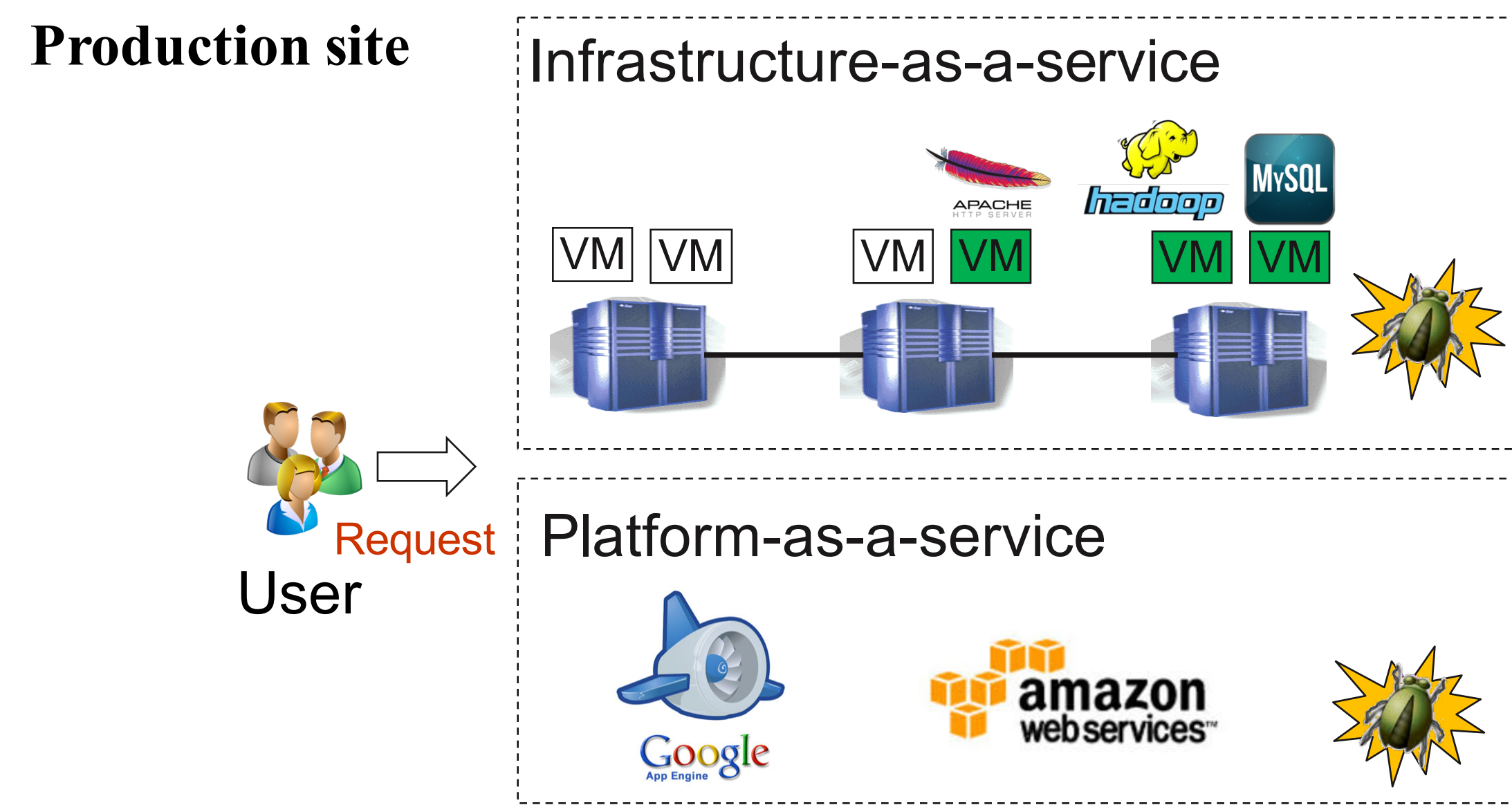


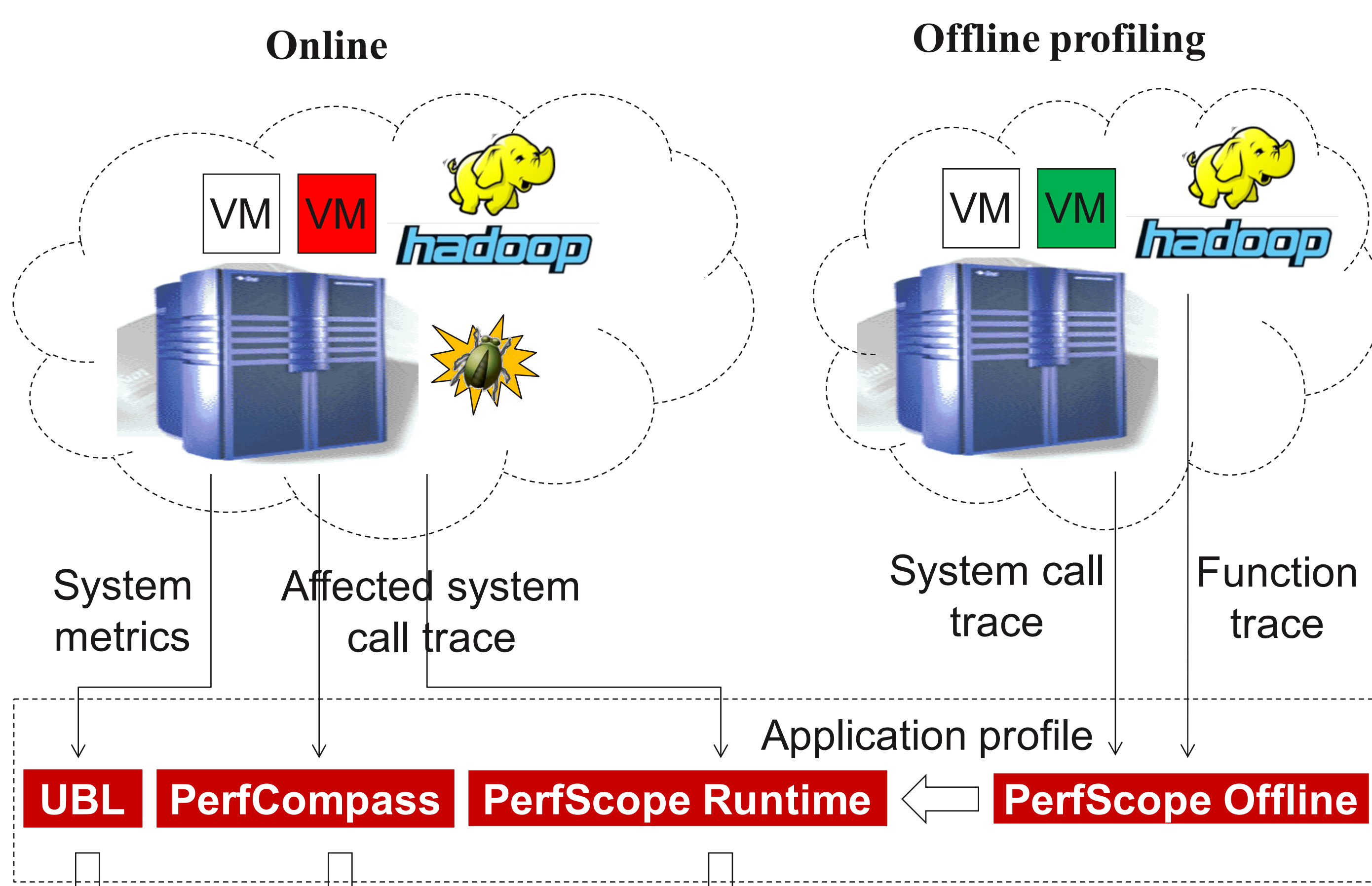
Motivation and Problems



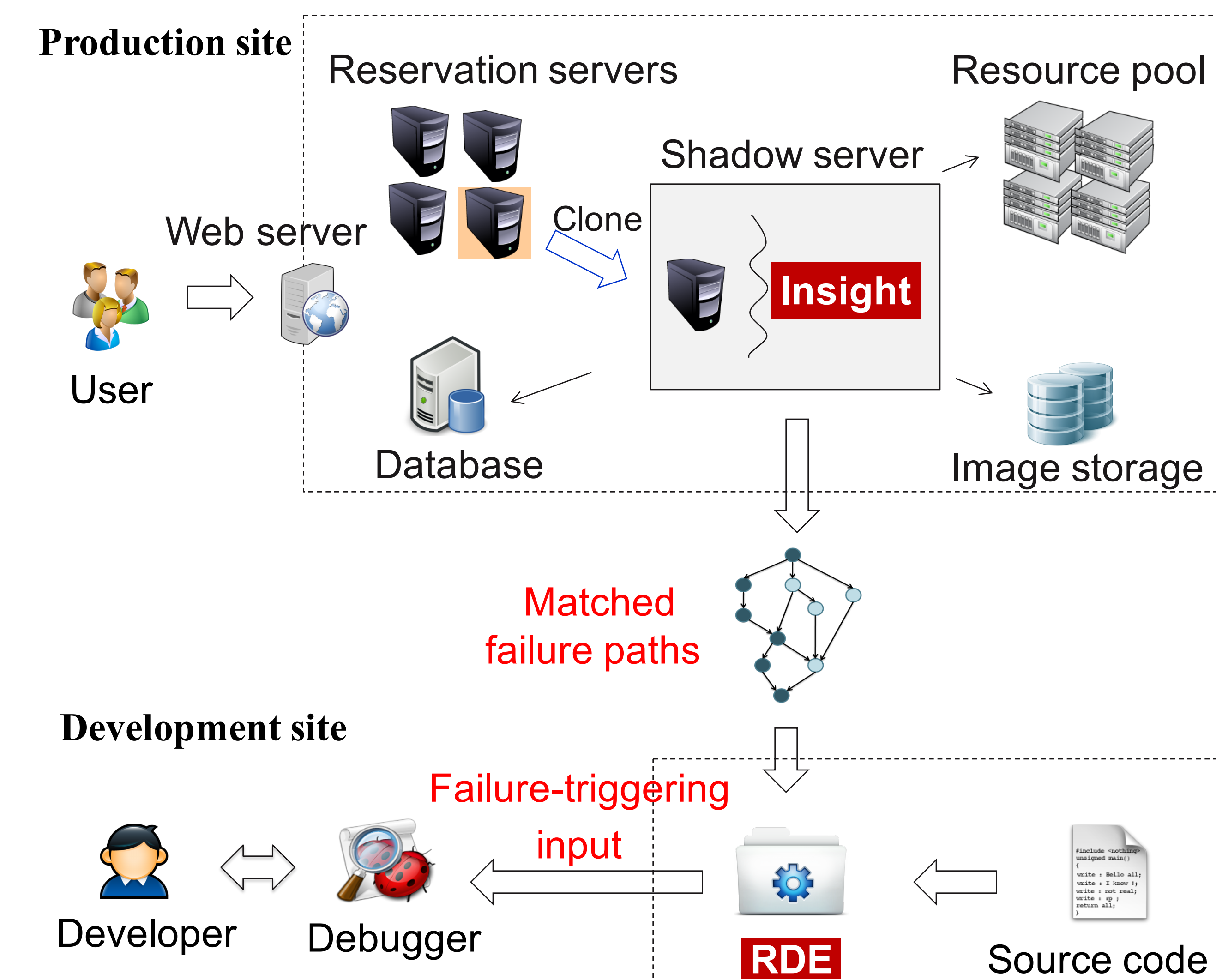
- Online services are prone to failures.
- Non-crashing failures often go unnoticed
- Error message does not tell why failures occur
- Reproducing production site failures is difficult
- Record and replay imposes high overhead and has privacy concerns
- Performance bugs are notoriously difficult to diagnose
- No runtime feedback (e.g., stack trace, error msg)
- No environment data
- Offline bug reproduction & debugging is **HARD**

Approach

- **UBL**: Unsupervised Behavior Learning for Predicting Performance Anomalies in Virtualized Cloud Systems
- **PerfCompass**: Toward Runtime Performance Anomaly Fault Localization for Infrastructure-as-a-Service Clouds
- **PerfScope**: Practical Online Server Performance Bug Inference in Production Cloud Computing Infrastructures

