Diagnosis and Automation for Deployed Distributed Systems

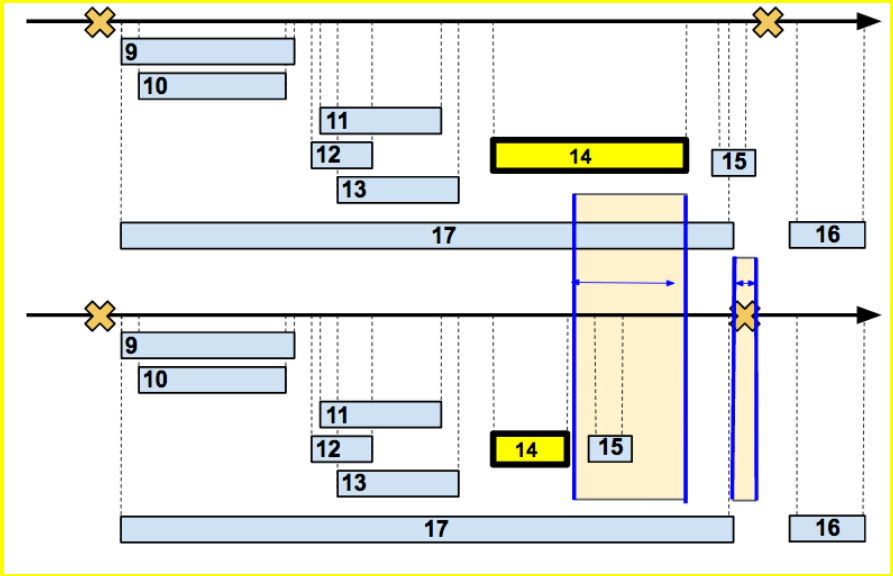
Google™

The Weatherman Effort

Gideon Mann, Mark Sandler, Eyal Even-dar, Darja Krushevskaja, Sudipto Guha, Krzysztof Ostrowski, Sebastian Pueblas, Lev Ratinov, Eran Gabber

D ation f ed S

Today: Profiling Latency in Deployed Distributed Systems



Google™

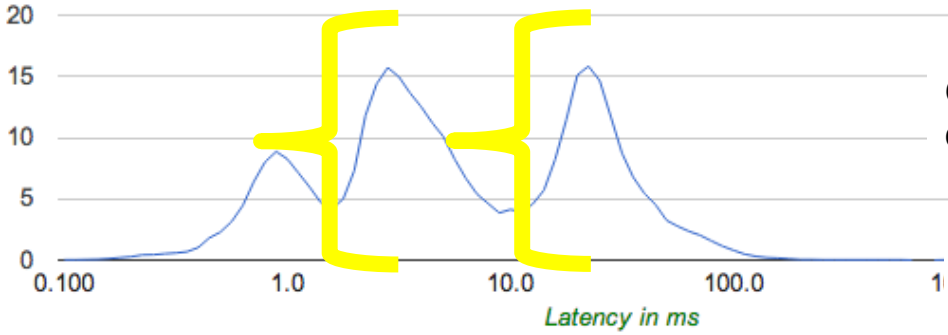
The Weatherman Effort

Gideon Mann, Mark Sandler, Eyal Even-dar, Darja Krushevskaja, Sudipto Guha, Krzysztof Ostrowski, Sebastian Pueblas, Lev Ratinov, Eran Gabber

Google

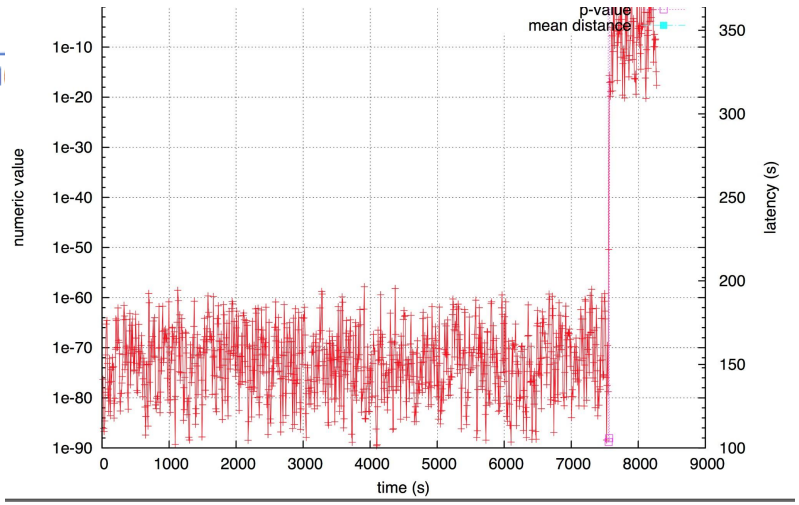
Two Scenarios of Interest

Latency distribution has multiple peaks, at different orders of magnitude.

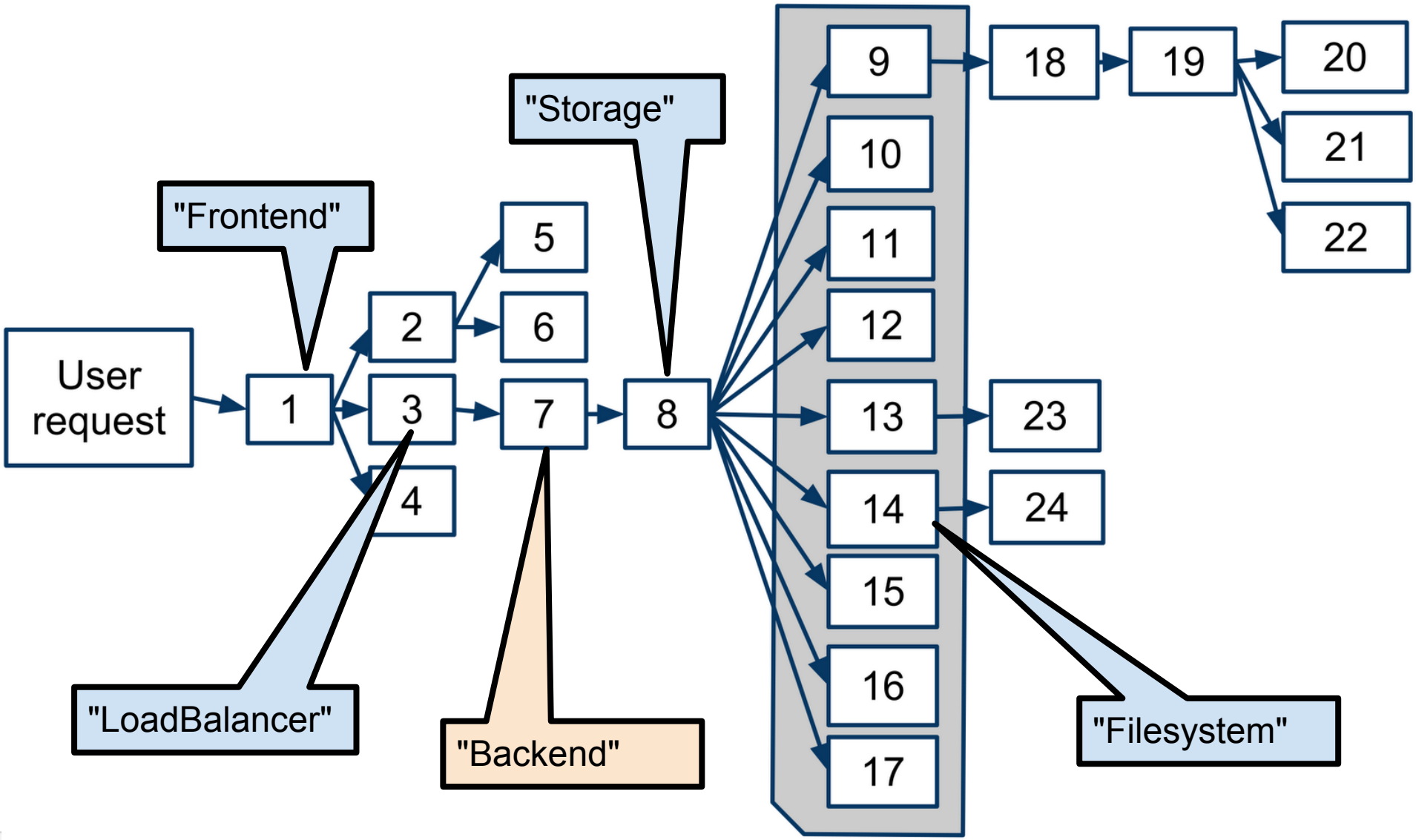


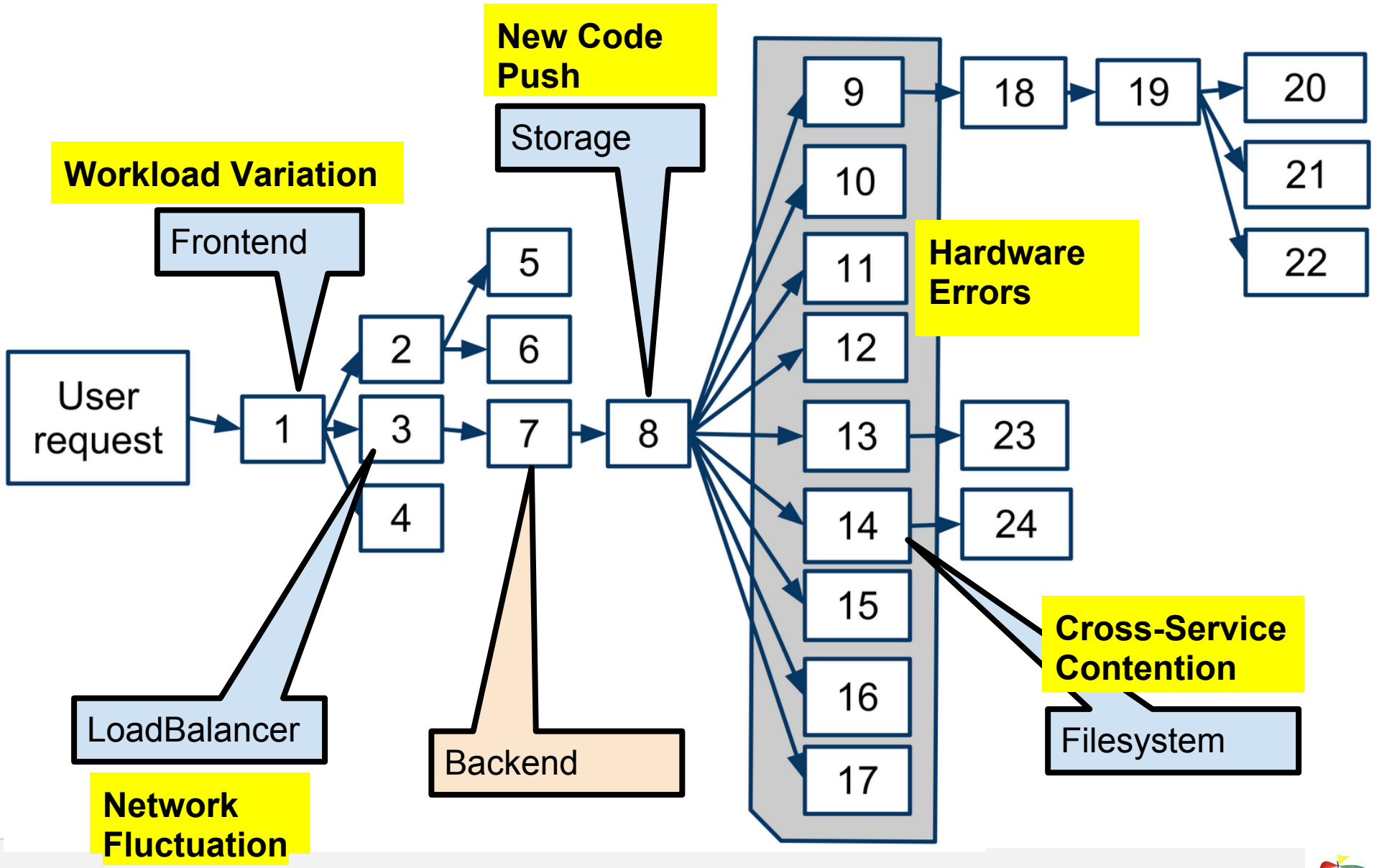
Load-testing isn't enough.

Over time, the latency has dramatically changed ... what happened?



Well, probably a lot has changed. What's the most significant?

