DViz

Jodi Spacek and Stewart Grant

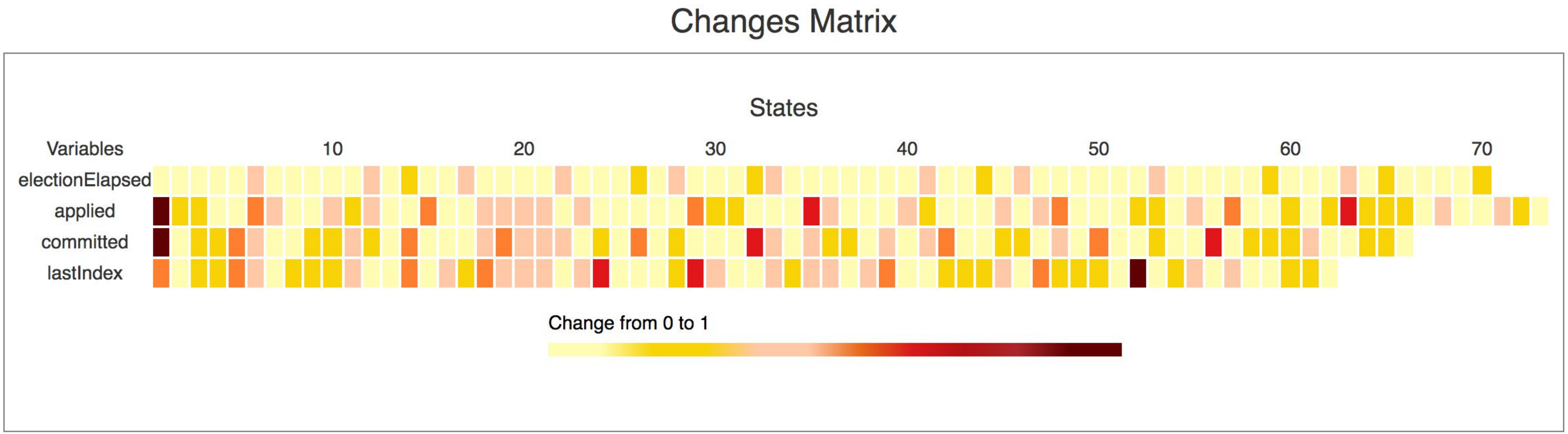
Reasoning about Distributed Systems

Why are distributed systems are difficult to reason about?

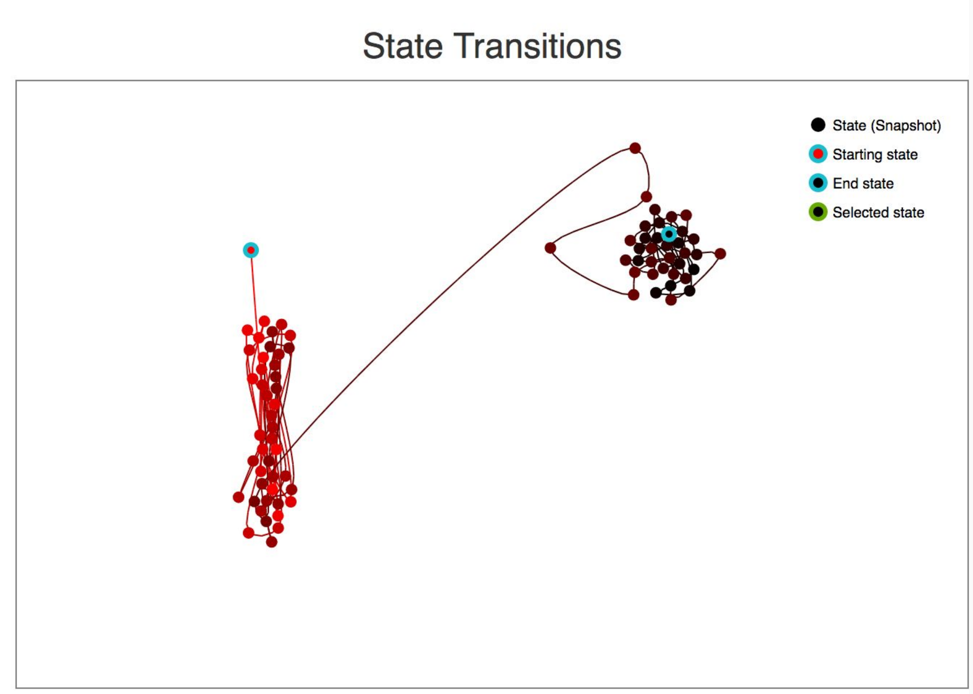
- Time
- State
- Nodes

What do we want to see in a system execution?

