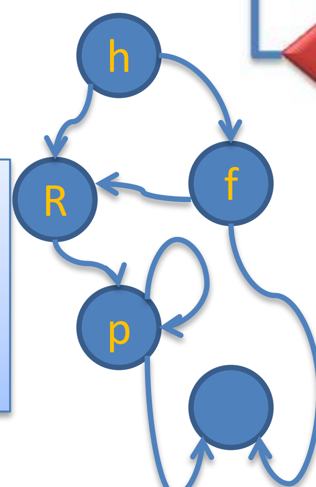
E.g.: ping, execute transaction, sample cpu, etc.

Watchdogs: Asynchronously monitoring machines and sending signals

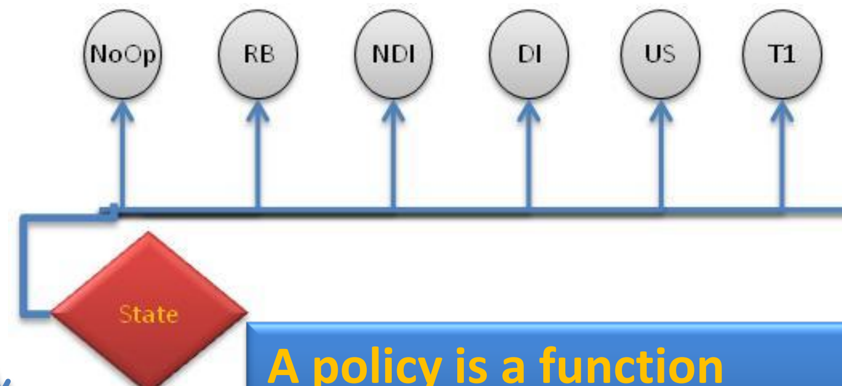
Each machine has a state associated with it



E.g.: healthy, probation, faulty, rebooted_once, etc.



State transitions are regulated by an automaton. A signal or a repair action will cause a state transition



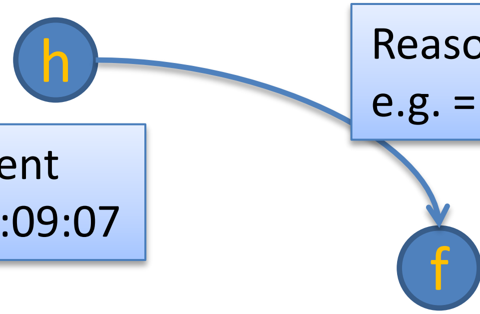
A policy is a function from State to Repair Action

E.g.:
If probation do_nothing.
If rebooted_once reboot.
If dead call tier_1 operator

Logs

Log consists of
3 months of data
collected from
~ 2k machines

```
LocalTime, FromState, ToState, Reason, HostID, requestor
"2009-02-21 02:09:07.733", H, F, 8382, 14, machine
"2009-02-21 02:11:03.377", F, P, NULL, 14, machine
"2009-02-21 04:11:46.780", P, H, 0, 14, machine
"2009-02-21 04:56:31.380", H, F, 8360, 120, machine
"2009-02-21 05:01:06.080", F, P, NULL, 120, machine
"2009-02-21 07:07:22.430", P, H, 0, 120, machine
"2009-02-21 18:49:21.060", H, F, 8360, 134, machine
"2009-02-21 18:51:14.690", F, P, NULL, 134, machine
"2009-02-21 20:51:20.123", P, H, 0, 134, machine
"2009-02-22 05:17:26.937", H, F, 8360, 168, machine
"2009-02-22 05:21:22.147", F, P, NULL, 168, machine
"2009-02-22 07:21:50.440", P, H, 0, 168, machine
"2009-02-23 11:02:29.197", H, F, 8360, 184, machine
"2009-02-23 11:06:45.733", F, P, NULL, 184, machine
"2009-02-23 11:37:02.417", P, F, 8383, 184, machine
"2009-02-23 11:41:46.473", F, RB, NULL, 184, machine
"2009-02-23 11:47:22.297", RB, P, 0, 184, machine
"2009-02-23 13:49:15.810", P, H, 0, 184, machine
```



