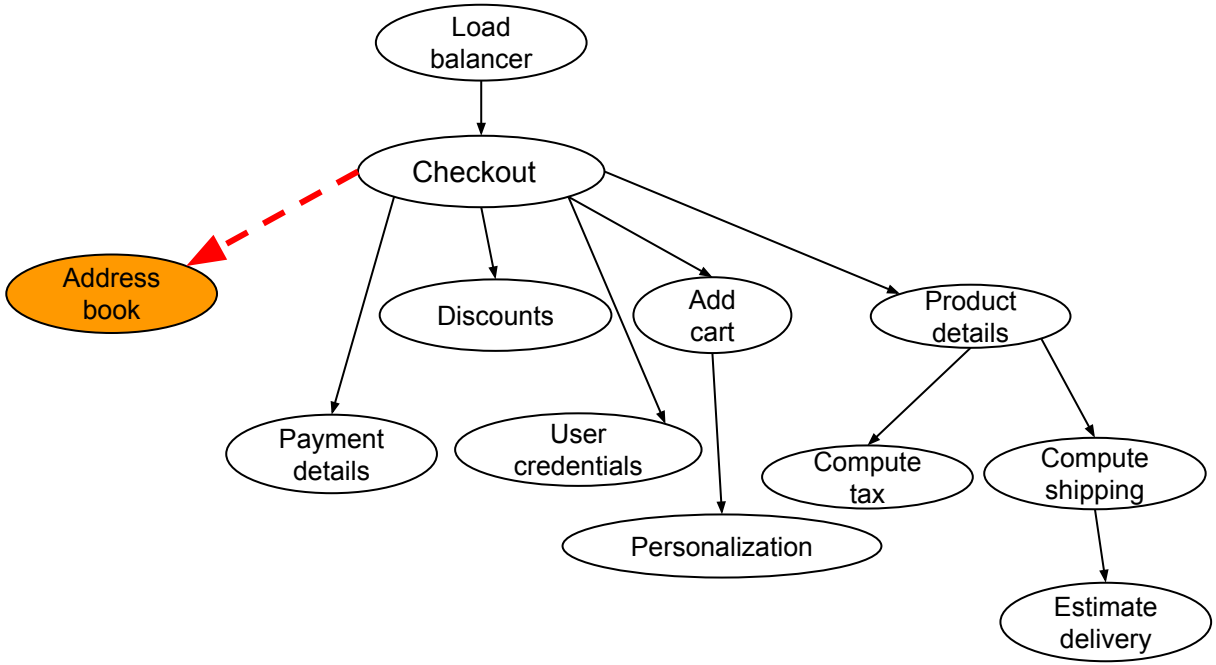
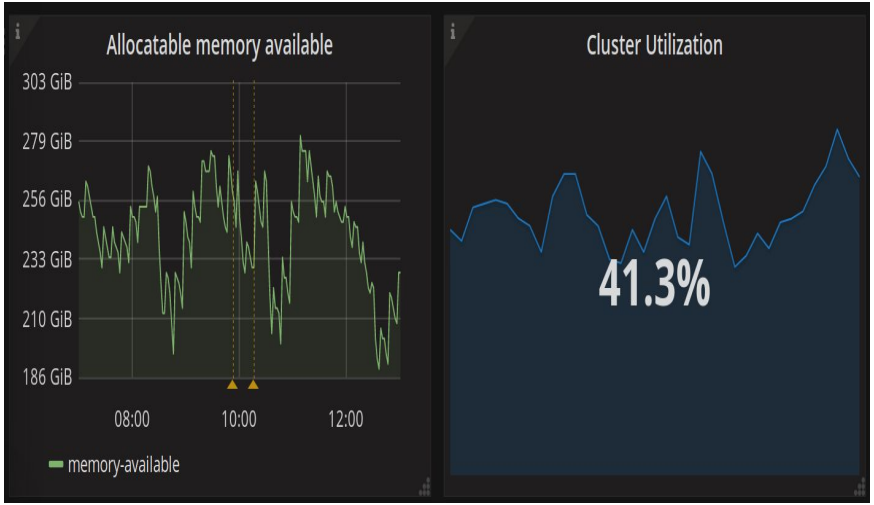
Automated fault diagnostics

Kamala Ramasubramanian
Disorderly Labs, UCSC

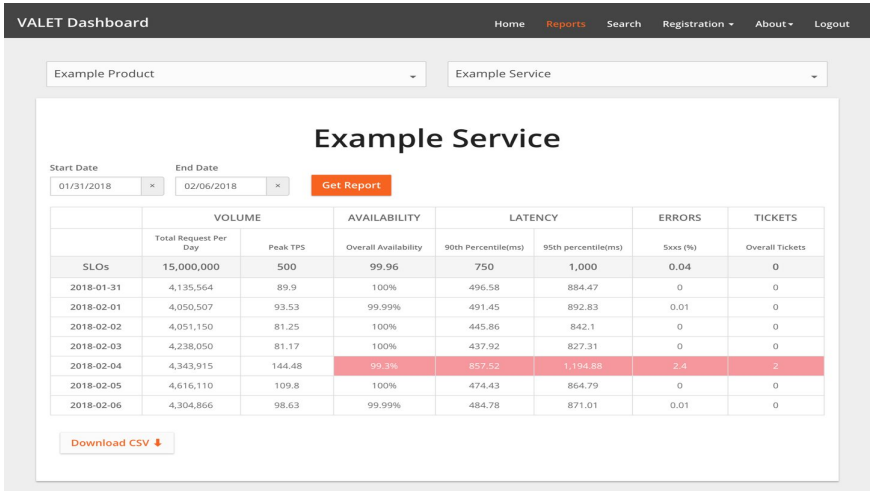
Problem addressed: Real-time debugging

Operators must perform remediation to mitigate site outages, but the right mitigation often requires understanding the chain of event interactions