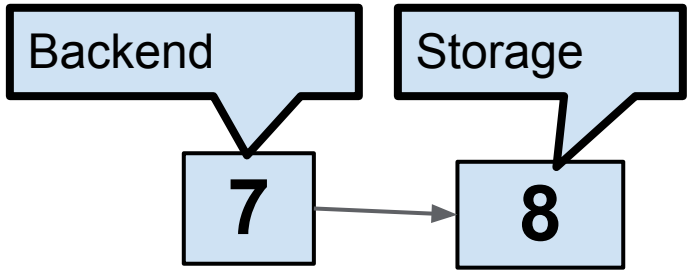

User request
1
"Frontend"
2
3
4
5
6
7
"Storage"
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
"LoadBalancer"
"Backend"
"Filesystem"

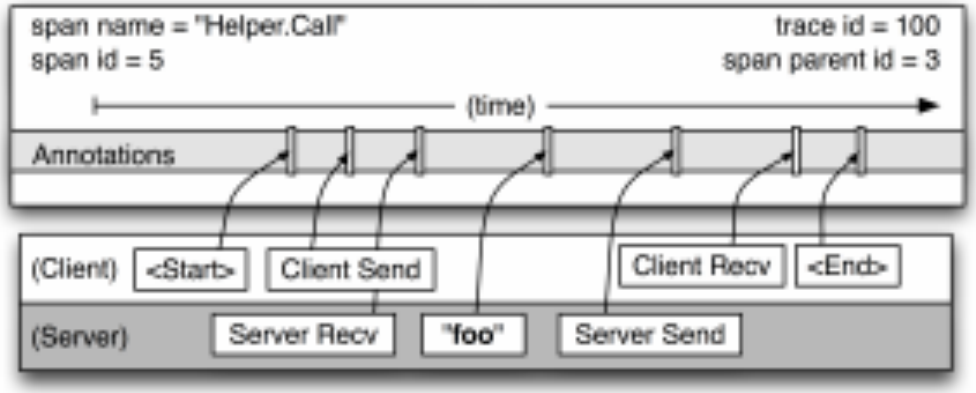

New Code Push
Storage
Workload Variation
Frontend
User request
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Hardware Errors
Cross-Service Contention
Filesystem
LoadBalancer
Network Fluctuation
Backend

Google

Dapper Distributed Trace Collection

Sigelman, Barroso, Burrows, Stephenson,
Plakal, Beaver, Jaspan, Shanbhag '10

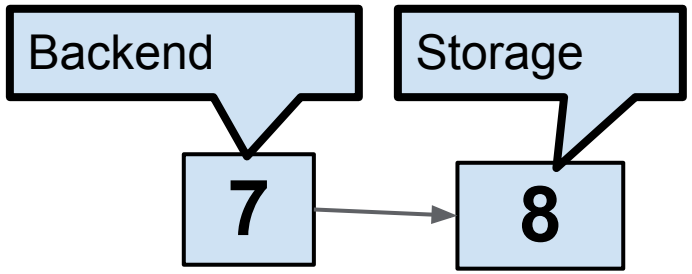


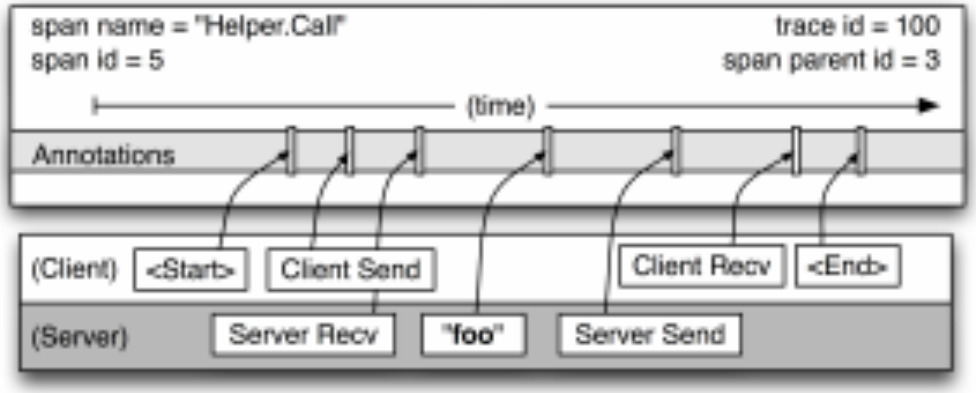


Google

Dapper Distributed Trace Collection

Sigelman, Barroso, Burrows, Stephenson, Plakal, Beaver, Jaspan, Shanbhag '10





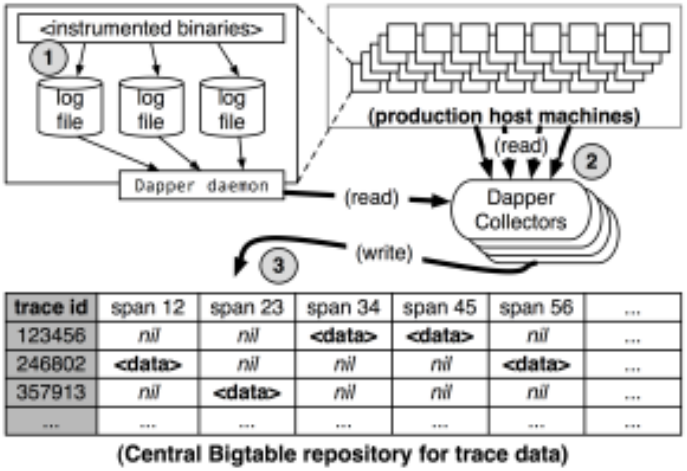
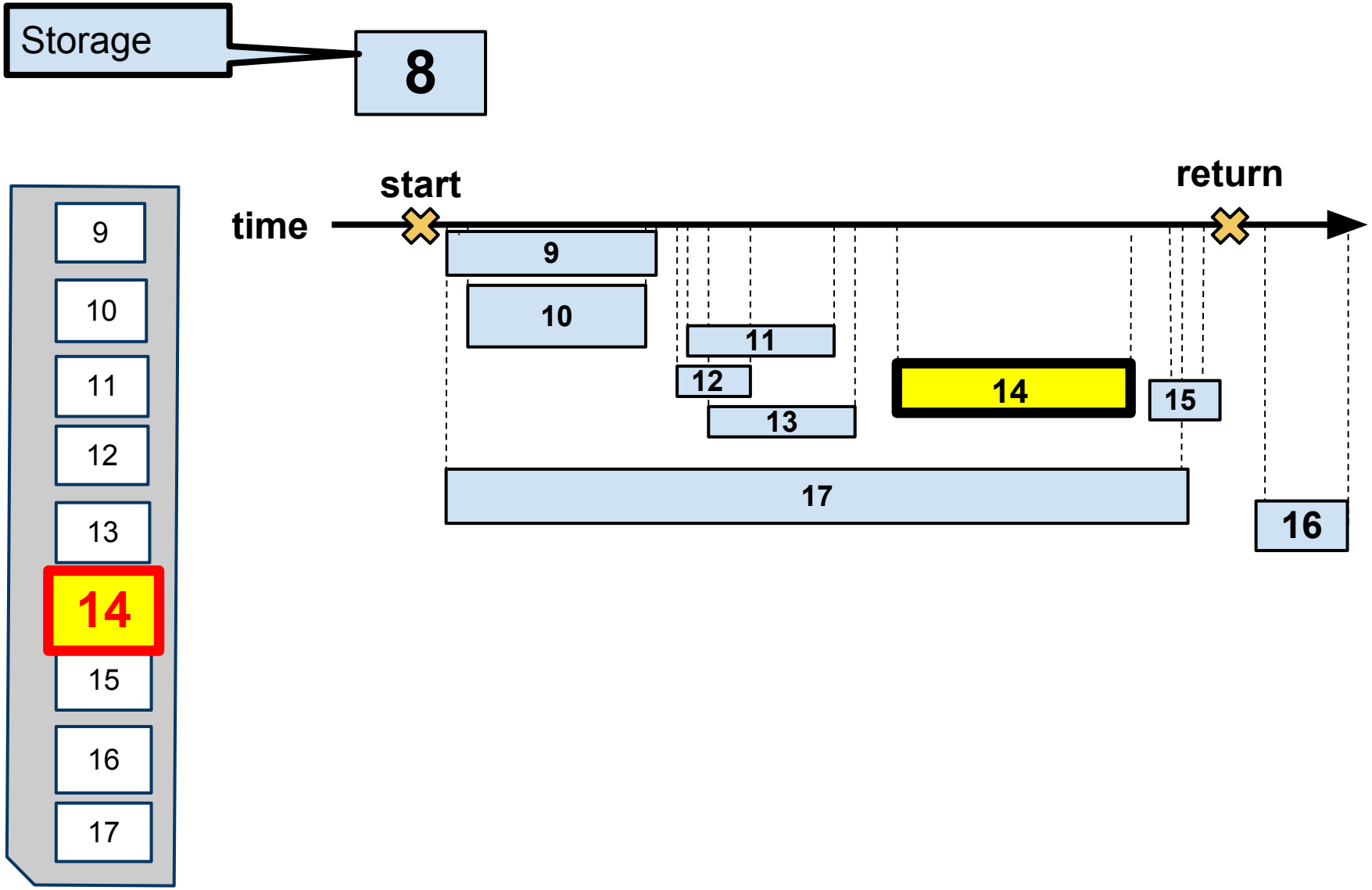
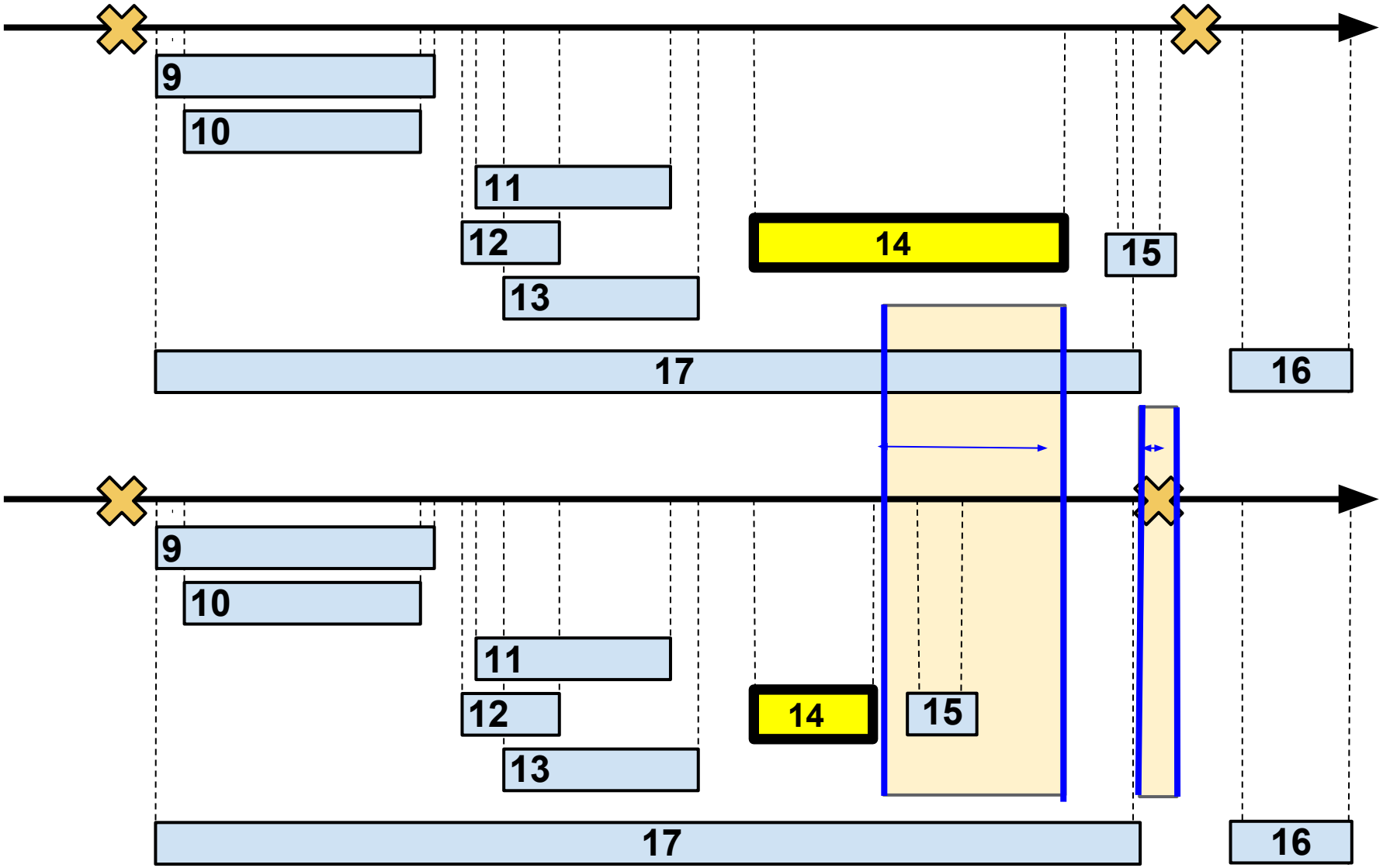