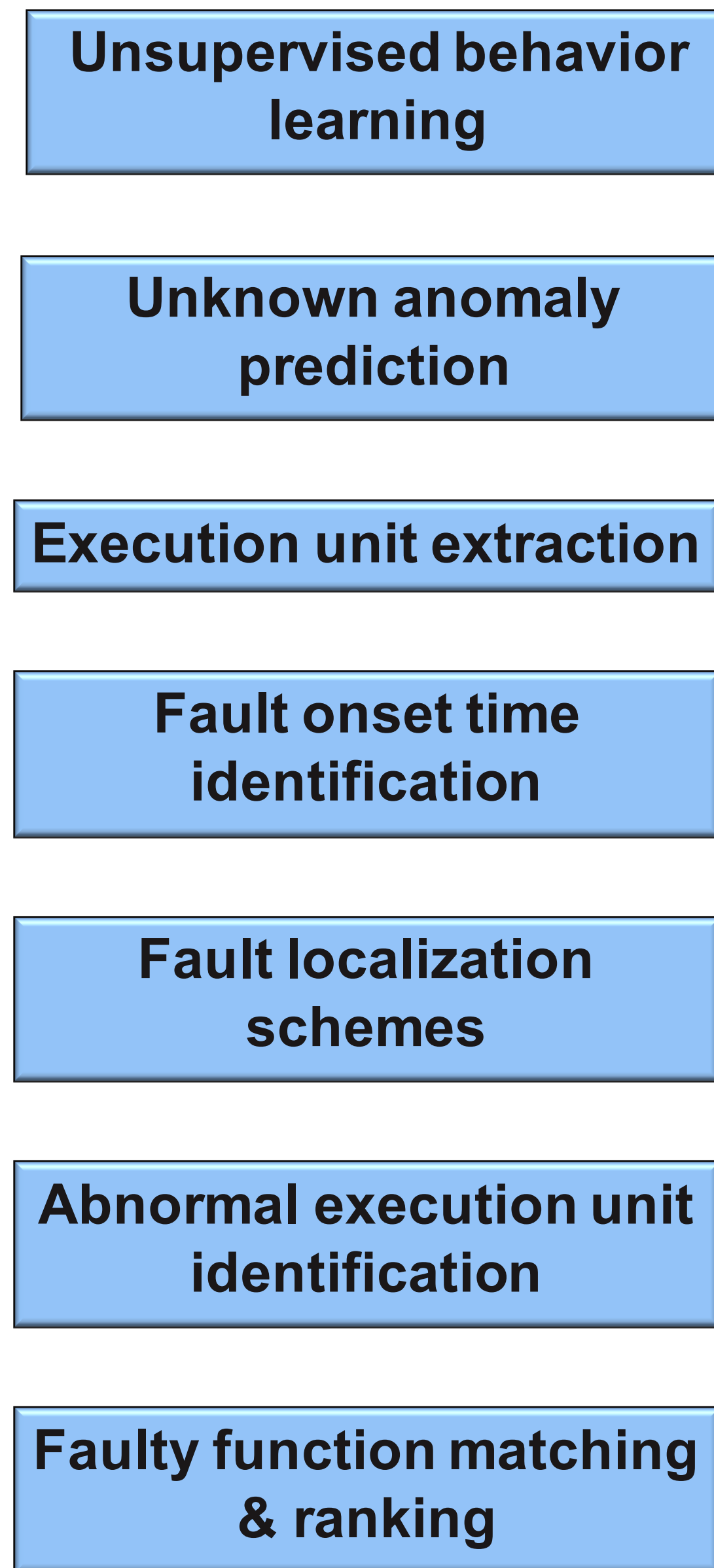


- **Insight**: In-situ Online Service Failure Path Inference in Production Computing Infrastructures
- **RDE**: Replay DEbugging for Diagnosing Production Site Failures