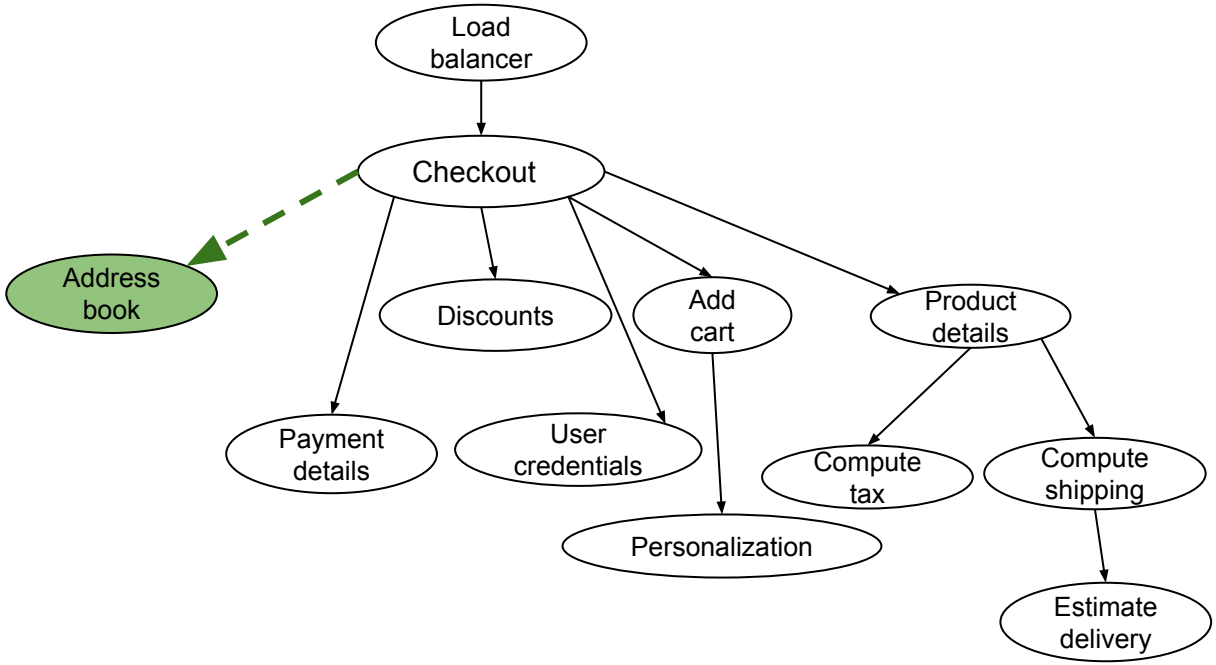

Load balancer
Checkout
Address book
Discounts
Add cart
Product details
Payment details
User credentials
Compute tax
Compute shipping
Personalization
Estimate delivery

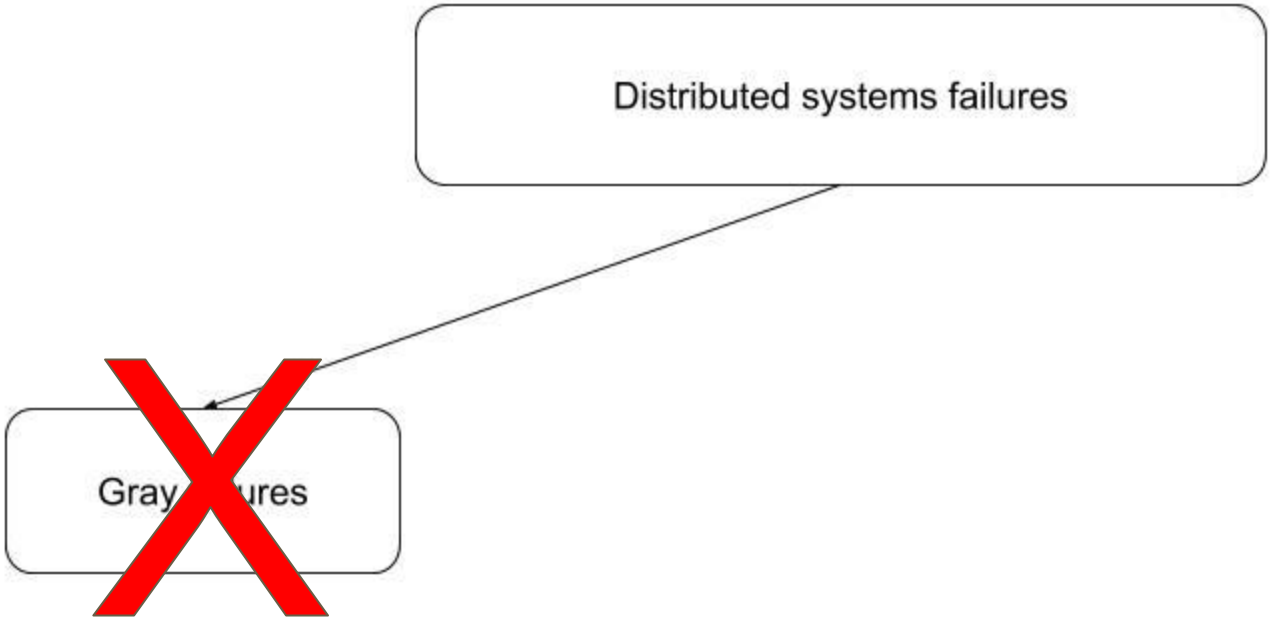

Allocatable memory available
303 GiB
279 GiB
256 GiB
233 GiB
210 GiB
186 GiB
08:00
10:00
12:00
memory-available
Cluster Utilization
41.3%