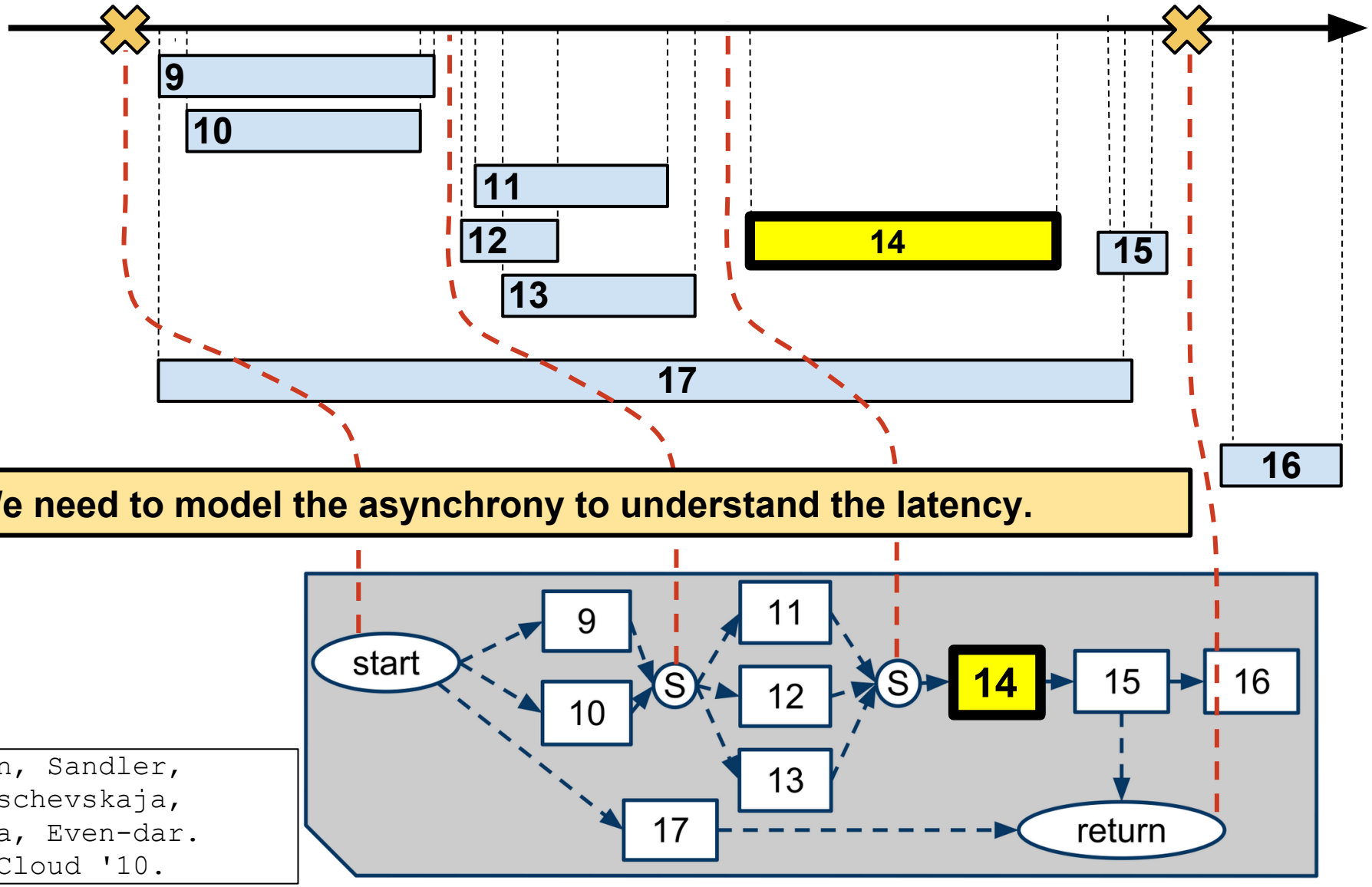
Figure 5: An overview of the Dapper collection pipeline.


Storage
8
9
10
11
12
13
14
15
16
17
start
return
time
9
10
11
12
13
14
15
17
16

Google "If time in 14 is reduced, what is the predicted latency?"



Google



Mann, Sandler, Kruschevskaja, Guha, Even-dar. HotCloud '10.

λ_t observed latency for RPC t

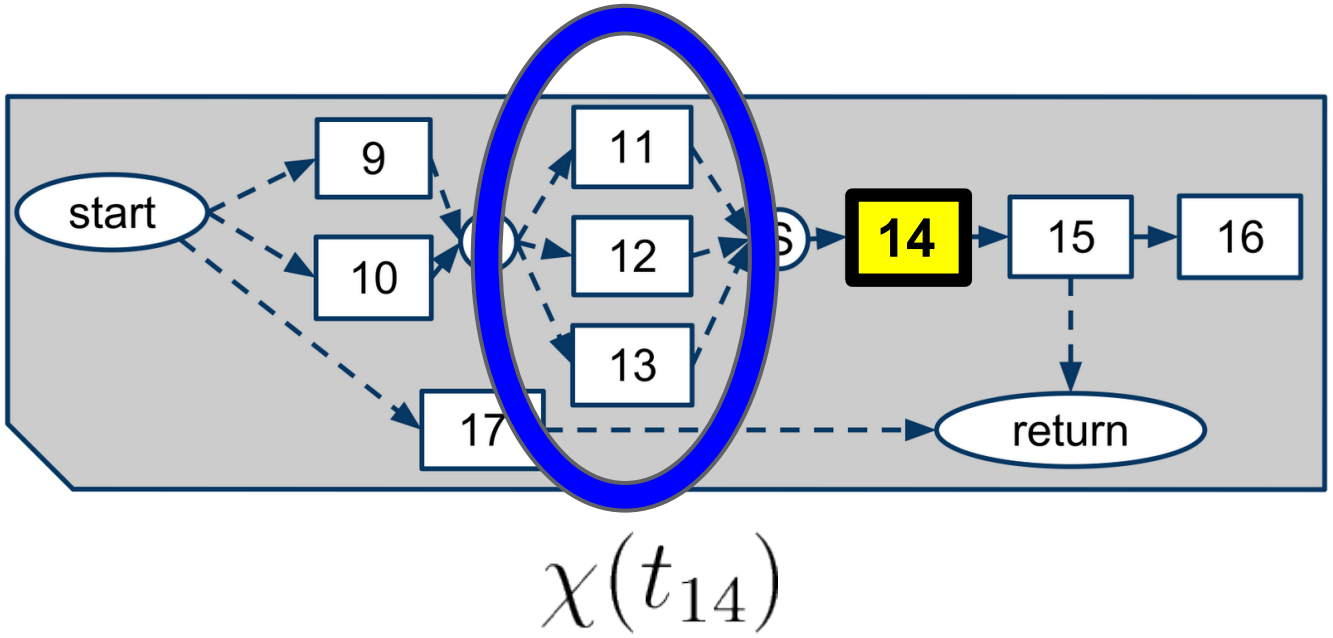
$\hat{\lambda}_t$ predicted latency for RPC t

We predict the latency of a given RPC given the latency of its children C(t), and relevant features of the RPC t and children.

$$\hat{\lambda}_t = \Psi(f_t, f_{C(t)}, \lambda_{C(t)})$$

χ An (asynchronous) execution model

$\chi(t_i)$ The blocking set for a RPC t_i



$\chi(t_{14})$

χ An (asynchronous) execution model

$\chi(t_i)$ The blocking set for a RPC t_i

$$\psi_{\chi(t_i)} = \lambda_{t_i} + \max_{t \in \chi(t_i)} \psi(t)$$

The finishing time for a RPC t_i is it's own latency and the finishing times of the RPCs in its blocking set.

$$\begin{aligned}\hat{\lambda}_p &= \Psi(f_p, f_{C(p)}, \lambda_{C(p)}) \\ &= \Psi(\chi, \lambda_{C(p)}) \\ &= \psi_\chi(t_{return})\end{aligned}$$

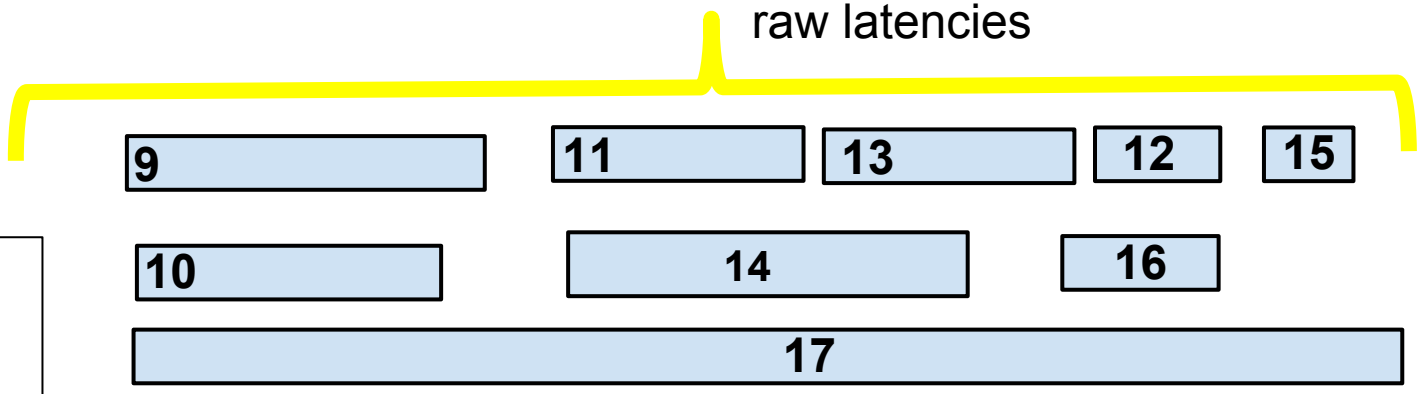
Given the children latencies and the execution model, we can compute the latency for a given parent RPC.

Google

Find best matching flow via Nearest Neighbor :

$$\lambda_{t_i}^{\chi} \sim N(\bar{\lambda}_{t_i}, \bar{\sigma}_{t_i})$$

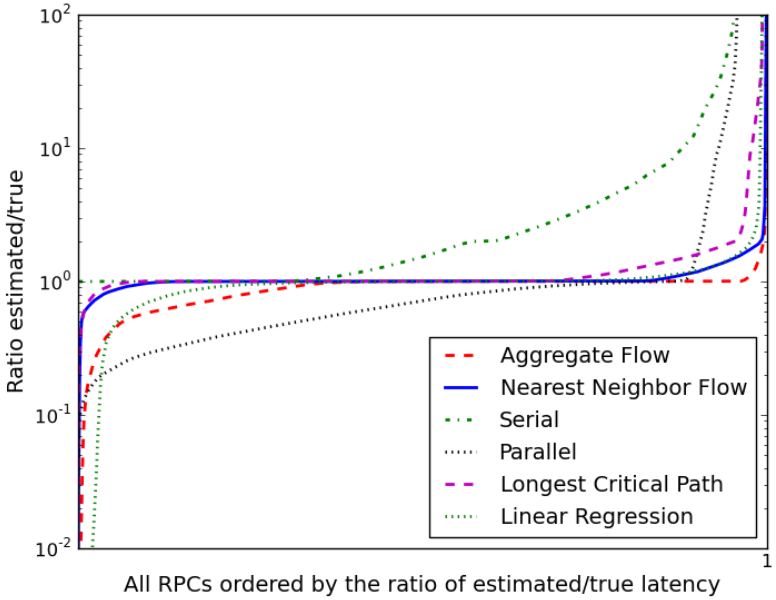
$$\log P_{\chi}(\lambda_t) \propto \sum_{t_i \in C(t)} (\bar{\lambda}_{t_i} - \lambda_{t_i})/\bar{\sigma}_{t_i}$$

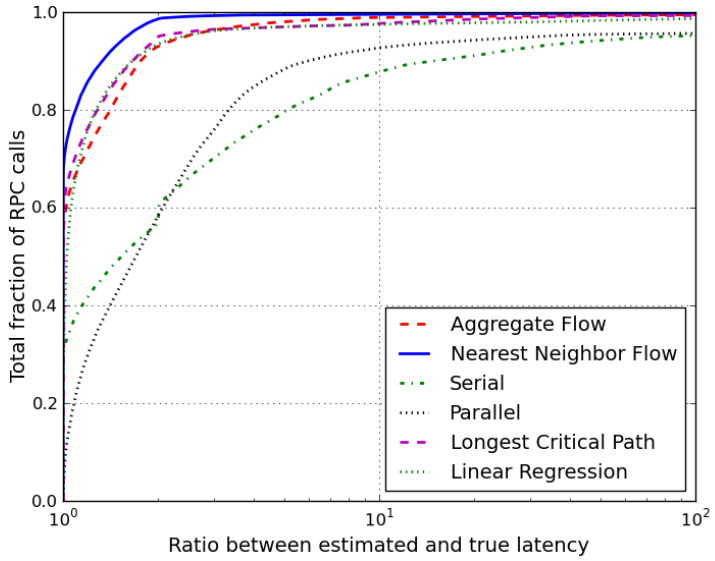


Mann, Sandler, Kruschevskaja, Guha, Even-dar. HotCloud '10.