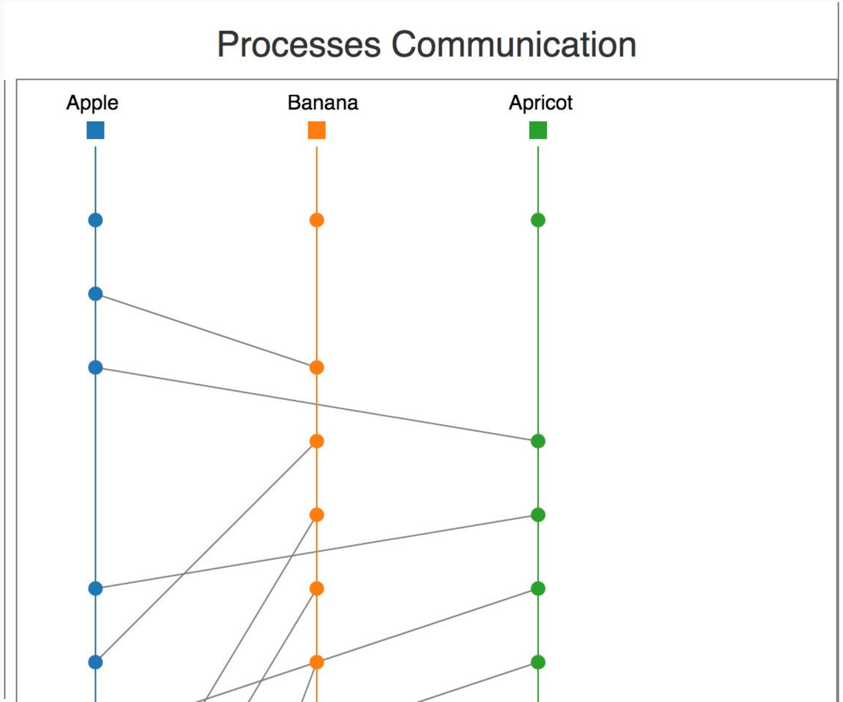
Changes over a System Execution



System Execution Drill Down



State Transitions



Processes Communication

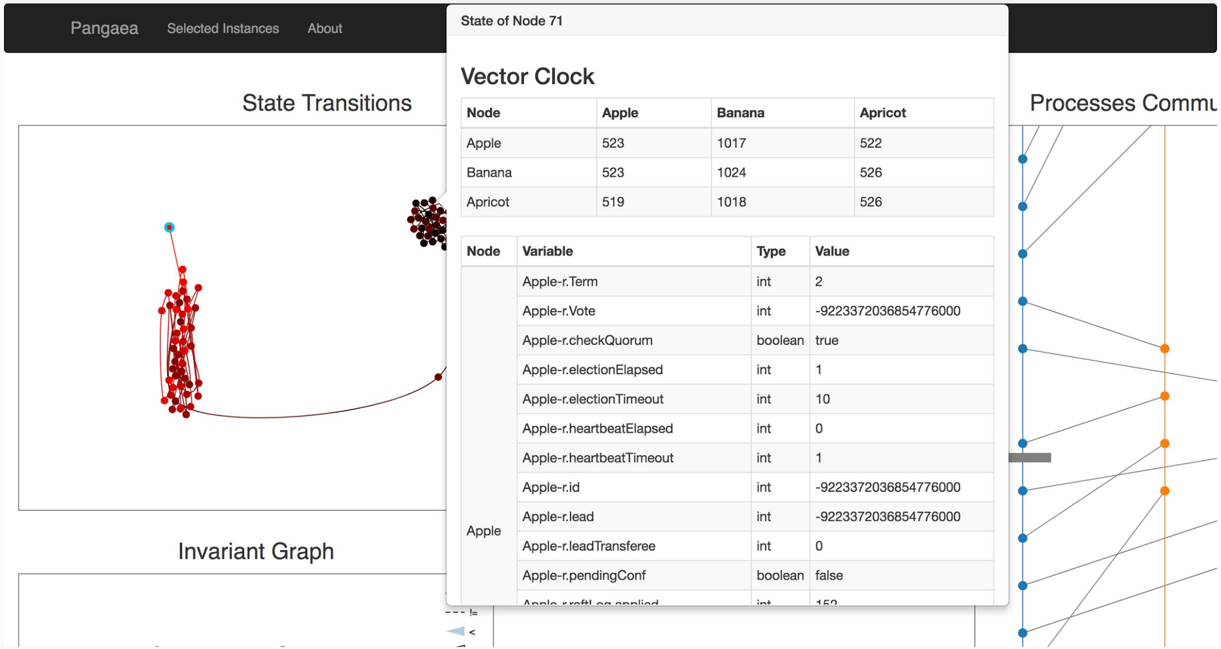
Navigating Change

Demo time!

Pangaea

Absolute Data View

Pangaea



State of Node 71

Vector Clock

Node	Apple	Banana	Apricot
Apple	523	1017	522
Banana	523	1024	526
Apricot	519	1018	526

Node	Variable	Type	Value
Apple	Apple-r.Term	int	2
	Apple-r.Vote	int	-9223372036854776000
	Apple-r.checkQuorum	boolean	true
	Apple-r.electionElapsed	int	1
	Apple-r.electionTimeout	int	10
	Apple-r.heartbeatElapsed	int	0
	Apple-r.heartbeatTimeout	int	1
	Apple-r.id	int	-9223372036854776000
	Apple-r.lead	int	-9223372036854776000
	Apple-r.leadTransferee	int	0
	Apple-r.pendingConf	boolean	false