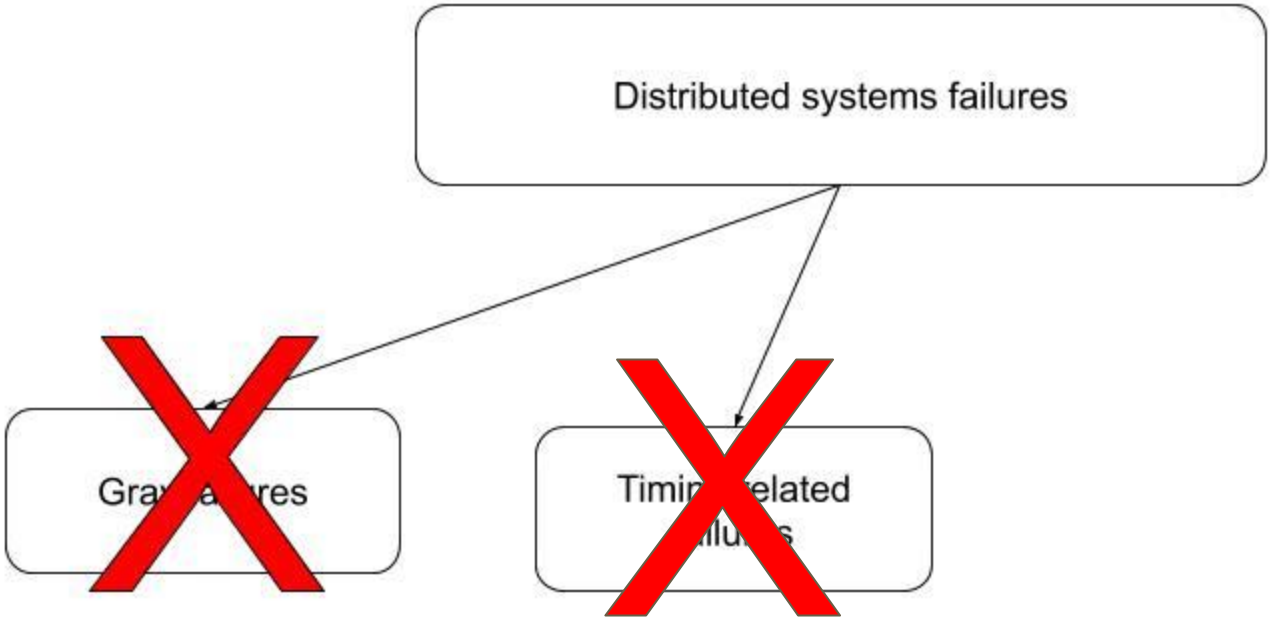

OverOps / Reliability Trend
Last 14 days
Env
OverOps Prod: S6295
Apps
All
Servers
All
Deployments
All
Top
10
Chart
Unique Events
HELP
Deployments: Unique Events
Applications: Unique Events
.v4.0.2108
.v4.0.2109
.v4.0.2110
.v4.0.2111
.v4.0.2112
.v4.0.2113
.v4.0.2115
.v4.0.2118
.api
.app
.service
.sqs-aa
.sqs-general
.sqs-stats
.sqs-subtask-general
Deploy
New
Inc
Slow
Avg Score
v4.0.2118
16 (2 p1)
4 p1
3
27.5
v4.0.2115
16 (1 p1)
3 p1
71.3
v4.0.2113
10
3 (2 p1)
87.5
v4.0.2112
23 (6 p1)
2 (1 p1)
1
86.3
v4.0.2111
13 (4 p1)
2 p1
27 (9 p1)
82.2
v4.0.2110
4
98.9
v4.0.2109
4
98.9
v4.0.2108
18 (1 p1)
15 p1
7 (4 p1)
89.3
App
New
Inc
Slow
Avg Score
api
48 (11 p1)
4 p1
1
87.9
app
23 (2 p1)
2
1
95.0
service
19
3 p1
2
95.2
sqs-aa
5
99.1
sqs-general
11 (1 p1)
3 p1
96.8
sqs-stats
100.0
sqs-subtask-general
100.0

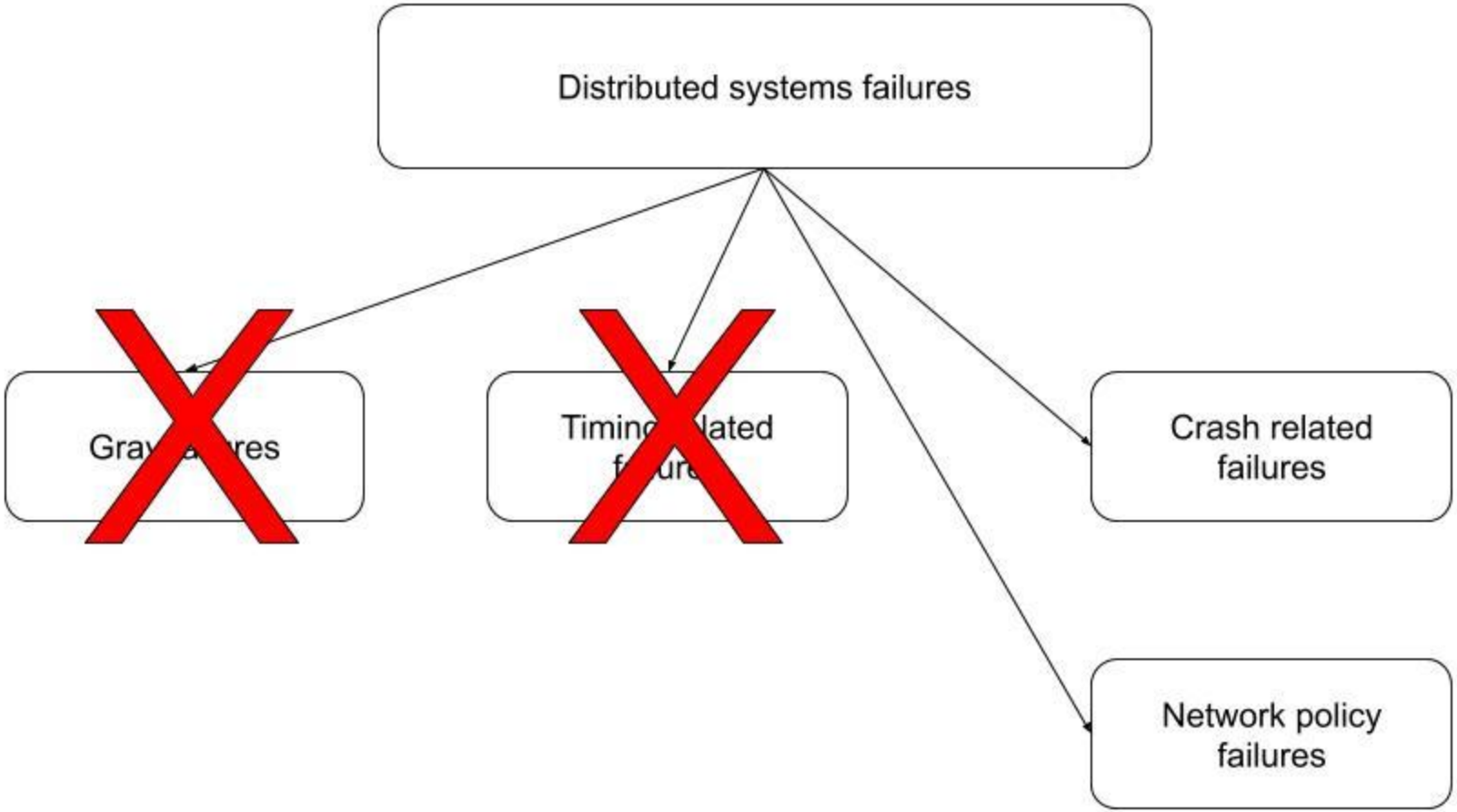

VALET Dashboard
Home
Reports
Search
Registration
About
Logout
Example Product
Example Service
Example Service
Start Date
01/31/2018
End Date
02/06/2018
Get Report
VOLUME
AVAILABILITY
LATENCY
ERRORS
TICKETS
Total Request Per Day
Peak TPS
Overall Availability
90th Percentile(ms)
95th percentile(ms)
5xxs (%)
Overall Tickets
SLOs
15,000,000
500
99.96
750
1,000
0.04
0
2018-01-31
4,135,564
89.9
100%
496.58
884.47
0
0
2018-02-01
4,050,507
93.53
99.99%
491.45
892.83
0.01
0
2018-02-02
4,051,150
81.25
100%
445.86
842.1
0
0
2018-02-03
4,238,050
81.17
100%
437.92
827.31
0
0
2018-02-04
4,343,915
144.48
99.3%
857.52
1,194.88
2.4
2
2018-02-05
4,616,110
109.8
100%
474.43
864.79
0
0
2018-02-06
4,304,866
98.63
99.99%
484.78
871.01
0.01
0
Download CSV