Google

A perfect model would be a line at y=1.0





Induced models ("Nearest Neighbor Flow") are better predictors of parent latencies from children than:

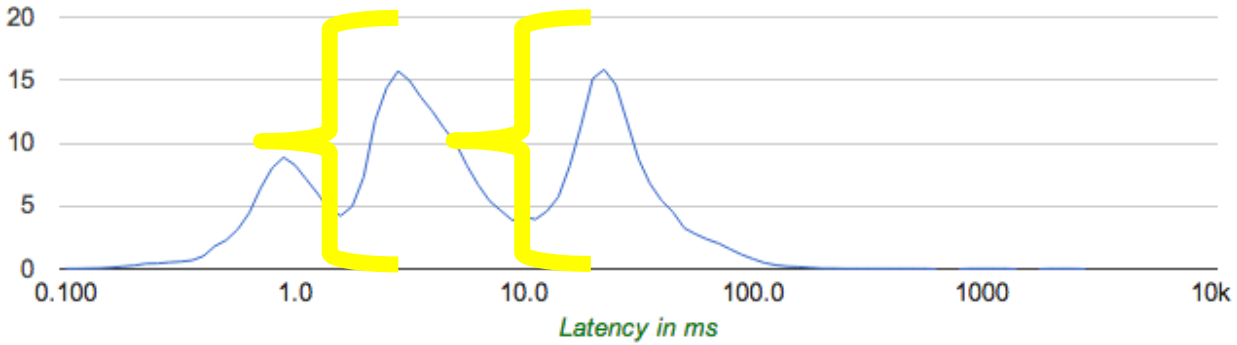
- Linear regression
- Longest critical path
- Either a full serial or full parallel RPCs

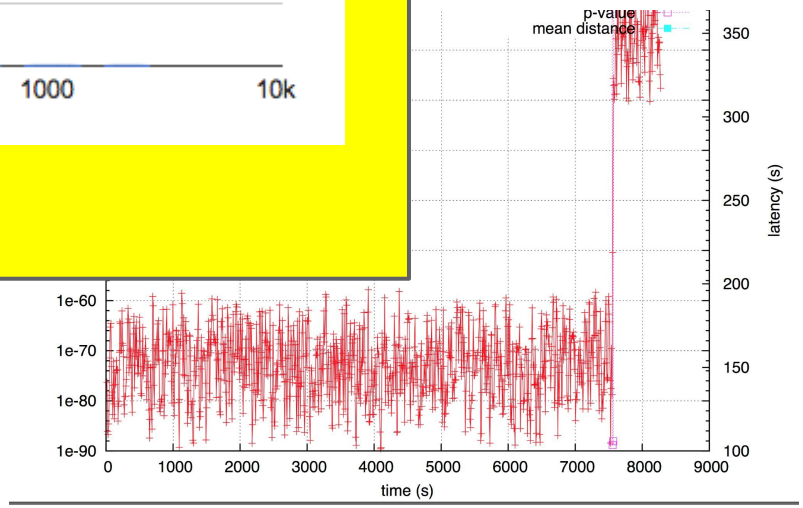
Mann, Sandler, Kruschevskaja, Guha, Even-dar. HotCloud '10.

Google

First Scenario of Interest

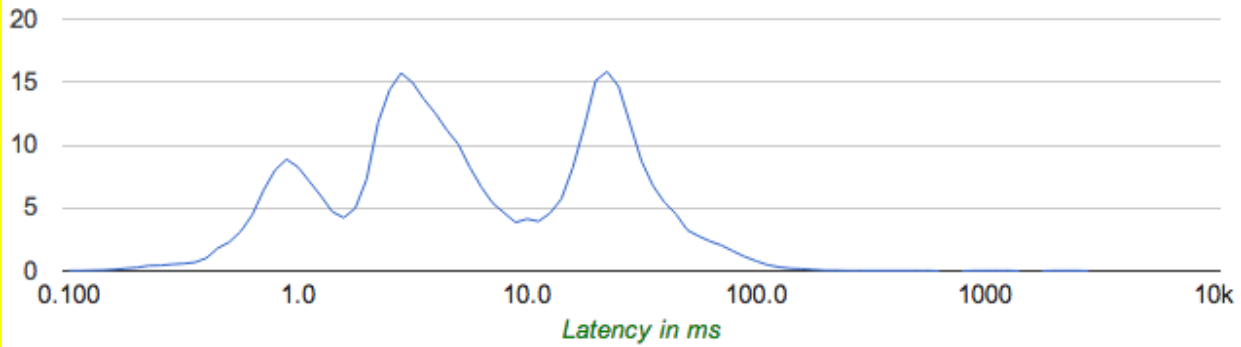
Latency distribution has multiple peaks at different orders of magnitude.





Google

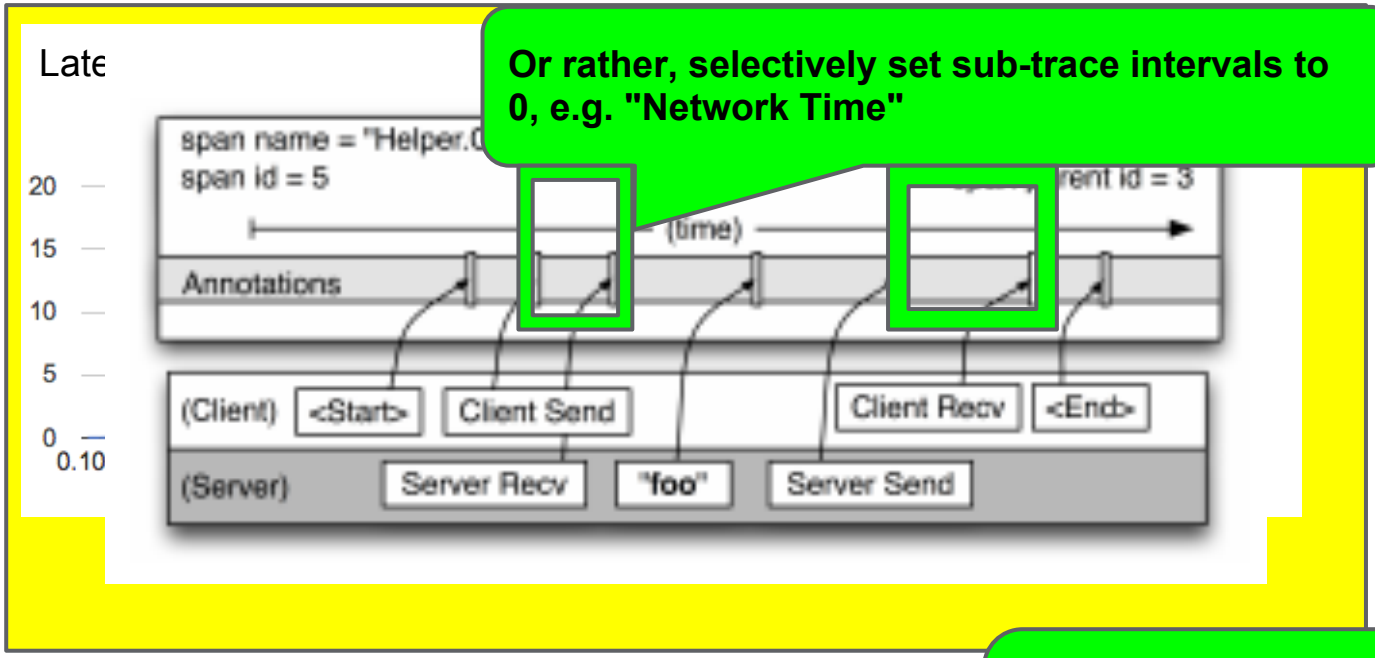
Latency distribution has multiple peaks at different orders of magnitude.



$$\Lambda_A = \mathbb{U}_{t \in A}(\lambda_t)$$

$$\Lambda_{\delta A} = \mathbb{P}_{t \in A}(\Psi(\chi, \lambda_{\delta C(t)}))$$

Selectively set latencies of specific traces to 0, and then simulate the effects.



Or rather, selectively set sub-trace intervals to 0, e.g. "Network Time"

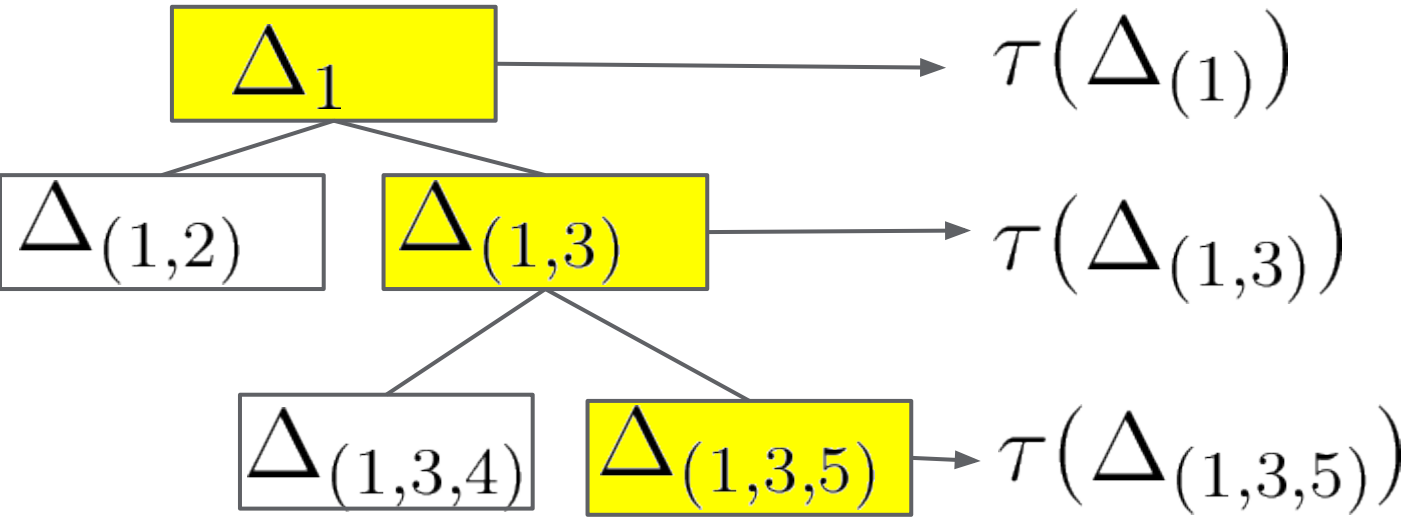
$$\Lambda_A = \mathbb{U}_{t \in A}(\lambda_t)$$

$$\Lambda_{\delta A} = \mathbb{P}_{t \in A}(\Psi(\chi, \lambda_{\delta C(t)}))$$

Selectively set latencies of specific traces to 0, and then simulate the effects.