Reason for transition
e.g. = e8382

Time of the event
2009-02-21 02:09:07

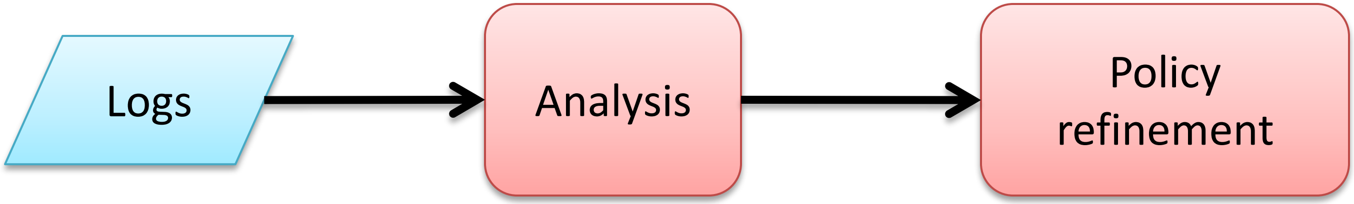
Research questions

Given the data in the logs:

1. Estimate the 'effectiveness' of a repair action

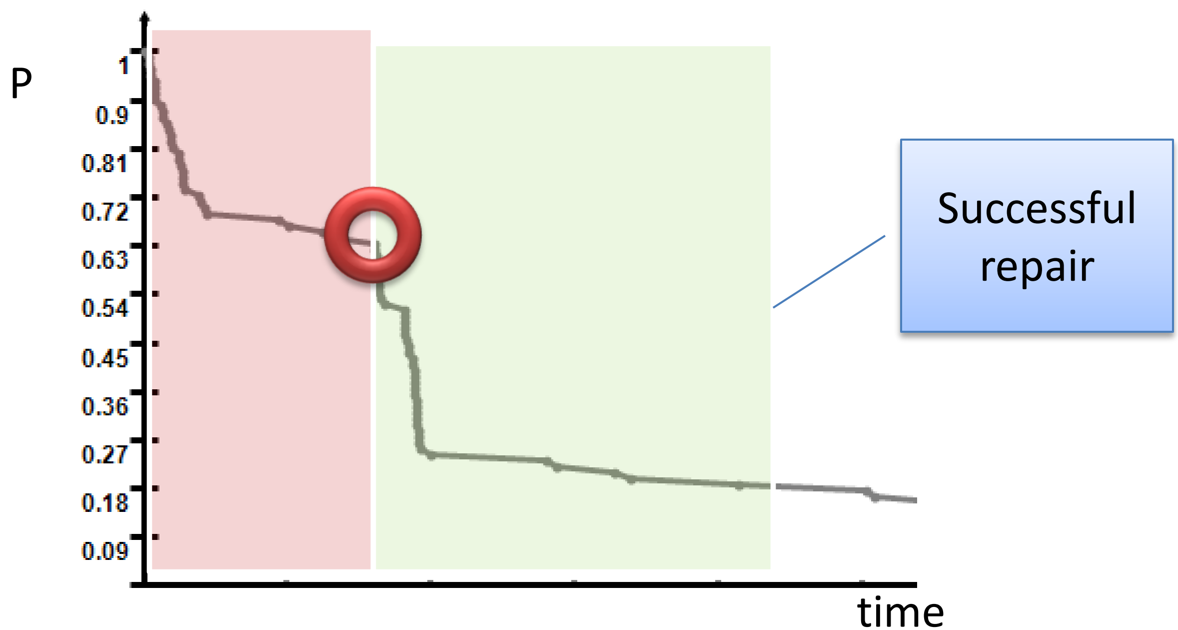
   What is a "successful" repair action?