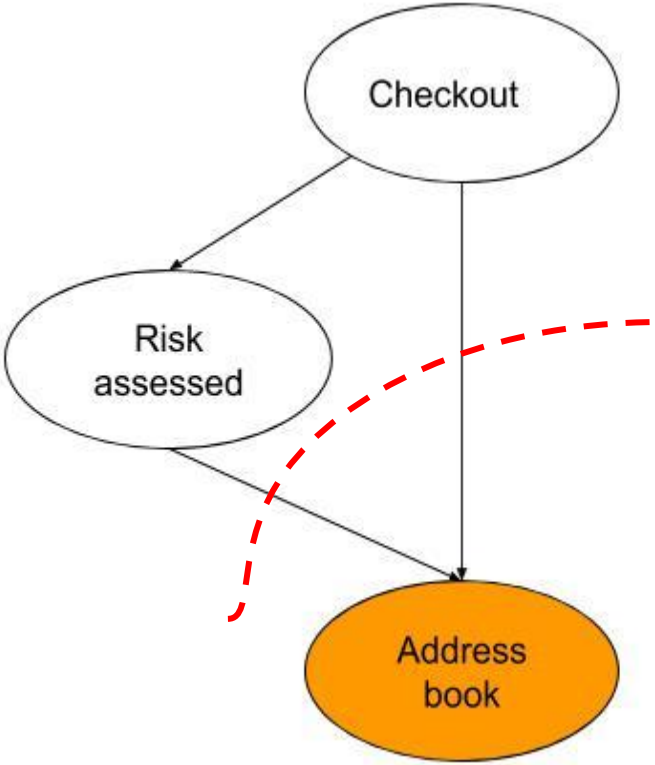

102
18
0
2
5

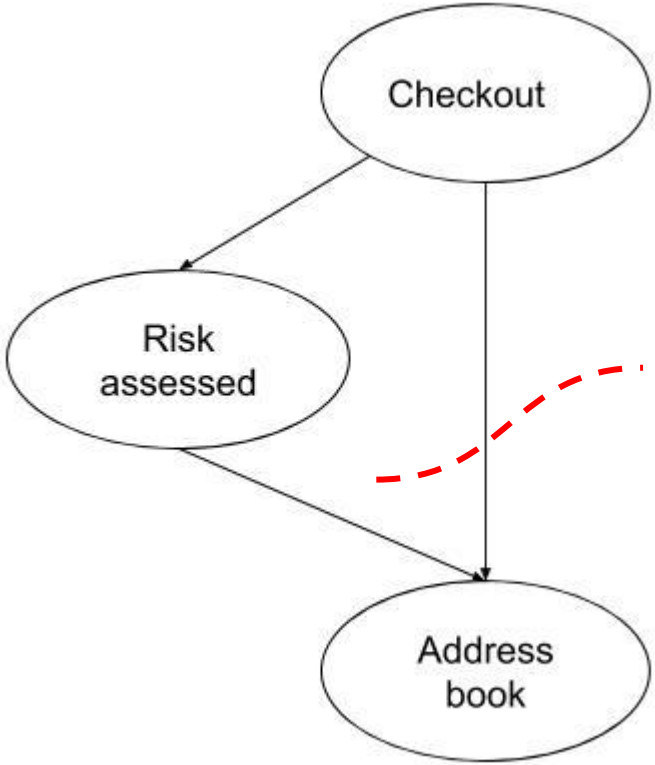

Load balancer
Checkout
Address book
Discounts
Add cart
Product details
Payment details
User credentials
Compute tax
Compute shipping
Personalization
Estimate delivery


Distributed systems failures
Gray ures


Distributed systems failures


Distributed systems failures
Crash related failures
Network policy failures


Checkout
Risk assessed
Address book

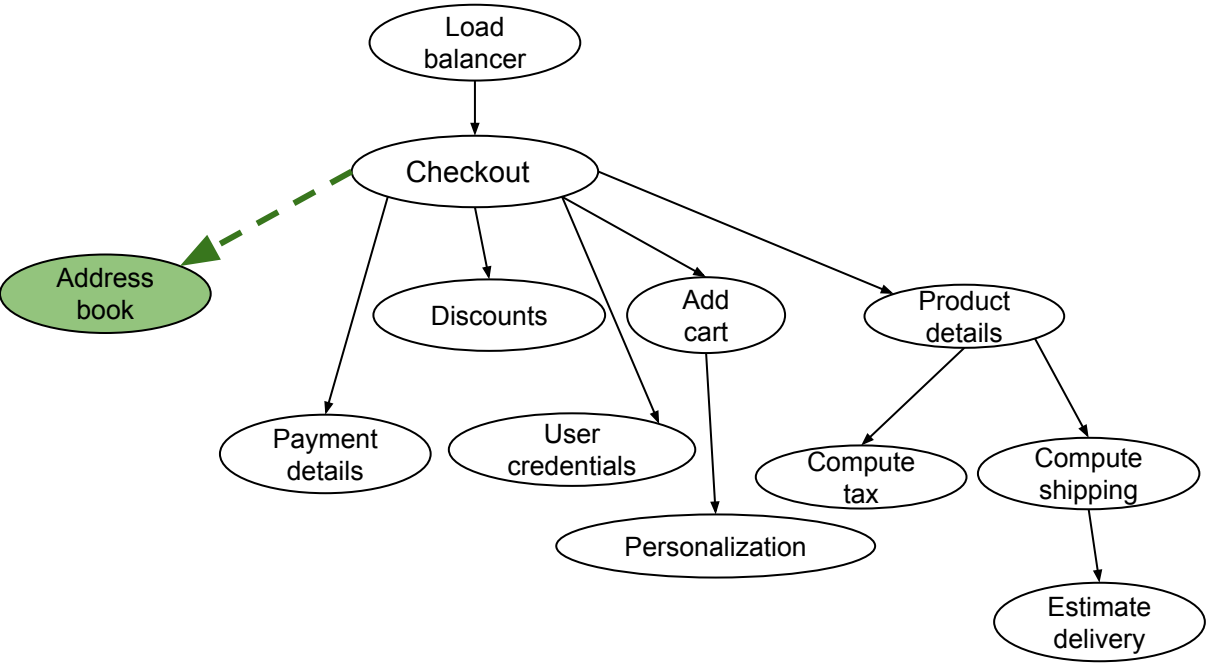

Checkout
Risk assessed
Address book

After finding an example anomalous execution, operators might ask:

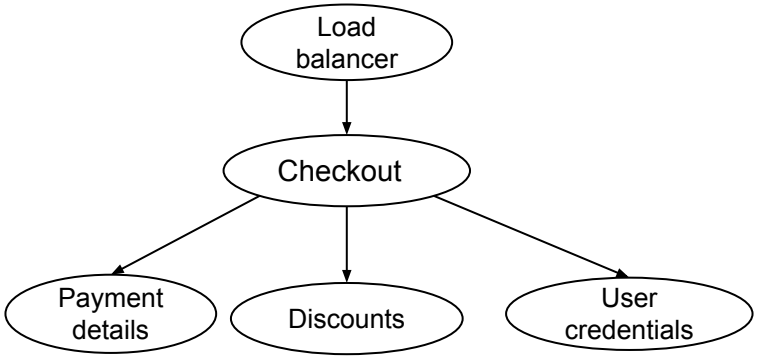
*How does *this* anomalous execution differ from those seen in steady state?*

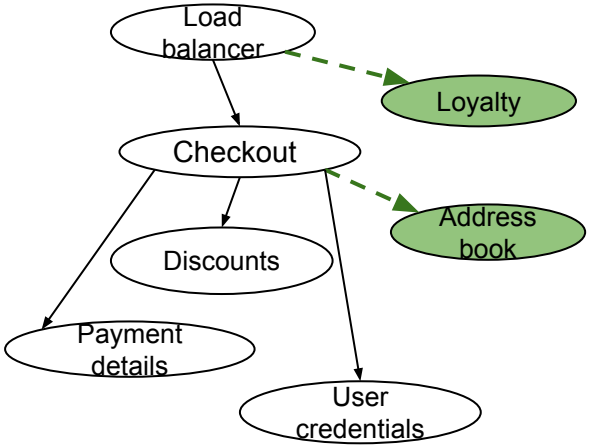
- Why traces?
  - Increasingly deployed in industry
  - Structure of traces encapsulates chain of interactions

- System traces contain the answer to this question!

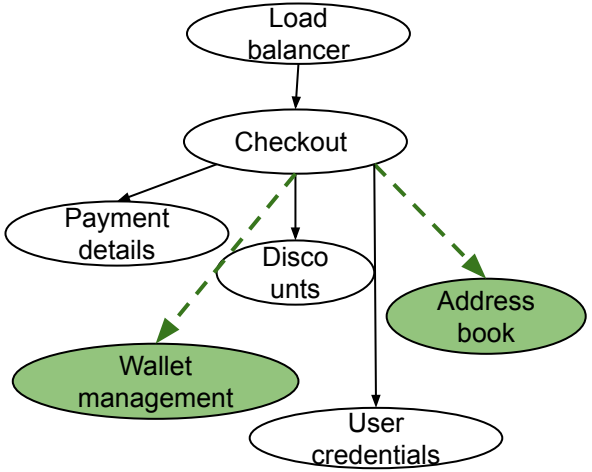


1. Get trace of an anomalous execution
2. Get trace of **one** successful execution
3. Difference the two



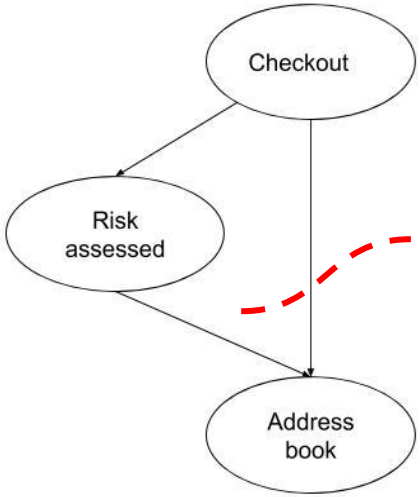


Add a **minimal** subgraph to the underlying graph of the failed execution trace so that it succeeds

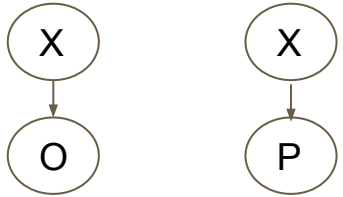


Naive approaches

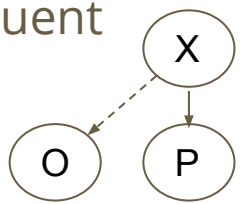
Count nodes:
O(N*M)



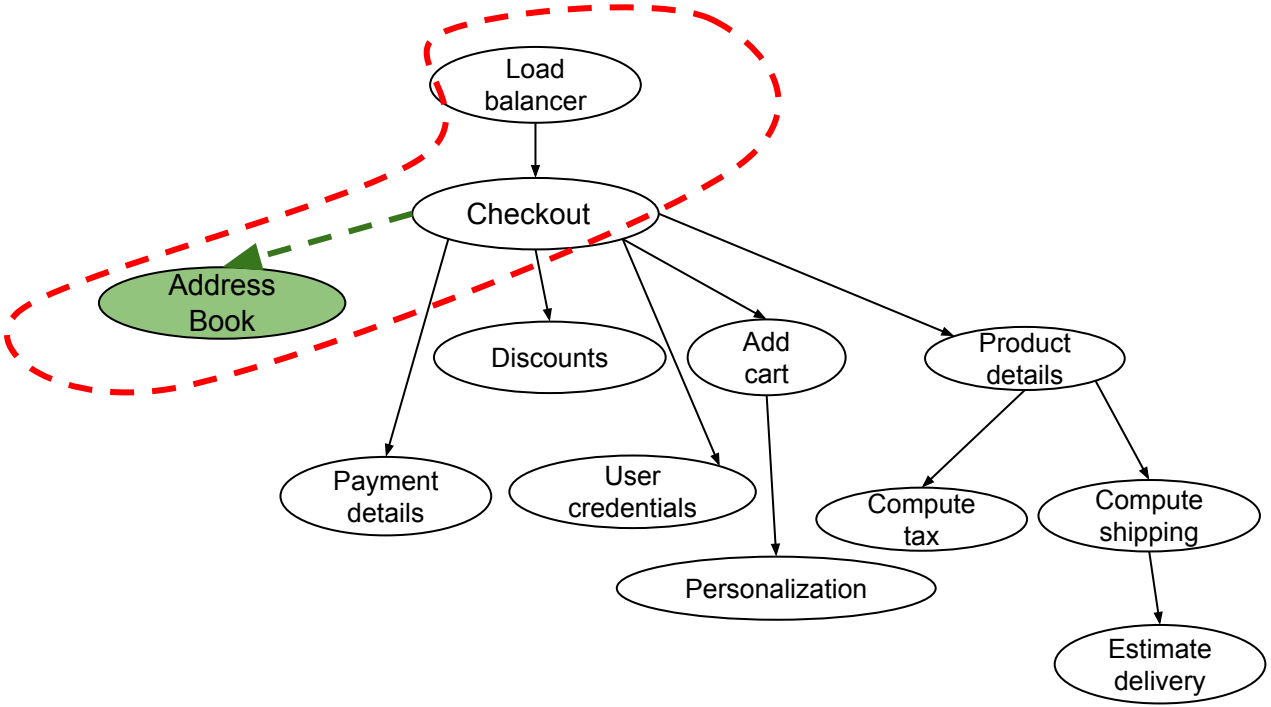
Markov models: O(N)