	Apple-r.raftLog.applied	int	152

Analysis

What: Data	Sequential (time curve) data points
What: Derived	Change between current and previous point
How: Reduce	Filter on changed values
How: Encode	Colour Map (table format), Saturation and Hue to represent degree of change
Why: Tasks	Locate, identify and compare changes in state

Limitations

- Matrix to Timecurve connections
- Support for multiple (> 80) snapshots
- Handles systems with expected changes (eg. elections)

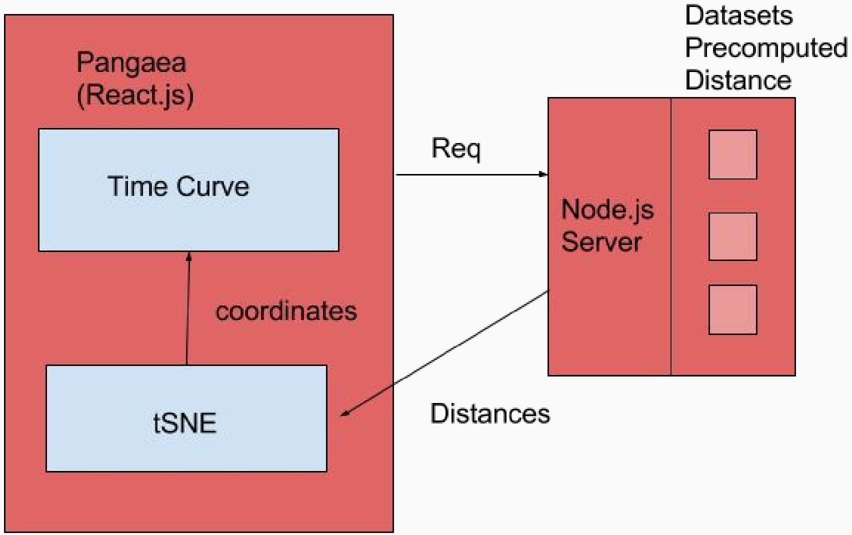
Interactivity and Data Refinement

Motivation

- Static visual prevented users from exploring datasets
- Why points are distant is mysterious
- Time curve computation was slow (8min for a 60s execution)
- Visual clusters lack identification information