2. Suggest alternative (better) policies (without intervention)



Effectiveness and success

- Effectiveness → time that a machine is 'usable'
- Estimate the survival curve of the repair action

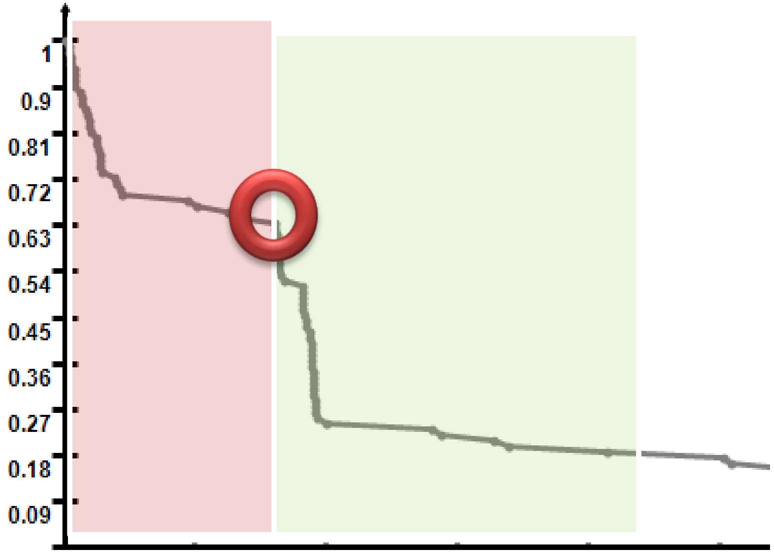


duration2	event2
19	0
134	0
277	1
555	1
572	0
632	1
722	1
827	0
929	1
1429	1
2594	1
2754	1
2828	1
3169	1
3446	1
3937	1

Successful repair = threshold on *P* of survival <u>and</u> time

Modeling successful repairs

Automatically find a function from watchdog-signals to success



Machine learning to the rescue:

classification with feature selection.

Logistic regression with L1 regularization

Models of success

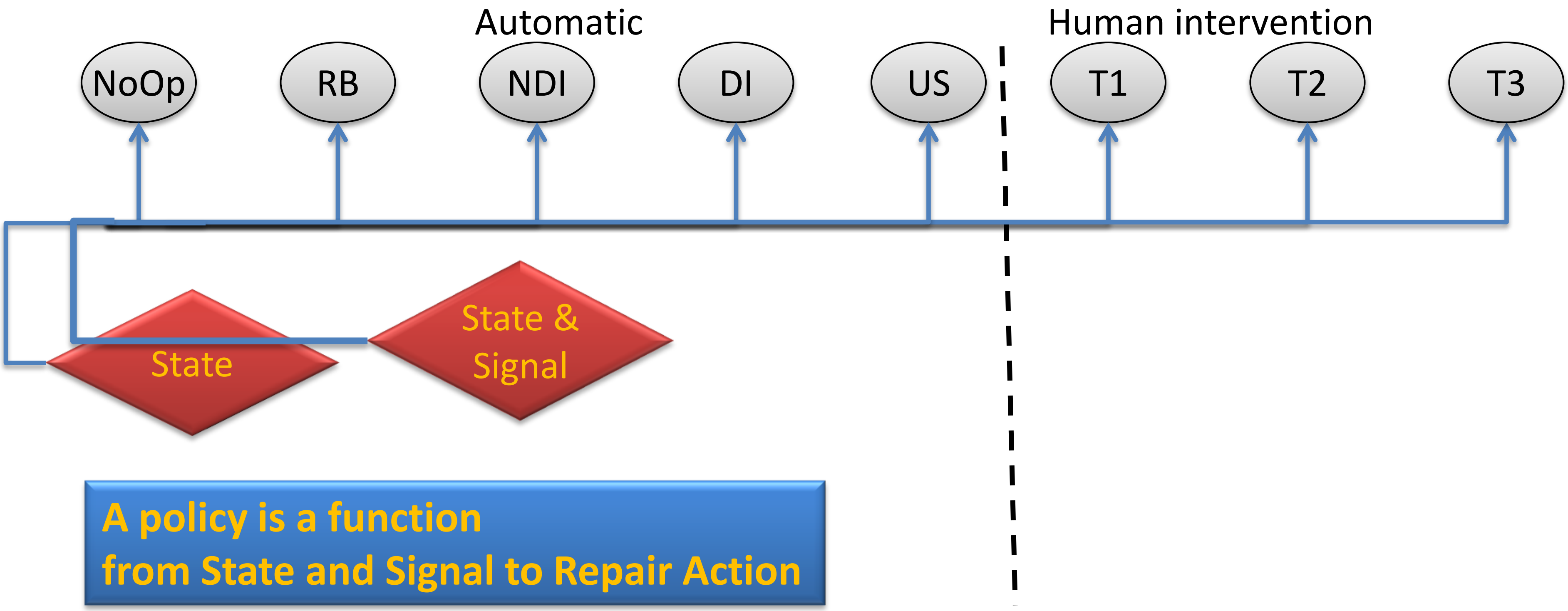
selected signals: 9
CV BA: 0.872
CV confusion matrix:

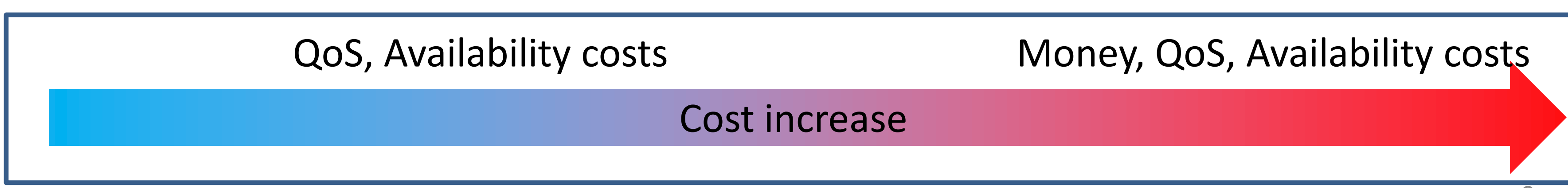
	below	above
pred below	89	14
pred above	11	71

	coeffs	ind	threshold
e50202	-0.79	0.965	0.00
e8240	-0.89	0.942	0.00
e8383	0.31	0.692	1.00
e8506	-0.84	0.861	0.00

185 samples with 42 signals

Refining policies

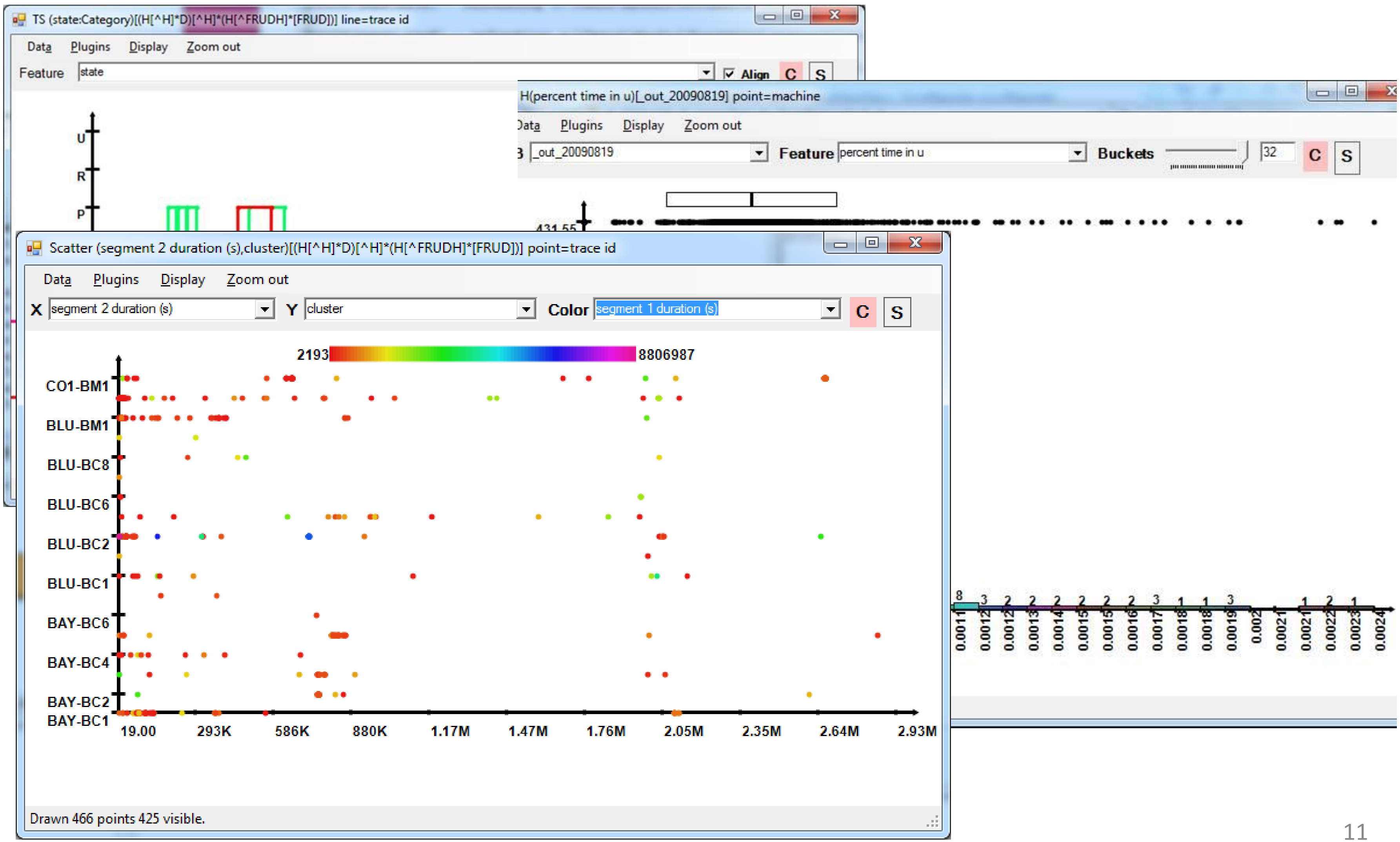




Data processing (with Artemis)

1. Use regular expression to extract segments of data
2. Extract duration and censoring events
3. Estimate survival curves
4. Define success
5. Extract the signals before the repair action
6. Induce models of success/fail
7. Present relevant signals

Data visualization (with Artemis)

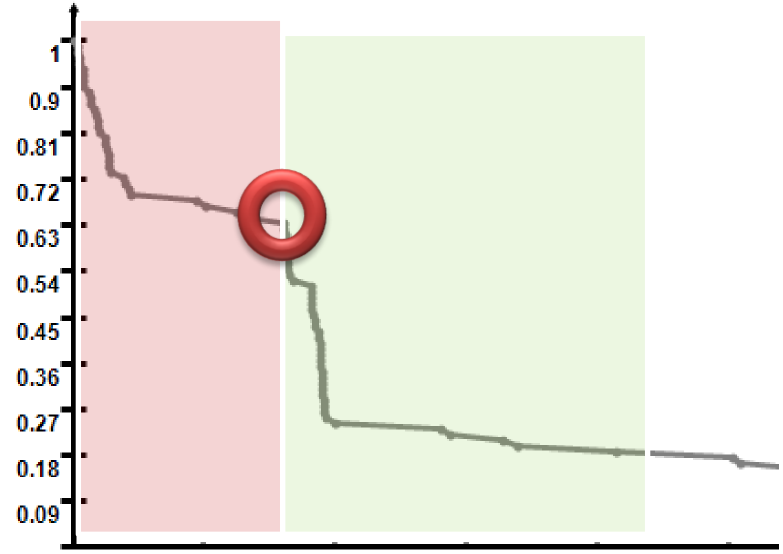


Results

- Comparing different datacenters
  - Statistical tests on the different survivability curves
  - Visualization (correlation graphs)
- Models for different repair actions

The bad sensor case

E8382



How come 1 signal was predicting with 98% accuracy the failure to repair?

Further investigation → faulty sensor!!

New models (3 months after the fix) have a mixture of many signals and E8382 appears as evidence for success…

Faulty repair procedure

Snippet of the T1-REPAIR model

	coeffs	ind	threshold
S1	-0.79	0.965	0.00
S2	-0.89	0.942	0.00
S4	-0.84	0.861	0.00

S2 is indicative of an easy fix... Why was not effective?

Bug in the repair instructions.... Fixed!

What about S1 and S4?

Final Remarks

- Models directed the debugging of the repair service.
  - Signals that are strong indications of failed repair
  - Signals that are irrelevant
- In two weeks the results helped improve a system that was "hand-tuned" during 6 months
- Further automate the whole workflow
- Induce models of correlated watchdogs
- Correlate to performance data