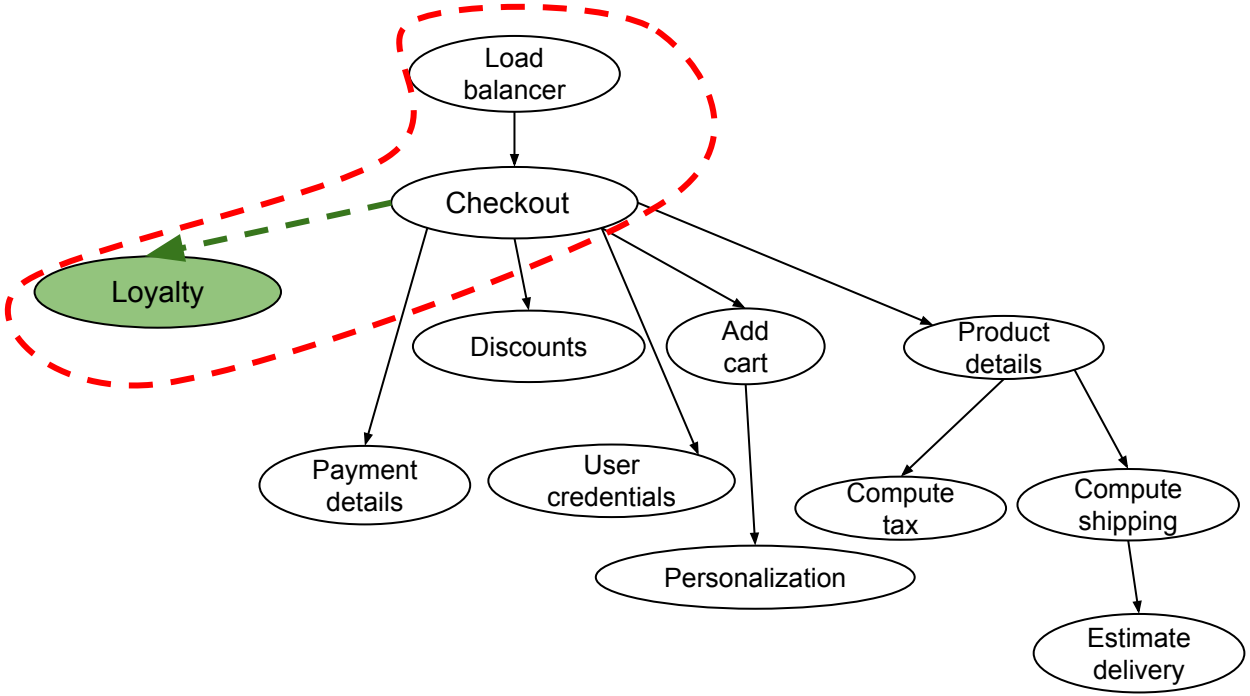


If X->O is 10X more frequent

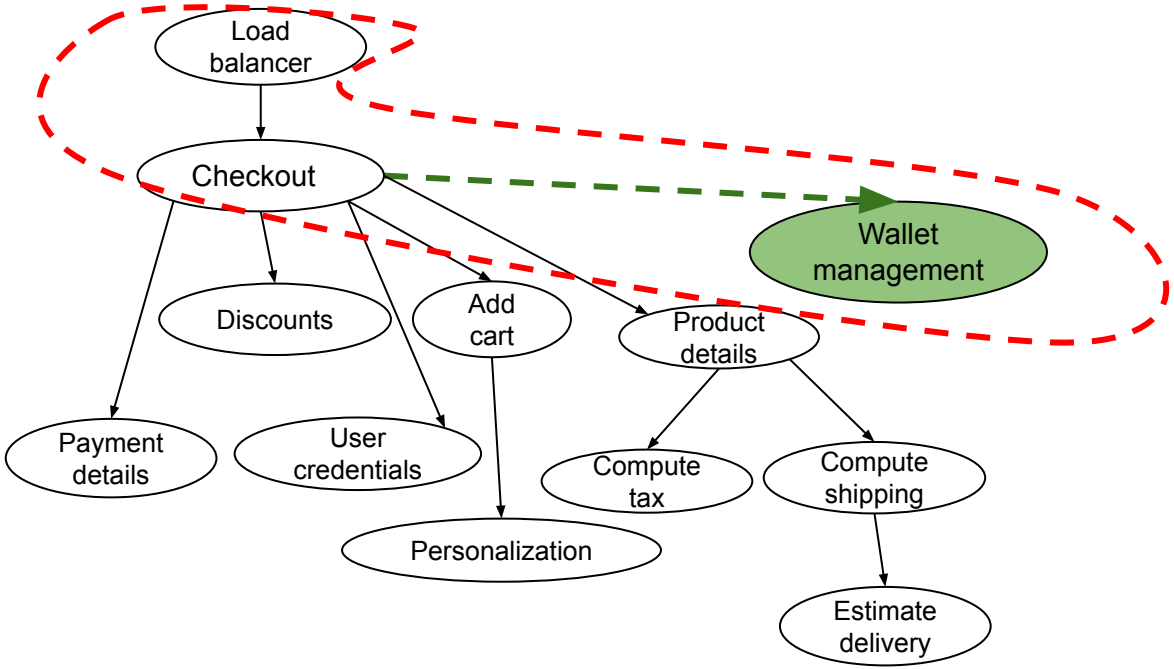


An alternate perspective

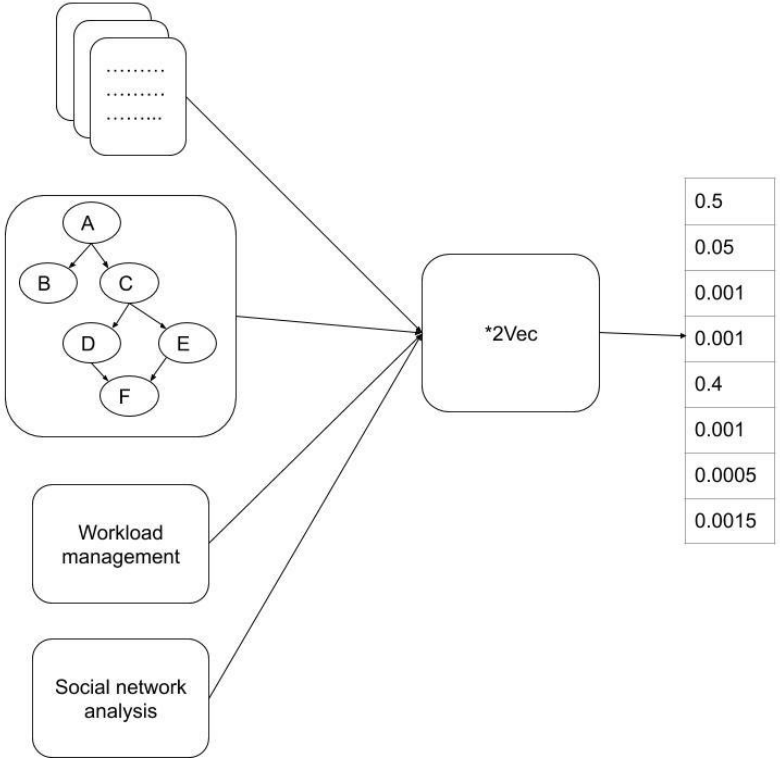



Load balancer
Checkout
Loyalty
Discounts
Add cart
Product details
Payment details
User credentials
Compute tax
Compute shipping
Personalization
Estimate delivery