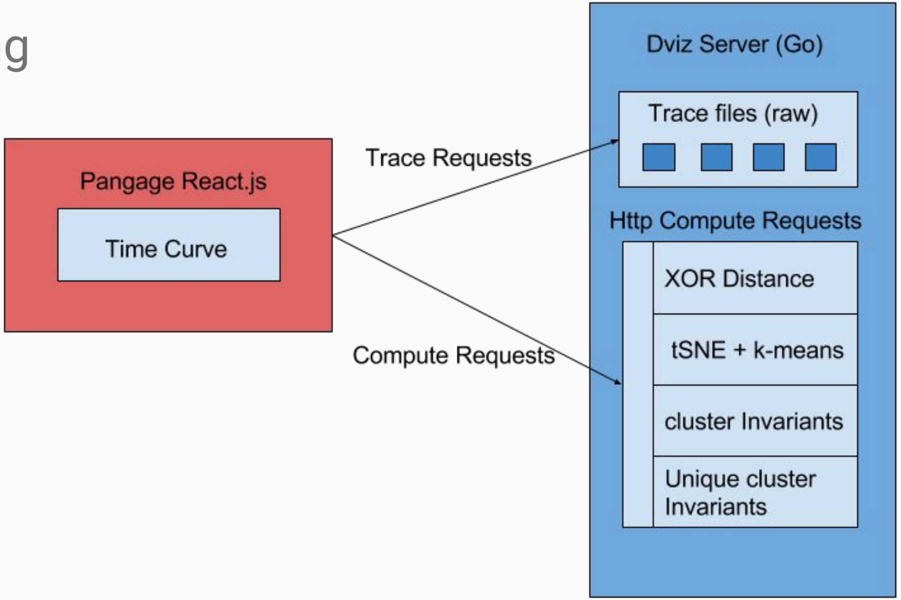
Slow Visualization (Thanks JavaScript)

- Precomputed distances
- tSNE too slow for large traces



Thin Client with Go Backend

- Decouples compute and rendering
- Fast parallel compute



XOR improvements

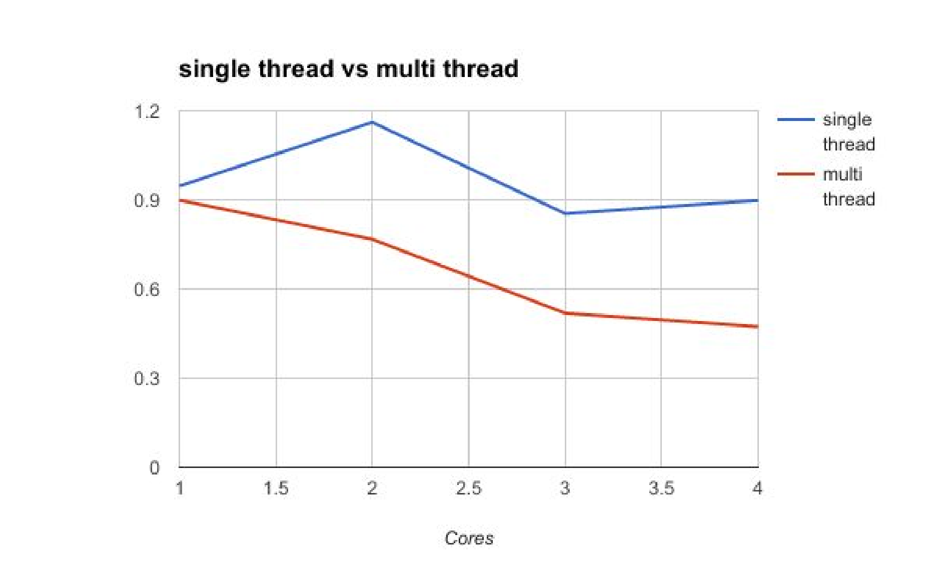
- Prior XOR took minutes to compute
- All variables treated equally regardless of importance
- Variables contributing to distances were hidden

XOR improvements

- Optimized single threaded 20x speedup + multi-threaded scalable
- Distance coefficients $\sqrt{v_1^2 + v_2^2 + \ldots + v_n^2} \rightarrow \sqrt{(c_1 v_1)^2 + (c_2 v_2)^2 + \ldots + (c_n v_n)^2}$
- Causal variables