Λ_A Random variable of Latency

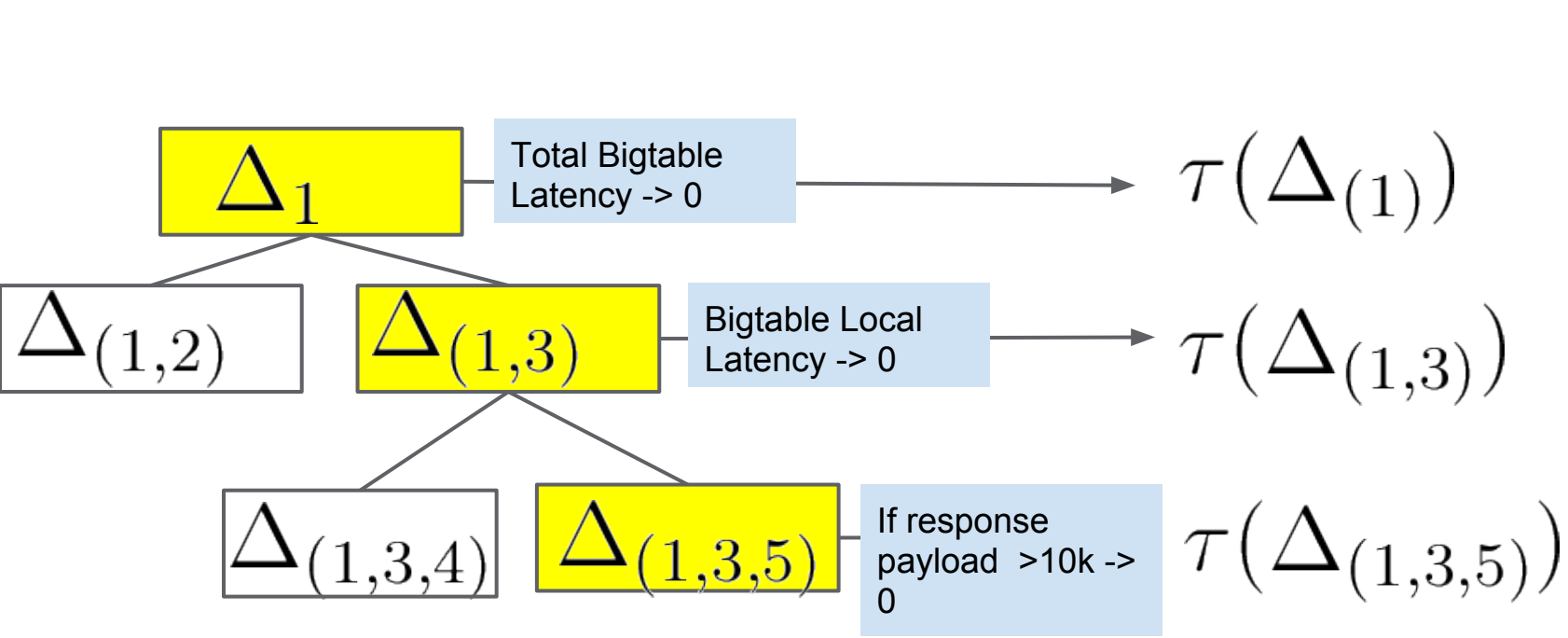
$$\tau(\Delta_j) = \mathbb{E}[\Lambda_{\Delta j A}]$$

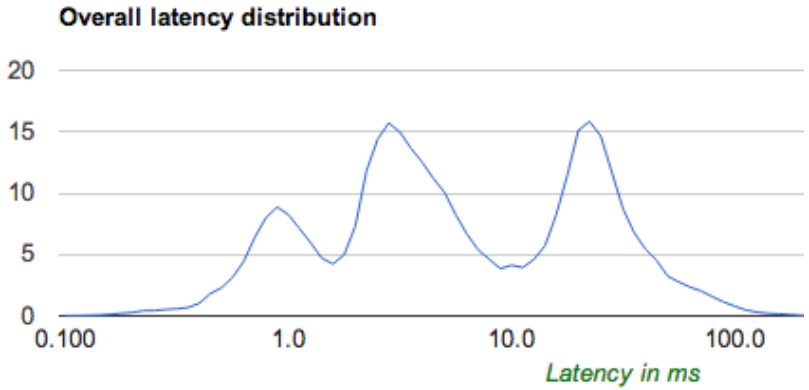


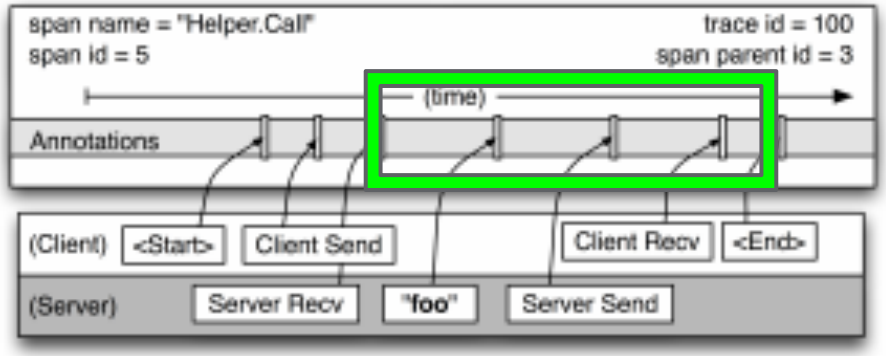
Λ_A Random variable of Latency

$$\tau(\Delta_j) = \mathbb{E}[\Lambda_{\Delta_j A}]$$

Mean Latency at 90% percentile

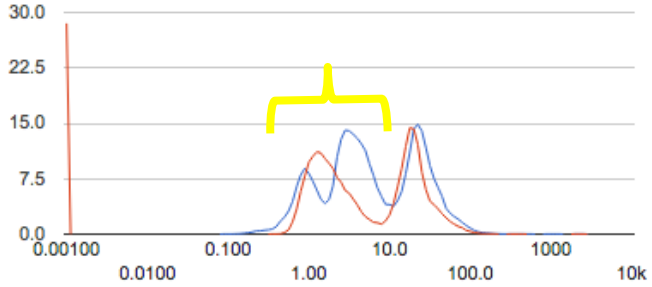


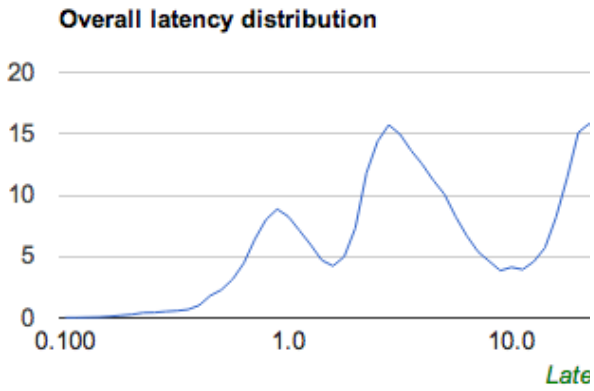


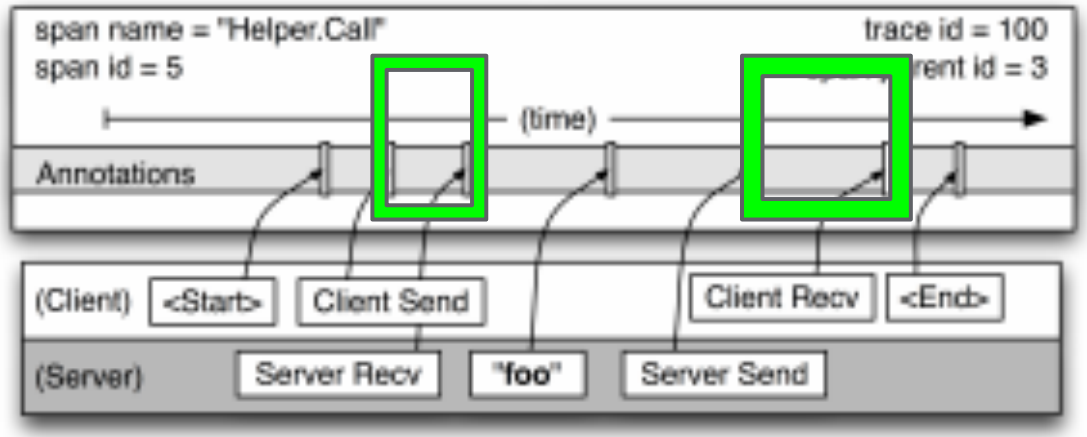


Red Line : top-level latency without time in ...

Bigtable Local Processing

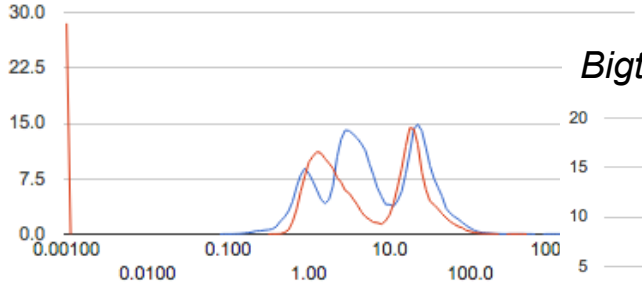




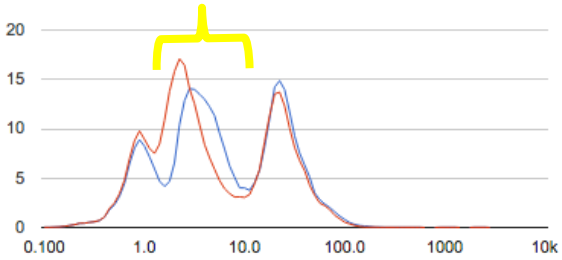


Red Line : top-level latency with

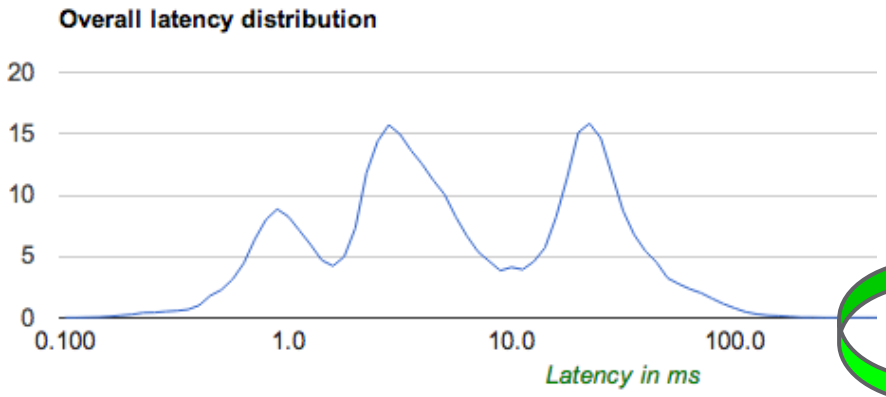
Bigtable Local Processing

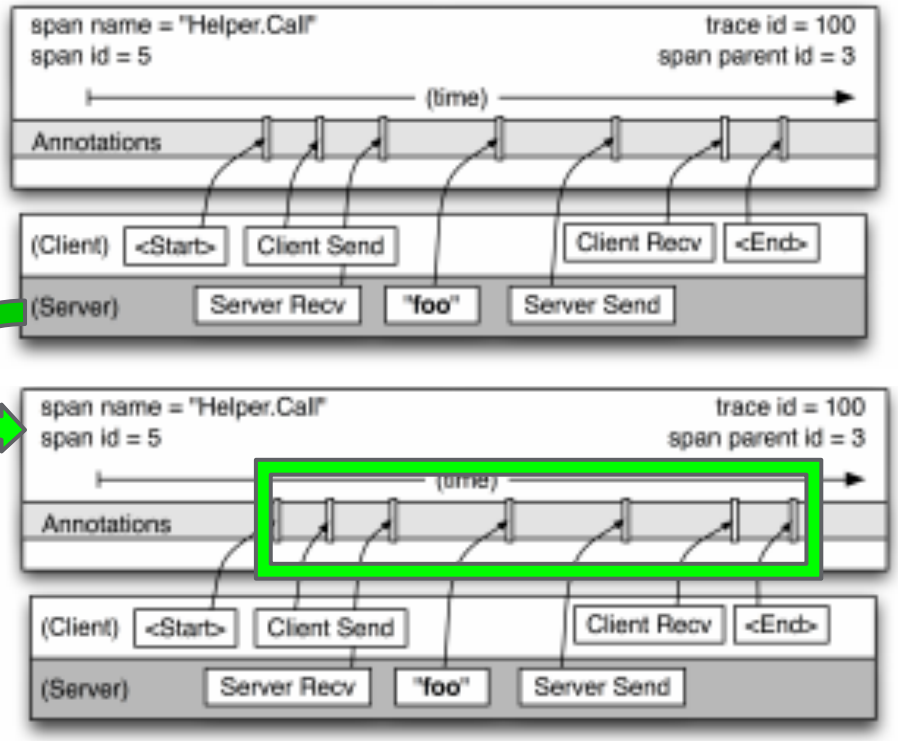


Bigtable Outgoing Network



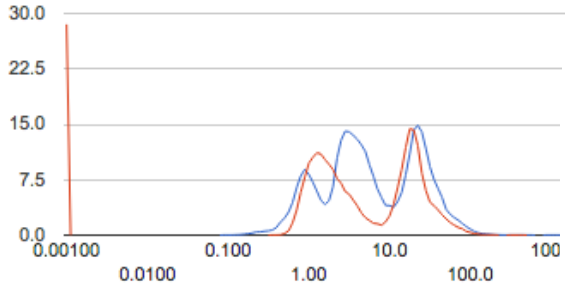
Google



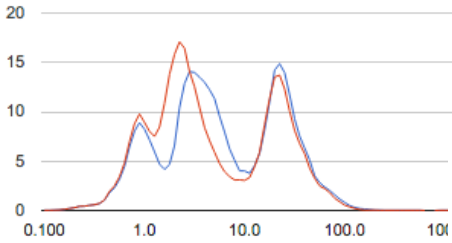


Red Line : top-level latency without time in ...