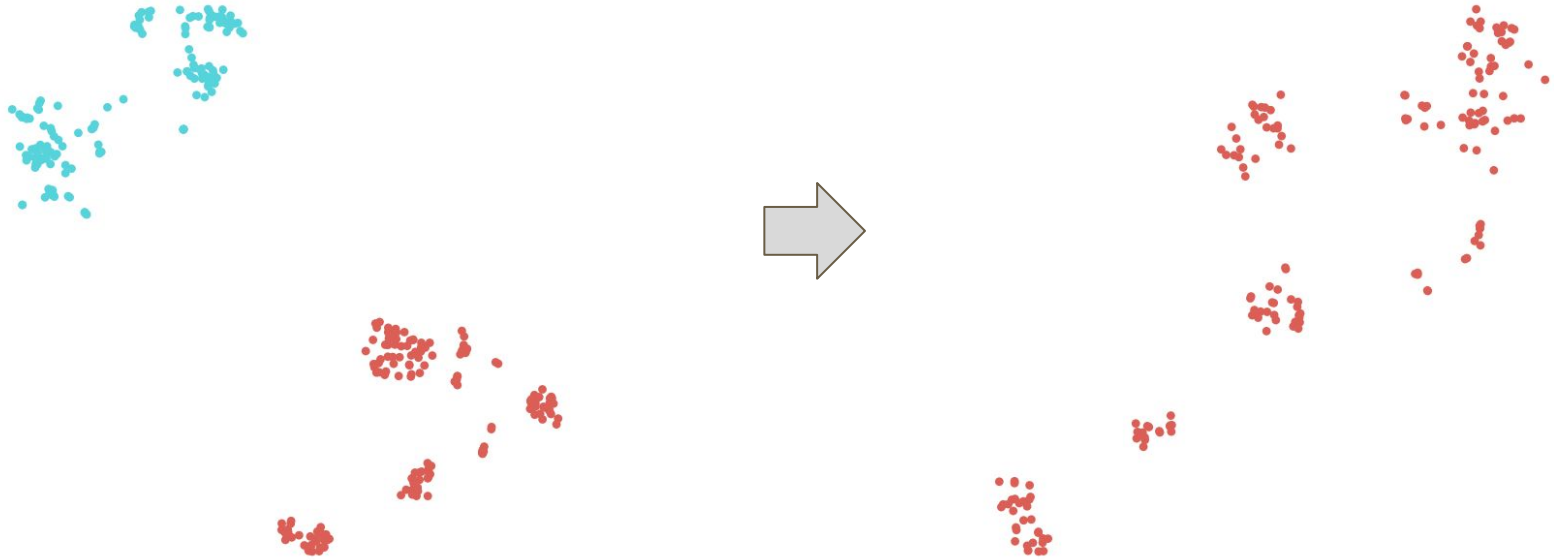
- Paths in traces: Sentences of a document
- Traces: Documents
- Traces are the language of distributed systems
- Could we use techniques from NLP?



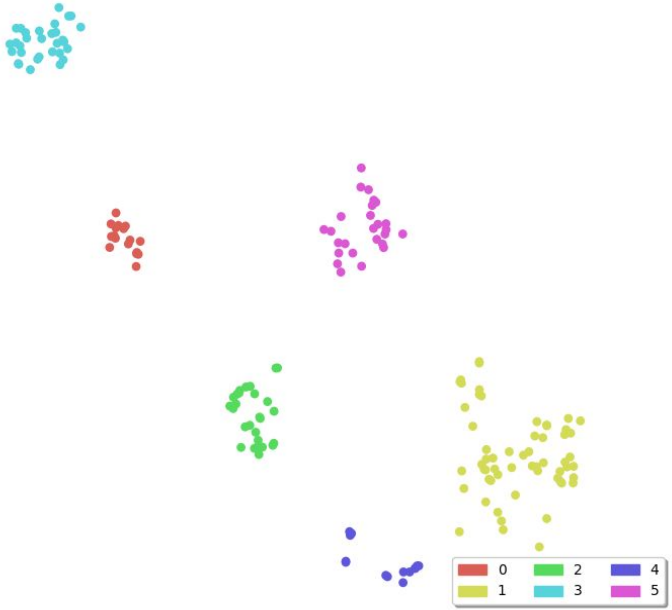
*2vec (Background)