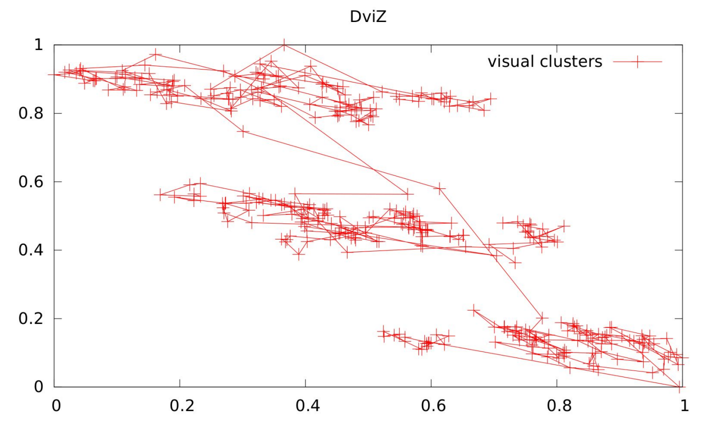
tSNE Improvements

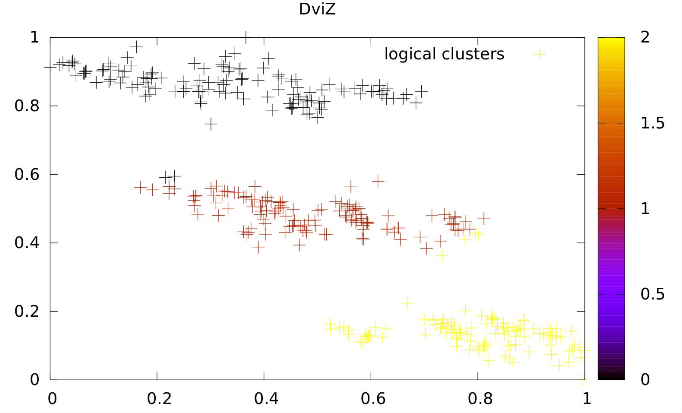
- tSNE in JavaScript is slow, we moved to a Go implementation
- Go was still too slow for large traces, We implemented parallel tSNE
- 5x faster in Go



Newly Derived Data (Logical Clustering)

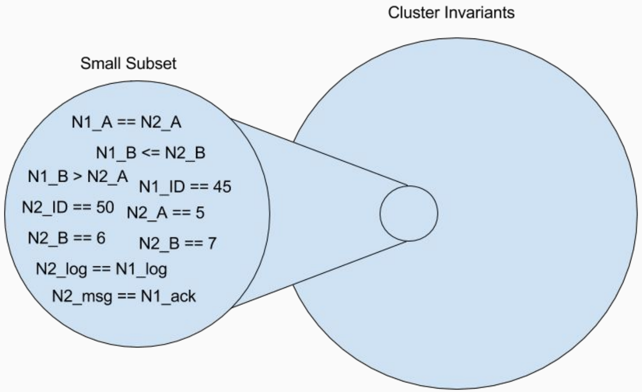
- tSNE generates visual clusters which cannot be further processed
- clustering tSNE coordinates with K-means allows for further processing





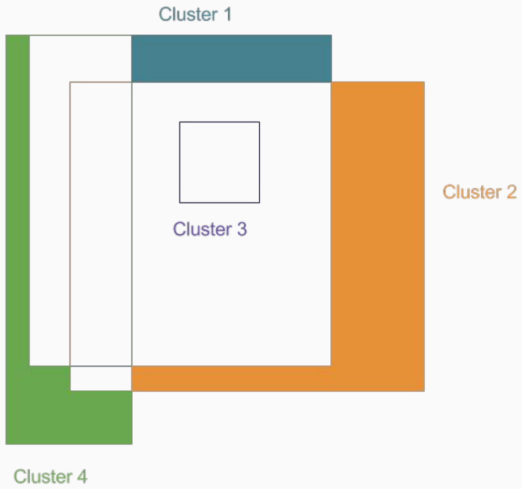
Cluster Invariants

- Logical clusters corresponded to similar stages of an execution
- Using Daikon we can identify data properties on individual clusters
- Logging tones of state leads to tons of invariants! (~300 per cluster)



Unique Cluster Invariants

- Compare all logical clusters to identify unique invariants.
- # of unique invariants is small enough to present to users (0 -10 per)



Summary

Interactivity:

- 8 min -> 10s xor + tSNE
- Variable weighting, and distance Causality

Data Refinement:

- Logical cluster detection
- Cluster Invariants
- Unique Cluster Invariants