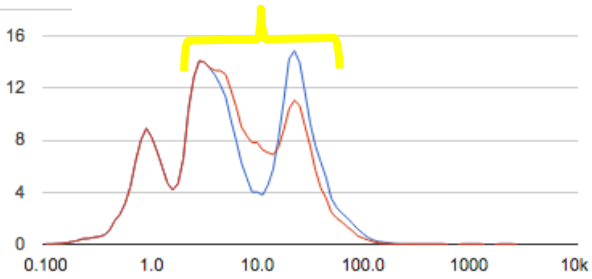
Bigtable Local Processing



Bigtable Outgoin

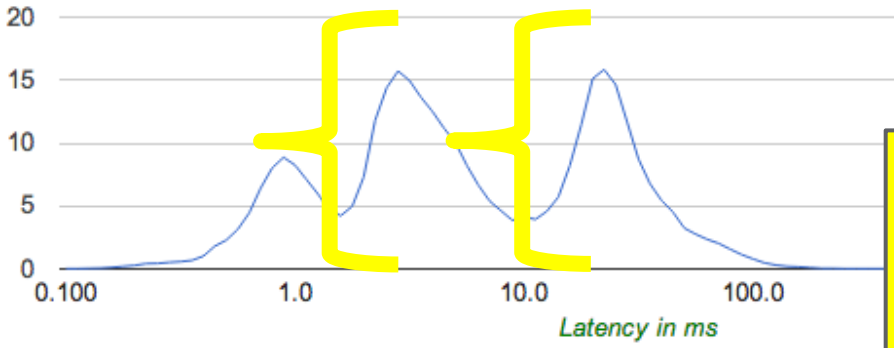


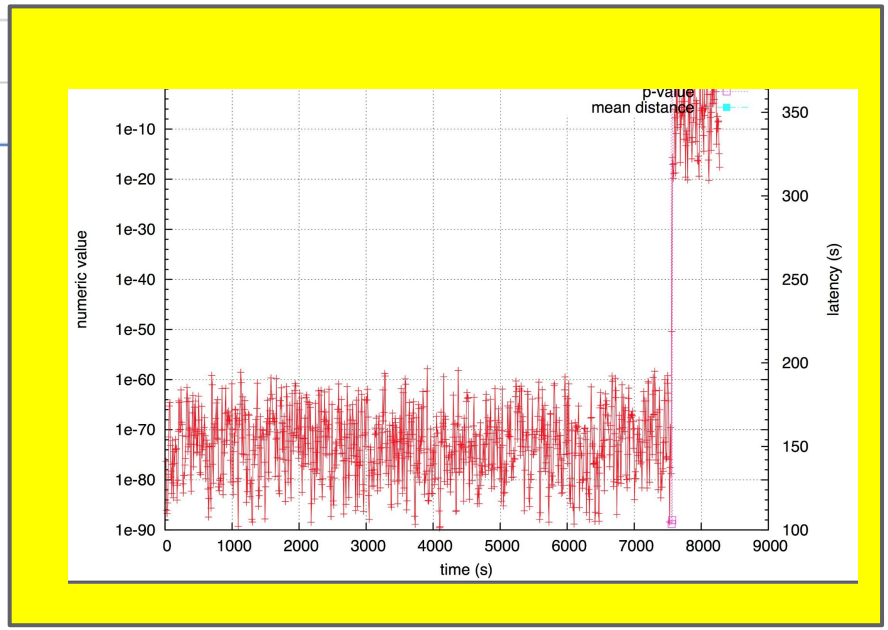
Total FileSystem Time

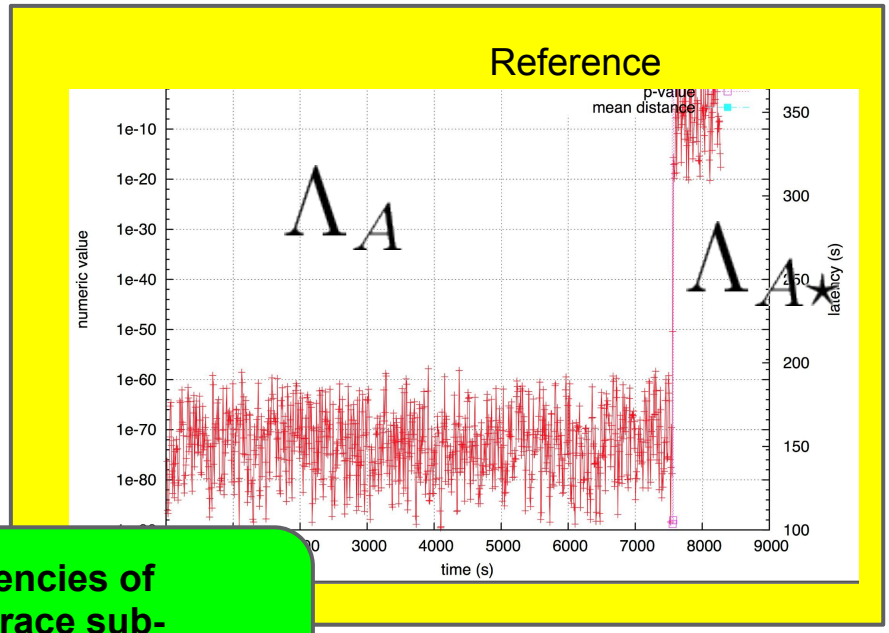


Second Scenario of Interest

Latency distribution has multiple peaks at different orders of magnitude.







Selectively alter latencies of specific traces (or trace sub-components) and then simulate the effects.

$$\Lambda_{\delta A} = \mathbb{P}_{t \in A}(\Psi(\chi, \lambda_{\delta C(t)}))$$

Alter overall distribution of traces and then simulate the effects.

$$\Lambda_{\delta A} = \mathbb{P}^{\delta}_{t \in A}(\lambda_t)$$

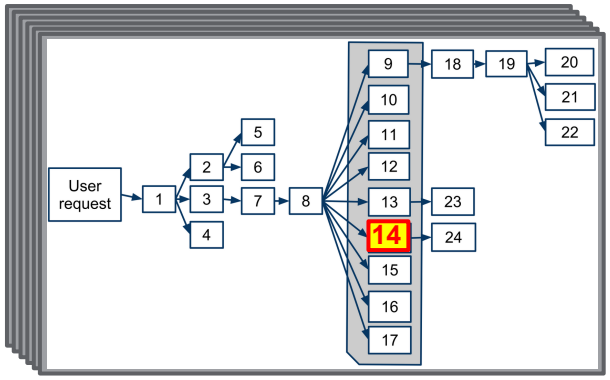
Google

Sample Based Approach

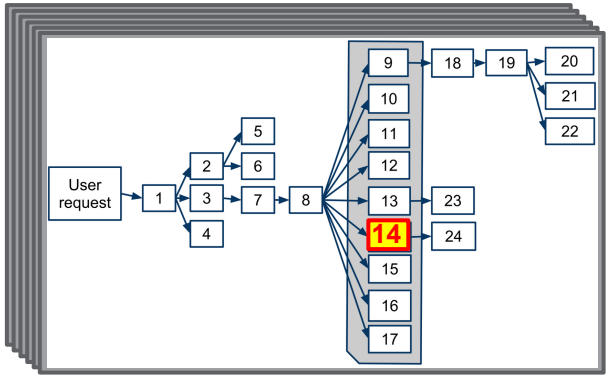
=> shift latency distribution

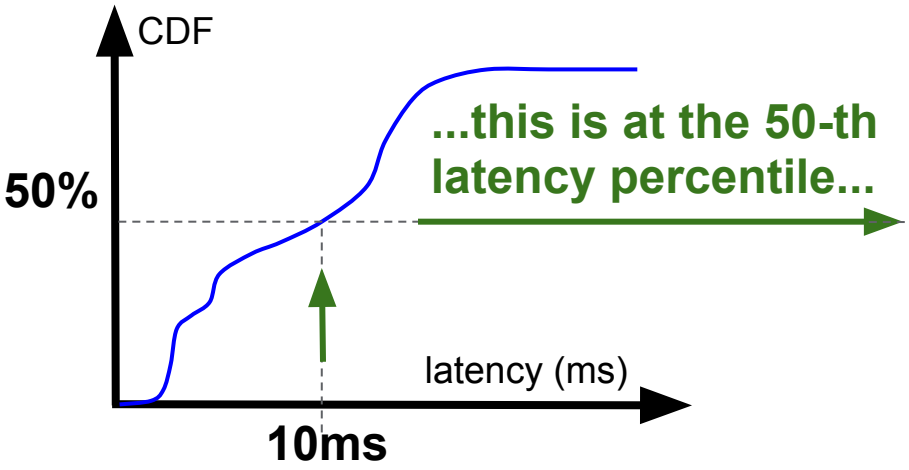
$$\Lambda_{\delta A} = \mathbb{P}_{t \in A}(\Psi(\chi, \lambda_{\delta C(t)}))$$

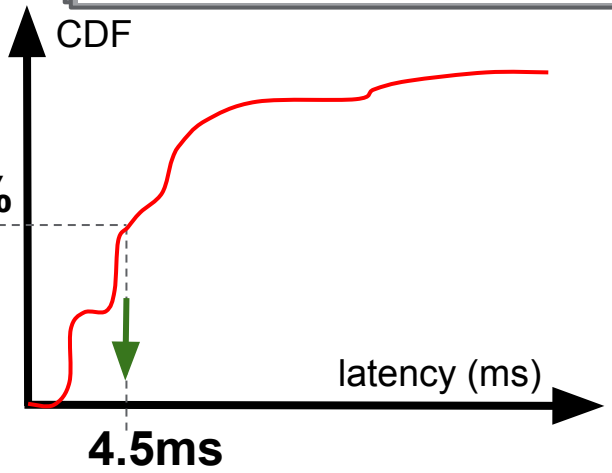
"base" traces



"target" traces







latency of RPC 14 in base trace T = 10ms...

...which used to be 4.5ms

Google

Sample Based Approach

$$\Lambda_{\delta A} = \mathbb{P}^{\delta}_{t \in A}(\lambda_t)$$

=> shift distribution of traces

= absent
= present

"base" traces

trace 1	trace 2	trace 3

"fix"

modified traces

trace 1	trace 2	trace 3
weight = 1/6	weight = 1/6	weight = 2/3

"target" traces

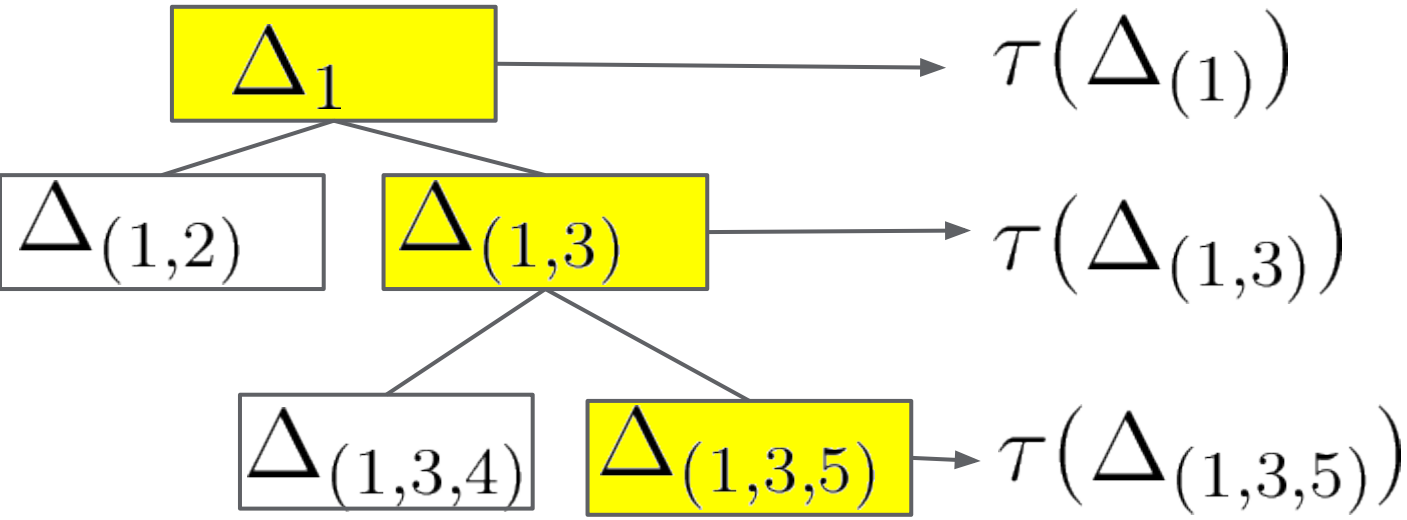
trace 4	trace 5	trace 6

Root Cause	25%	50%	75%	90%	95%	98%
▶ Total Server Detail			-8.67	-1048.20	-1618.00	-583.34
▶ :/Storage Read Detail	0.00	0.02	-9.37	-1015.80	-1516.00	-376.00
▶ Total Local Computation Detail	0.00	0.02	-9.36	-1015.80	-1516.00	-376.00
▶ :--method returned-- Detail	0.00	0.02	-9.36	-1015.80	-1516.00	-376.00
▶ Distribution of Annotation Existence: netsched send (624) Detail @Annotation Existence: netsched send (624): PRESENT Detail Distribution of Annotation Existence: netsched queuing send behind %d others (624) Detail @Annotation Existence: netsched queuing send behind %d others (624): PRESENT Detail	0.00	0.01	-1.89	-950.00	-1501.00	-382.29
▶ @Annotation Existence: starting lock acquisition (610): PRESENT Detail	-0.33	-0.47	-9.85	-1034.20	-1850.00	-6

Sample Expectation at of 90%.