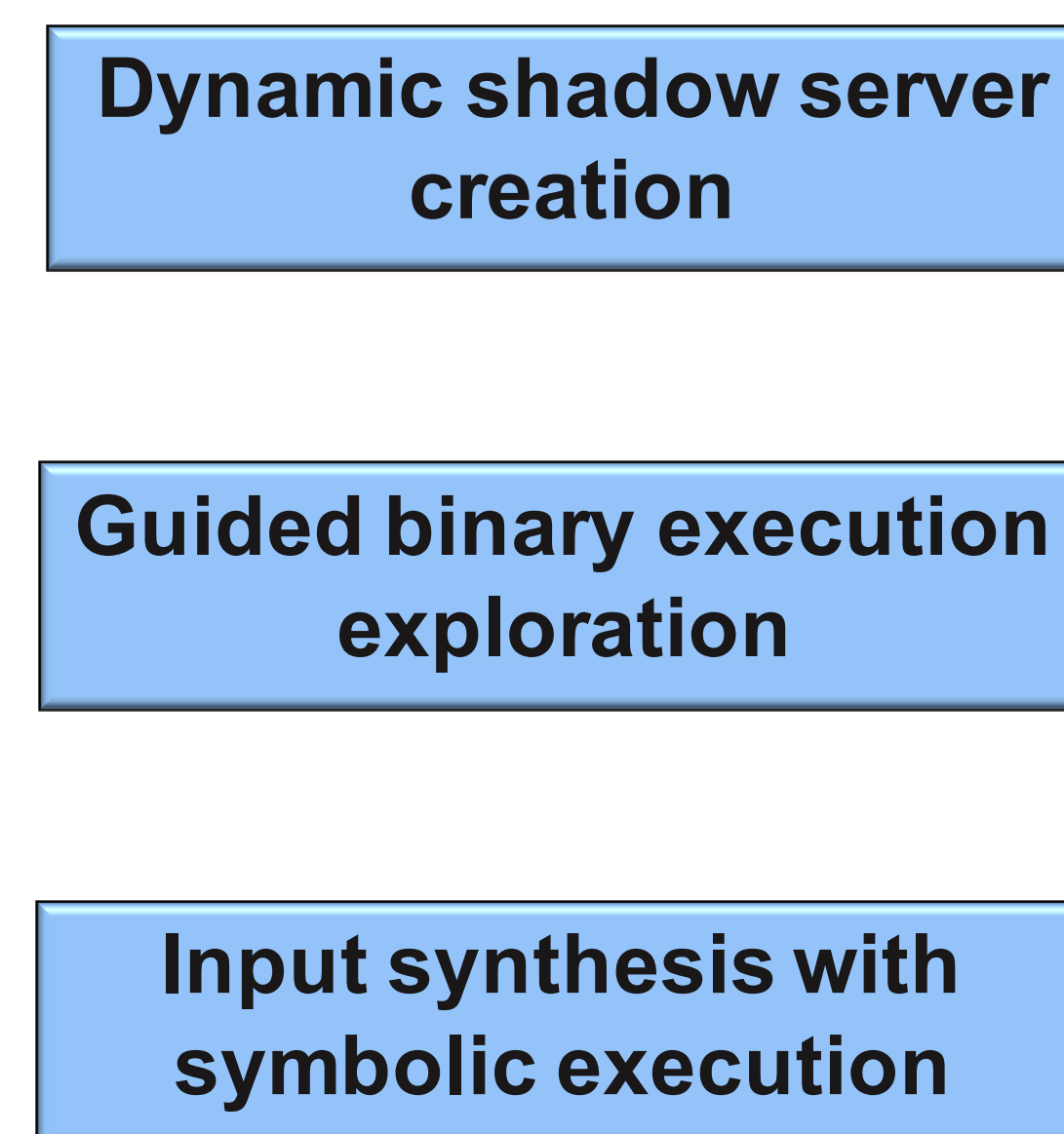
Components

Runtime UBL & PerfCompass & PerfScope



- UBL learns the normal behavior of the system using unsupervised learning
- Anomalies have pre-failure symptoms. Neighborhood area size is used to raise an alarm when pre-failure symptoms are seen.
- Segment large raw traces into smaller units
- Process all execution units to extract fault features
- Analyze extracted features to differentiate internal vs. external faults
- Identify execution units affected by the bug
- Identify and rank possible bug related functions