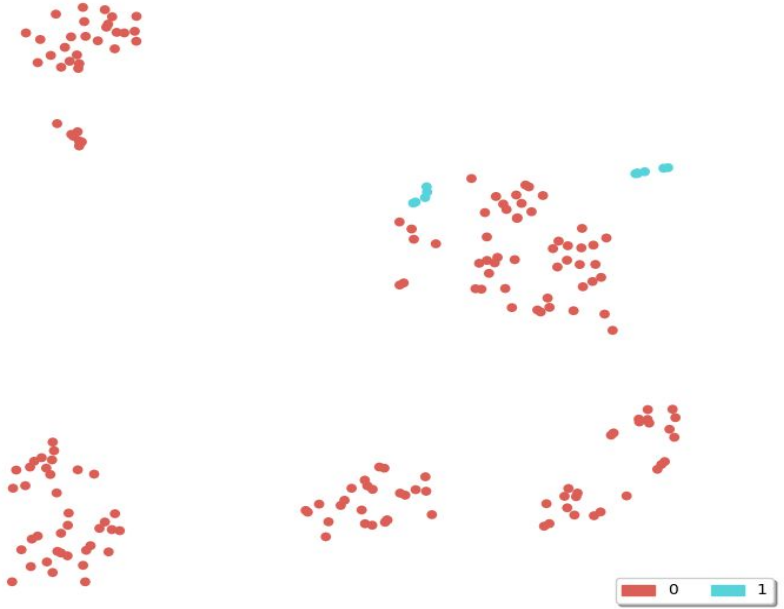
Doc2Vec



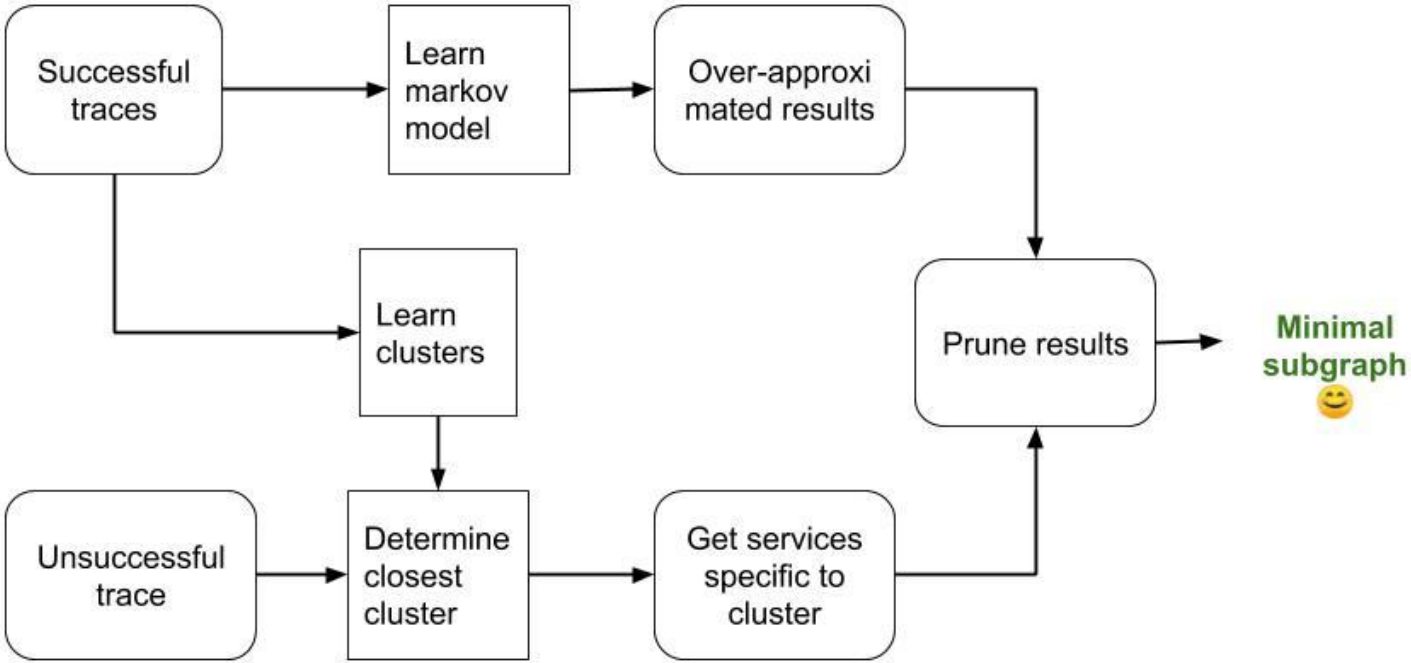
Clusters within a request type are automatically found using K-Means



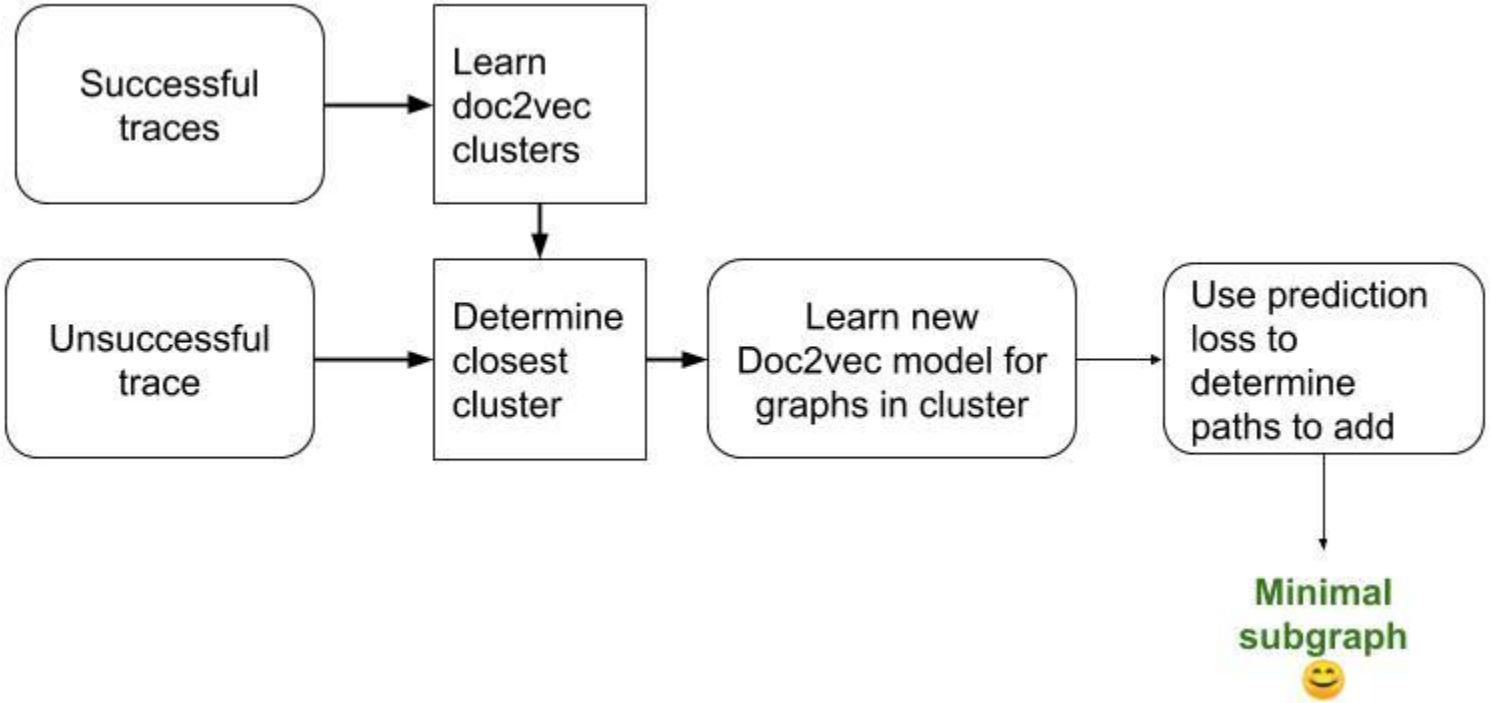
The blue points represent graphs from failed executions. These are all close to one cluster



Option #1


Successful traces
Learn markov model
Over-approxi mated results
Learn clusters
Prune results
Minimal subgraph
Unsuccessful trace
Determine closest cluster
Get services specific to cluster

Option #2


Successful traces
Learn doc2vec clusters
Unsuccessful trace
Determine closest cluster
Learn new Doc2vec model for graphs in cluster
Use prediction loss to determine paths to add
Minimal subgraph

Your thoughts?

kamala.ramas@ucsc.edu