Λ_A Sample $\Lambda_{A\star}$ Reference

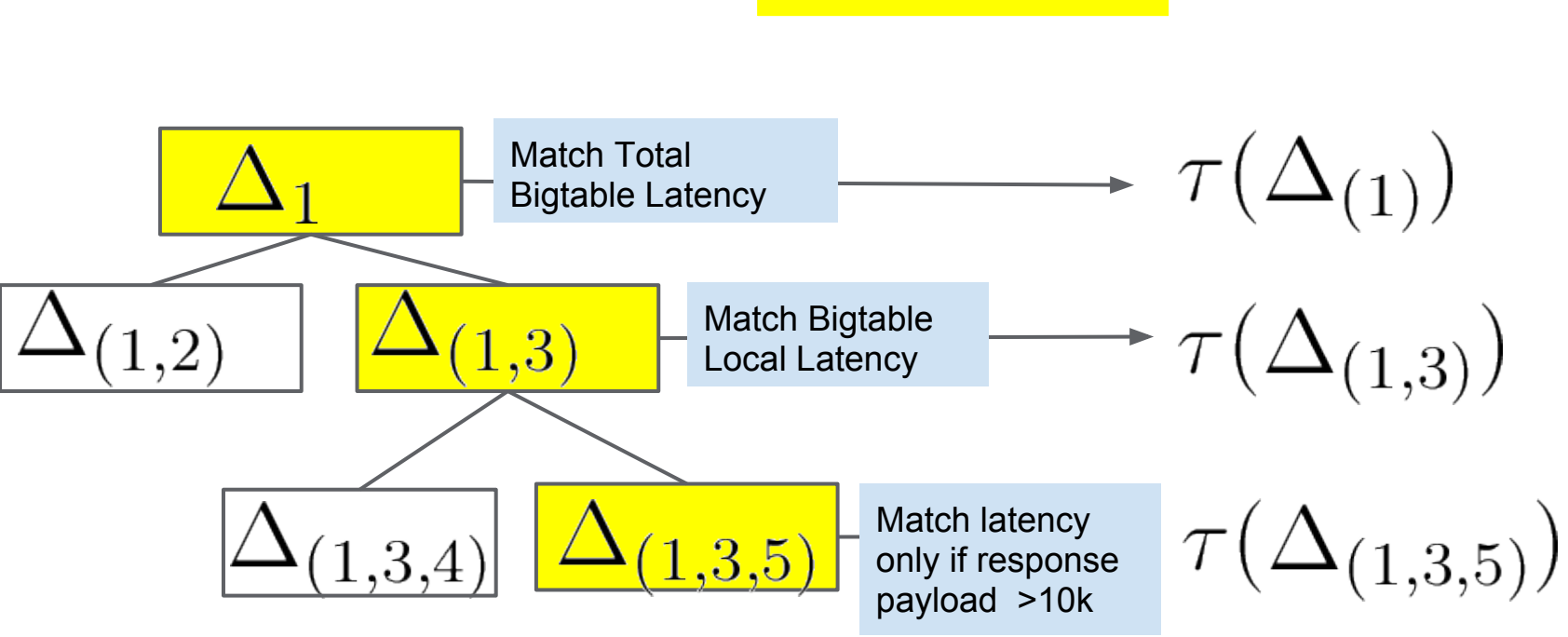
$$\tau(\Delta_j) = \mathbb{E}[\Lambda_{\Delta j A}] - \mathbb{E}[\Lambda_{A\star}]$$



Λ_A Sample $\Lambda_{A\star}$ Reference

$$\tau(\Delta_j) = \mathbb{E}[\Lambda_{\Delta j A}] - \mathbb{E}[\Lambda_{A\star}]$$

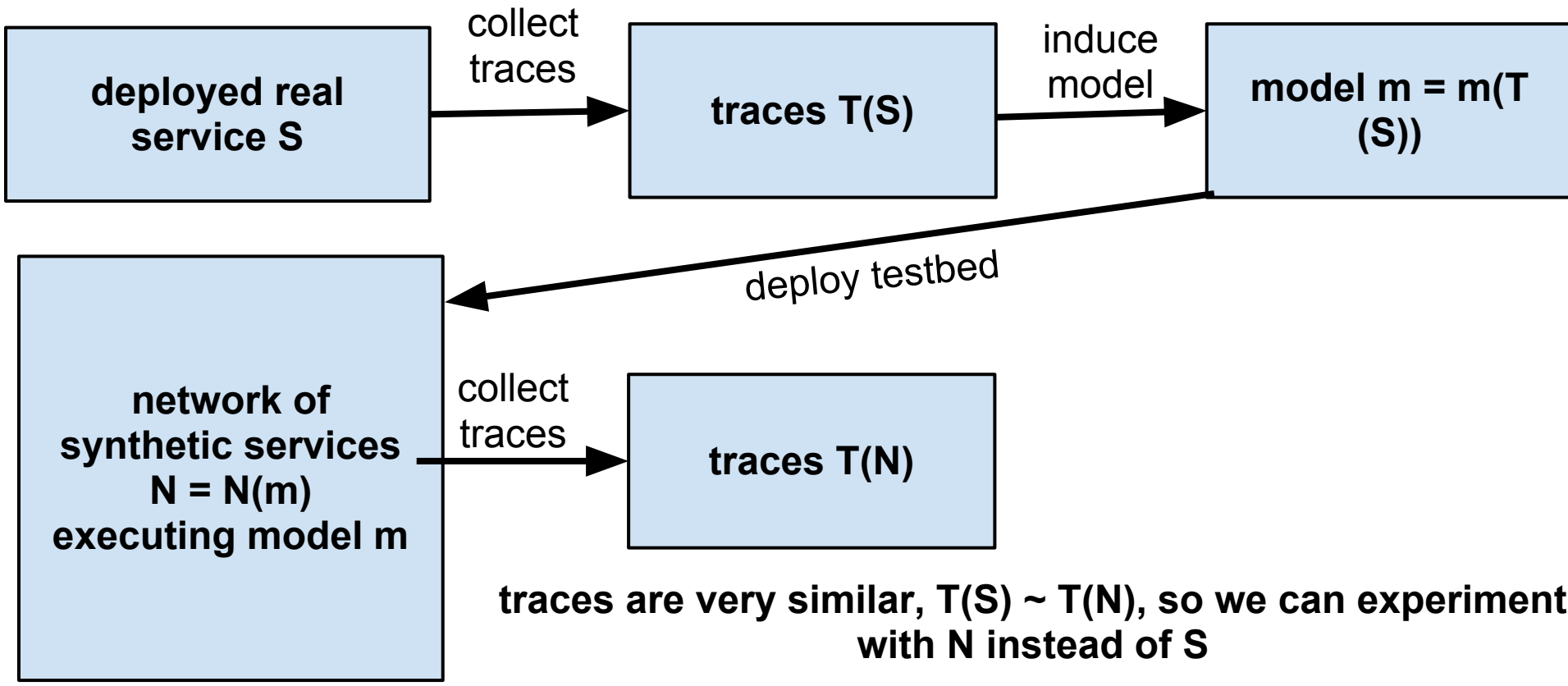
Mean Latency **Difference** at 90% percentile



Evaluation Testbed

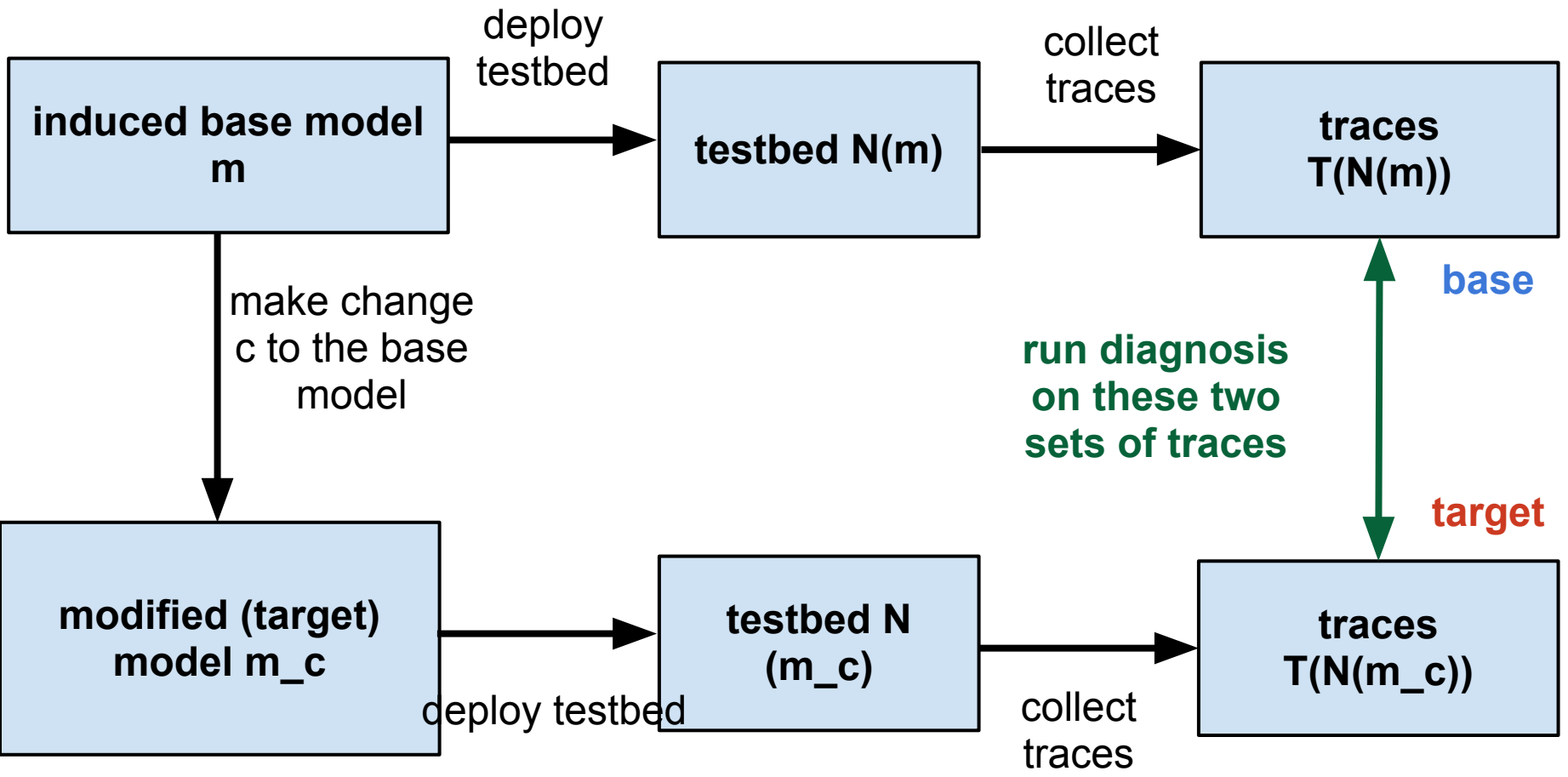
Synthetic Services: physical network delays, random lognormal processing times, temporal dependencies from the induced execution models

Diagnose changes we introduce: Modify the model of a service, generate a new set of traces using the modified model, and diagnose the difference between these sets of traces using our tool



traces are very similar, T(S) ~ T(N), so we can experiment with N instead of S

Ostrowski, Mann, Sandler. LADIS '11.

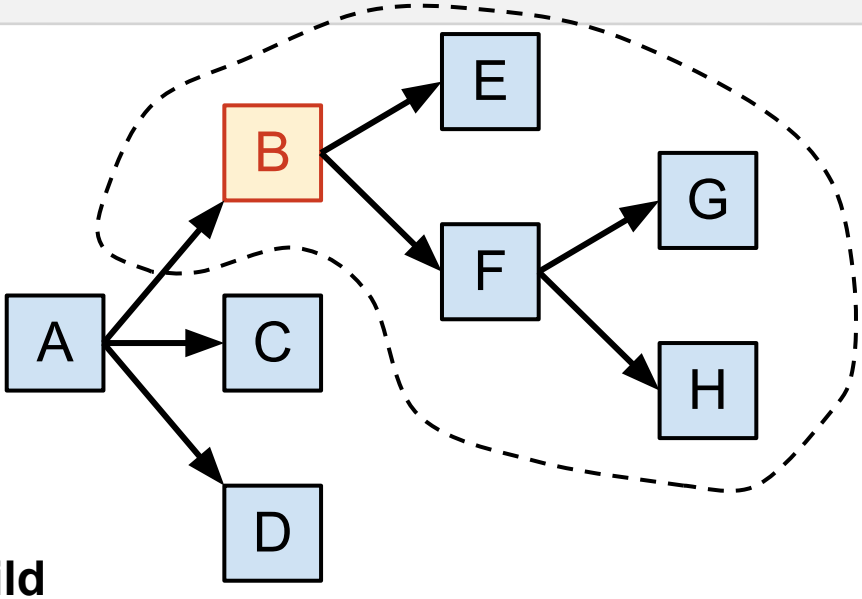


Ostrowski, Mann, Sandler. LADIS '11.