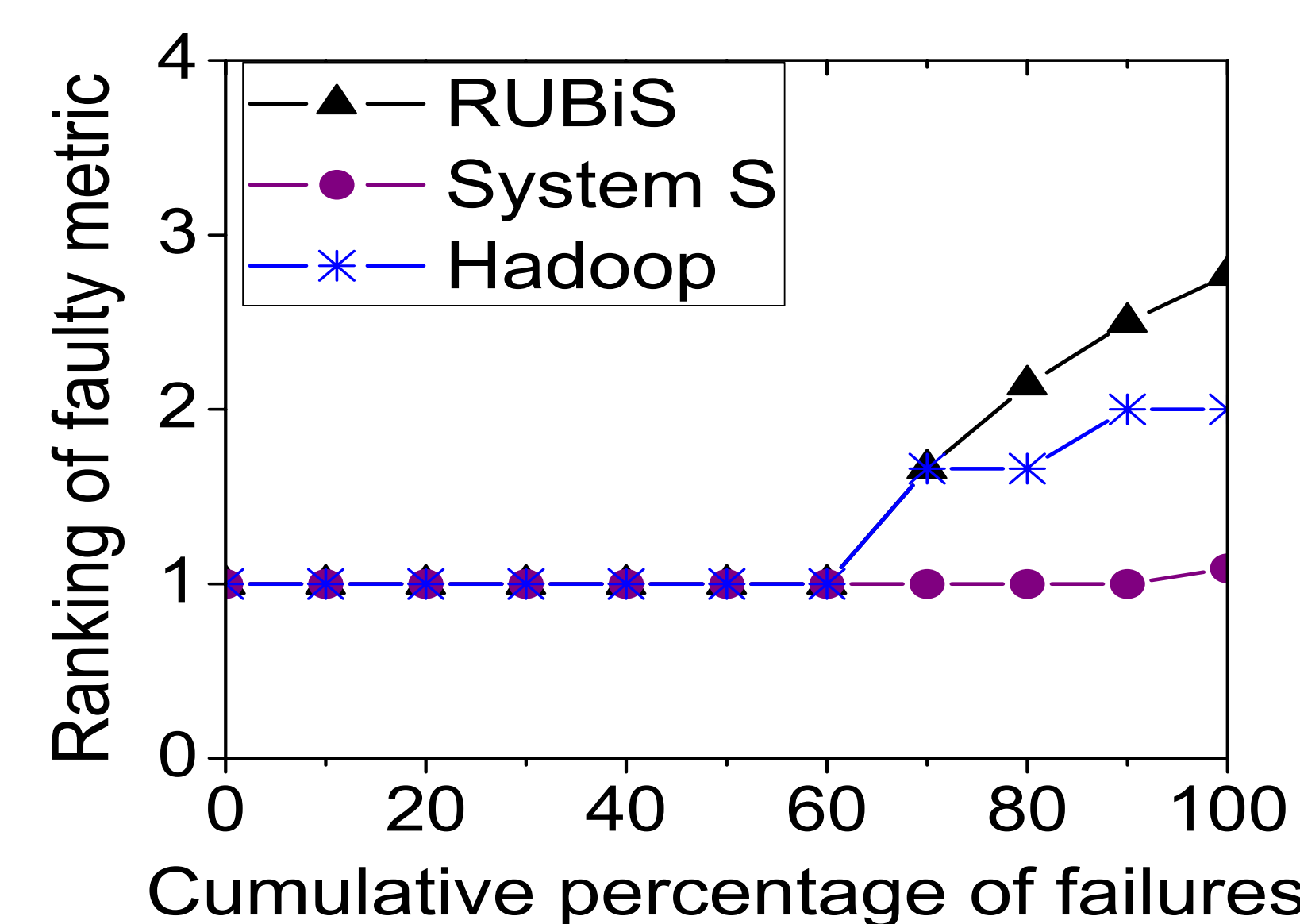
Just-in-time Insight & RDE



- Use live VM cloning to decouple analysis from the production run
- Leverage the production environment data and runtime outputs as guidance to search the failure paths
- Performs symbolic execution along the inferred path to compute and solve path constraints and synthesize the production site data

Results Summary

UBL



- UBL is able to consistently rank the anomaly cause in the **top three** metrics.

PerfCompass

System name	Fault description	Fault impact factor	Fault onset time dispersion
Apache	Flag setting bug	50 +/- 0.5 %	374 +/- 63ms
MySQL	Deadlock bug	40 +/- 0 %	38 +/- 3ms
MySQL	Data flushing bug	62 +/- 3 %	721 +/- 4ms
Squid	File access bug	83 +/- 1 %	0.35 +/- 0.09ms
Cassandra	Endless loop	51 +/- 5.7 %	25 +/- 0.98ms
Hadoop	Endless read	81 +/- 0 %	23 +/- 6ms
Hadoop	Thread shutdown	85 +/- 0.5 %	110 +/- 20ms