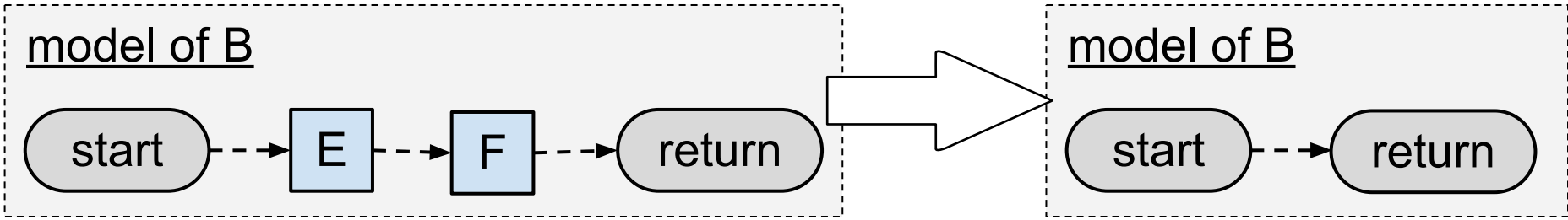
Google

Two Introduced Changes

1. Substitute ("S"): replace the models in a subtree of base services with those from the target. Only on one call path, so if B is called in two different places, only one of them is affected



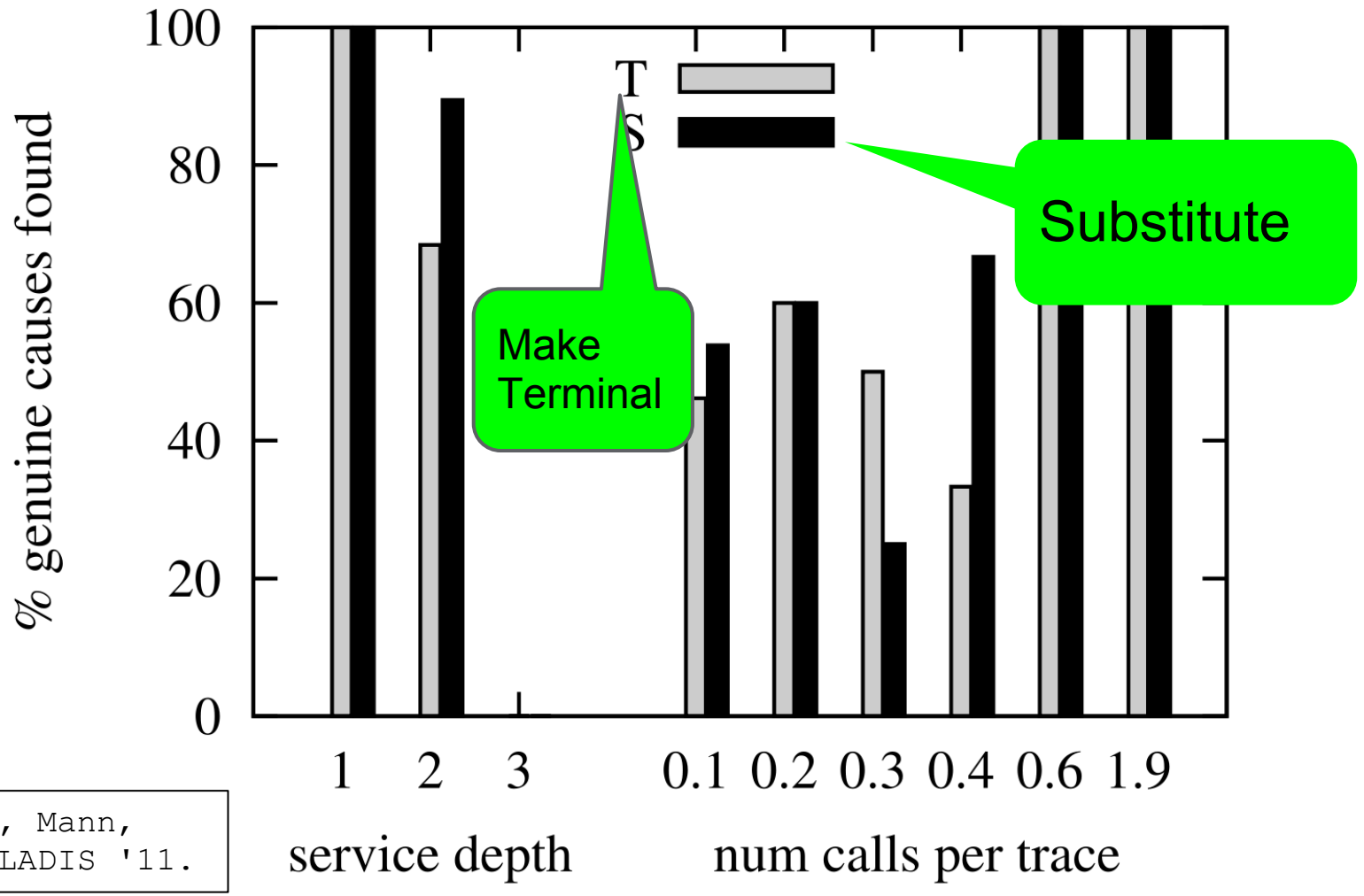
2. Make Terminal ("T"): remove all child RPCs in a model of a service; make it return immediately



Ostrowski, Mann, Sandler. LADIS '11.

Google

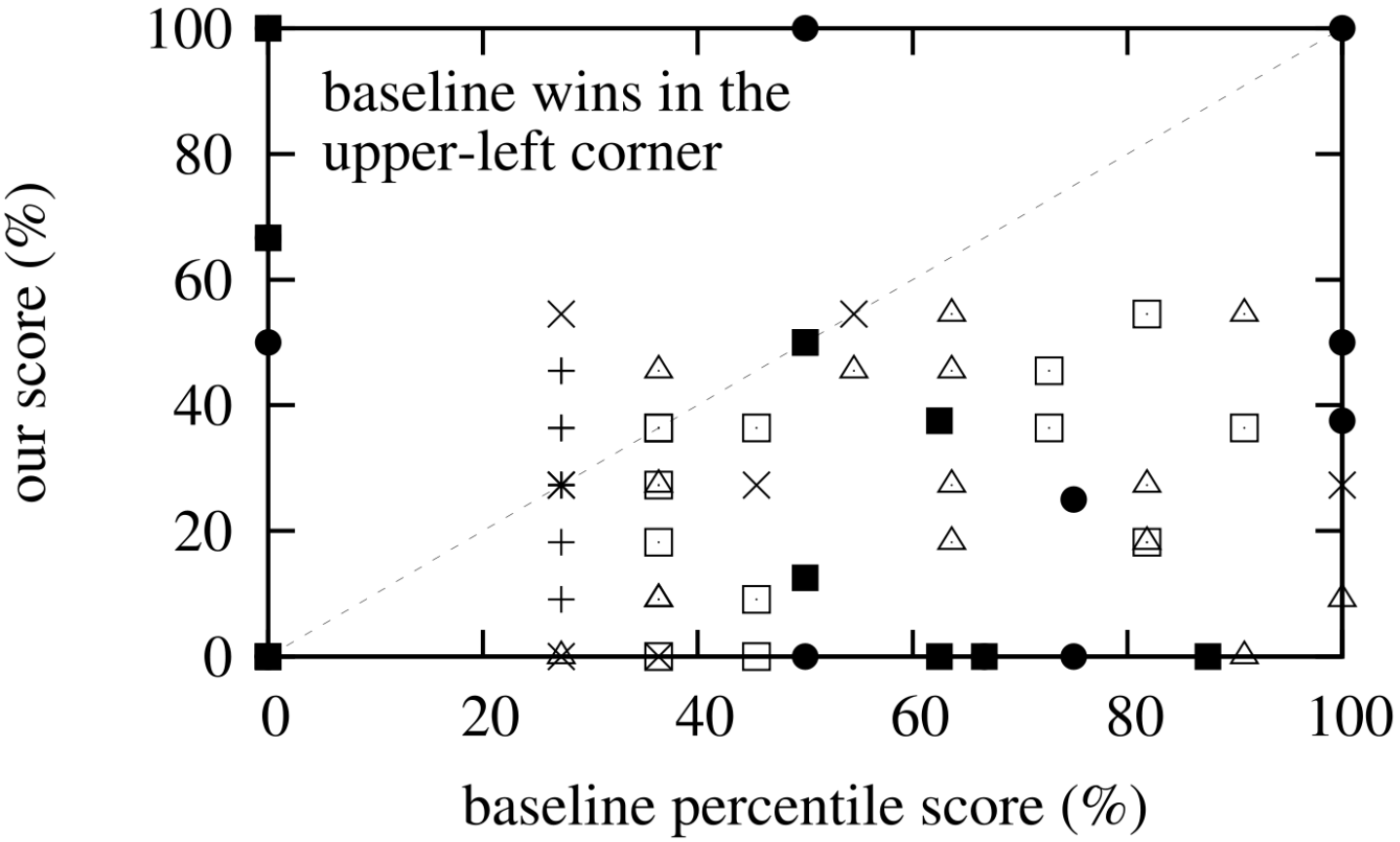
The number of genuine causes found and missed.



Ostrowski, Mann, Sandler. LADIS '11.

Google

The quality of ranking relative to the K-S baseline.

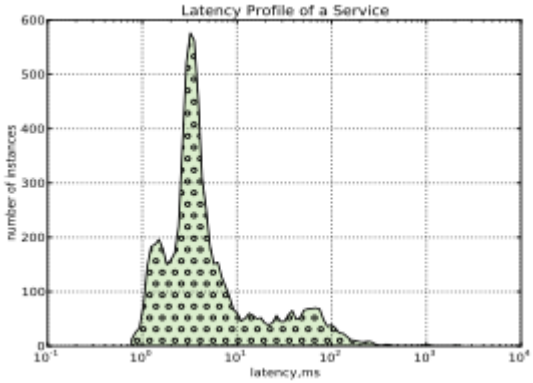


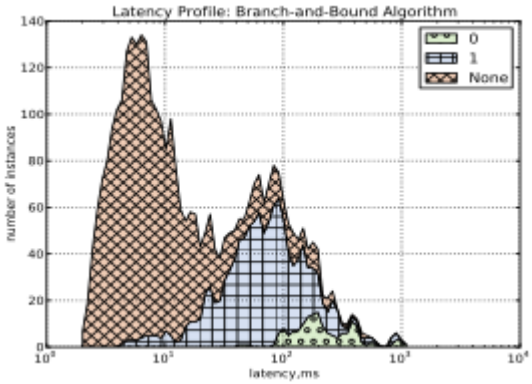
Ostrowski, Mann, Sandler. LADIS '11.

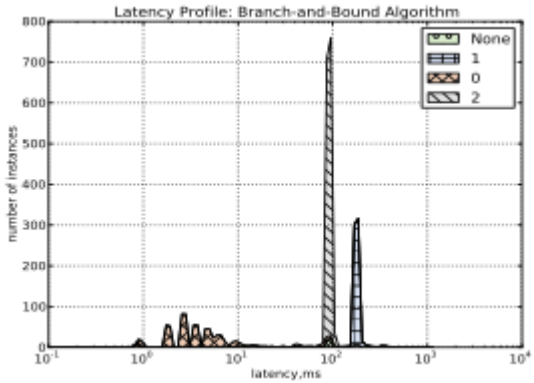
Google

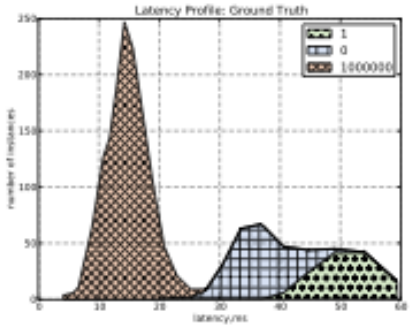
Recent Work

Understanding Latency Variation of Black Box Services









Krushevskaja, Sandler. WWW '13

Diagnosis and Automation for Deployed Distributed Systems

Google™

- Automatic Job Health Assessment
- Real-time Diagnosis of Problem Events
- Diagnosis of Text Bugs
- Intelligent Automation