PerfScope

Bug Name	Percent of identified functions	Example bug related function
Mapreduce-3738	0.52%	getState(4)
HDFS-3318	1.5%	Reader.performIO(3)
Cassandra-5064	0.6%	maybeSwitchMemtable(4)
Tomcat-53450	1.5%	addLifecycleListener(2)
Tomcat-53173	0.8%	LimitLatch.countDown(11)
Tomcat-42753	0.4%	Poller.run(3)
Apache-37680	0.7%	apr_allocator_mutex_get(14)
Apache-43238	2%	ssl_hook_pre_connection(9)
Lighttpd-1212	0.1%	fdevent_poll(1)
Lighttpd-1999	0.08%	connection_handle_read_state(3)
MySQL-54332	0.6%	Ha_lock_engine(14)
MySQL-65615	0.03%	buf_flush_list(4)

Insight & RDE

Failure name	Original input	Same input type + console log	Same input type + console log + system call
Apache (authentication failure)	0	17	11
Apache (CGI failure)	0	140	9
Squid (non-crashing stop failure)	0	0	0
Lighttpd (proxy failure)	0	0	0
PBZIP2 (decompression failure)	0	1	0
aget (download failure)	0	0	0
rmdir (option failure)	0	0	0
ln (option failure)	0	0	0
touch (time failure)	0	0	0

- Reproduced failure paths **always cover** root cause functions and branches

Conclusion

UBL & PerfCompass & PerfScope:

- Accurately capture system behavior using unsupervised black-box metrics
- Accurately predict unknown anomalies
- Effectively extracts fault features from large system call traces
- Successfully performs fault localization using extracted features
- Identifies bug-related functions for interpreted and compiled programs
- Lightweight with little impact to host systems
- Does not require source code or application instrumentation
- Does not require expensive online instrumentation

Insight & RDE:

- Enable failure path inference inside the production environment
- Use shadow component to decouple failure analysis from production run
- Guided binary execution exploration to find high fidelity failure paths quickly without source code
- Reproduce production-site failure execution at the development site using inferred failure path
- Provide guided symbolic execution exploration to synthesize failure-triggering user inputs

References

- P. Wang, H. Nguyen, X. Gu, S. Lu. RDE: Replay DEbugging for Diagnosing Production Site Failures. SRDS, 2016.
- D. Dean, H. Nguyen, P. Wang, X. Gu, A. Sailer, A. Kochut. "PerfCompass: Online Performance Anomaly Fault Localization and Inference in Infrastructure-as-a-Service Clouds". TPDS, 2015.
- D. Dean, P. Wang, X. Gu, W. Enck, G. Jin. "Automatic Server Hang Bug Diagnosis: Feasible Reality or Pipe Dream?". ICAC, 2015.
- P. Wang, D. Dean, X. Gu. "Understanding Real World Data Corruptions in Cloud Systems". IC2E, 2015.
- D. Dean, H. Nguyen, X. Gu, H. Zhang, J. Rhee, N. Arora, G. Jiang. "PerfScope: Practical Online Server Performance Bug Inference in Production Cloud Computing Infrastructures". SOCC, 2014.
- D. Dean, H. Nguyen, P. Wang, X. Gu. "PerfCompass: Toward Runtime Performance Anomaly Fault Localization for Infrastructure-as-a-Service Clouds". HotCloud, 2014.
- H. Nguyen, D. Dean, K. Kc, X. Gu. "Insight: In-situ Online Service Failure Path Inference in Production Computing Infrastructures". USENIX ATC, 2014.
- T. Ho, D. Dean, X. Gu, W. Enck. "PREC: Practical Root Exploit Containment for Android Devices". CODASPY, 2014.
- H. Nguyen, Z. Shen, X. Gu, S. Subbiah, J. Wilkes. "AGILE: elastic distributed resource scaling for Infrastructure-as-a-Service". ICAC, 2013.
- H. Nguyen, Z. Shen, Y. Tan, X. Gu. "FChain: Toward Black-box Online Fault Localization for Cloud Systems". ICDCS, 2013.
- D. Dean, H. Nguyen, X. Gu. "UBL: Unsupervised Behavior Learning for Predicting Performance Anomalies in Virtualized Cloud Systems". ICAC, 2012.
- Y. Tan, H. Nguyen, Z. Shen, X. Gu, C. Venkatramani, D. Rajan. "PREPARE: Predictive Performance Anomaly Prevention for Virtualized Cloud Systems". ICDCS, 2012.