An Infrastructure for Automating Large-scale Performance Studies (Elba under-the-hood)

Calton Pu

Professor and J.P. Imlay Chair in Software

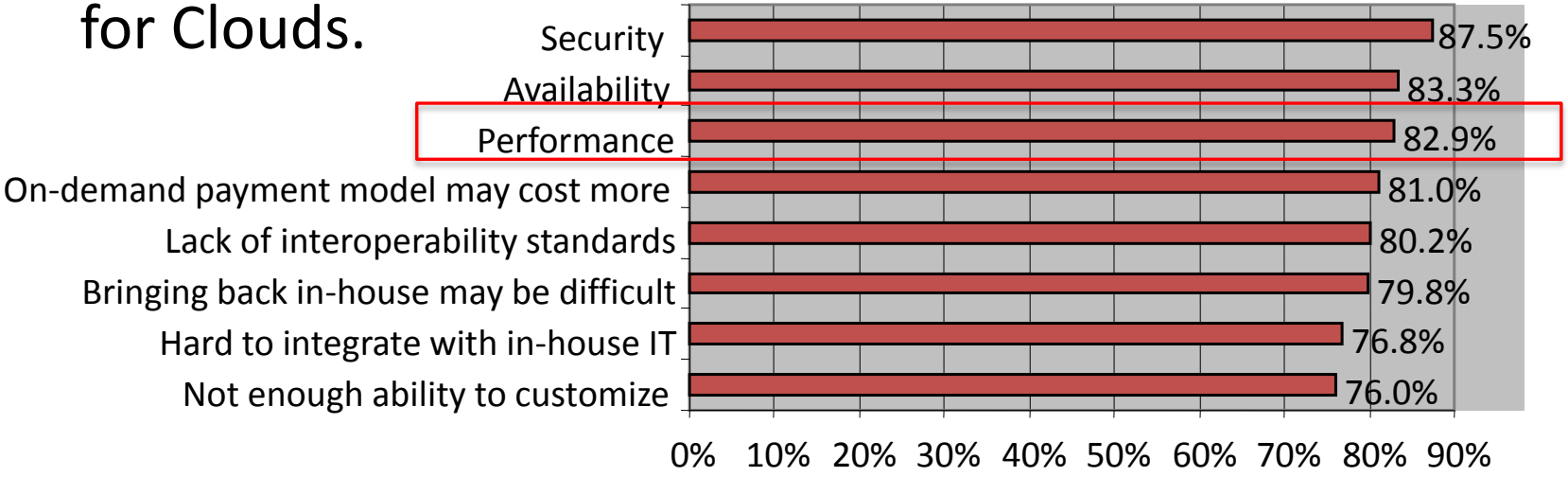
CERCS, Georgia Institute of Technology

Many PhD, MS, Undergrad students

Many company collaborators and support from several companies, particularly from Fujitsu and Intel.

Importance of Predictable Performance

- Extra delay of just 100ms could result in roughly 1% loss in sales. amazon.com
- Additional delay of just 500ms could reduce revenues by 20%. Google
- IDC reported performance to be top 3 user considerations for Clouds.



Cloud Computing & Performance

- Cloud is a ***black-box*** for many users.
- Application providers face non-trivial performance challenges.
- One of the most effective ways to understand a cloud is to measure it:
  - Run measurement studies and collect data
  - Maybe it's the only way

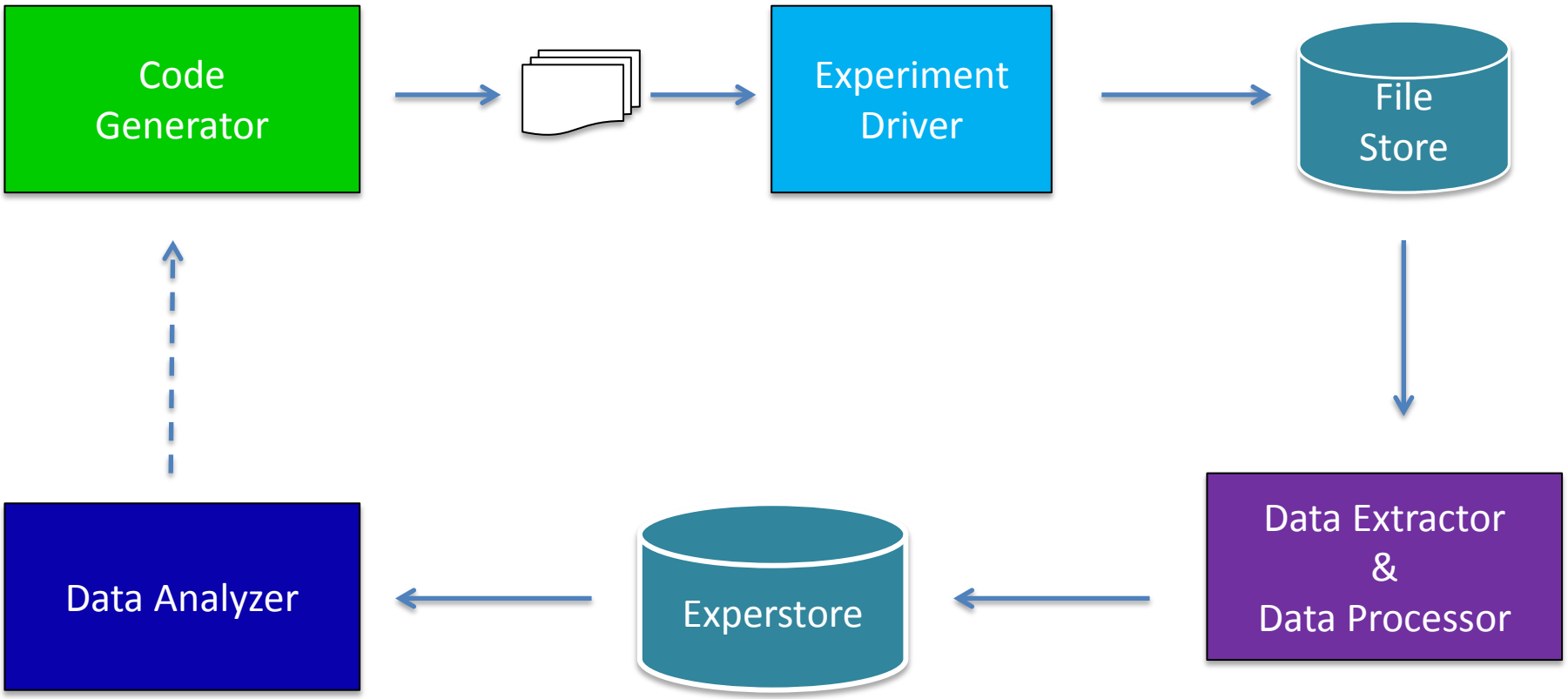
Large-scale Experimental Measurements

- **Goal**: Use large-scale experimental data as “real” (predictive) models for performance and scalability constraints;
- **Challenges:**
  - Deployment complexity due to configuration dependencies
  - Large state space: Many configuration options
  - Huge amount of data: >1GB/experiment, semi-structured data

Automating Large-scale Experiments

- **Create**
  - Prepare the platform, deploy and configure application.
- **Manage**
  - Start application, execute workload, data collection.
- **Analyze**
  - Data analysis (visualization) and building hypothesis.

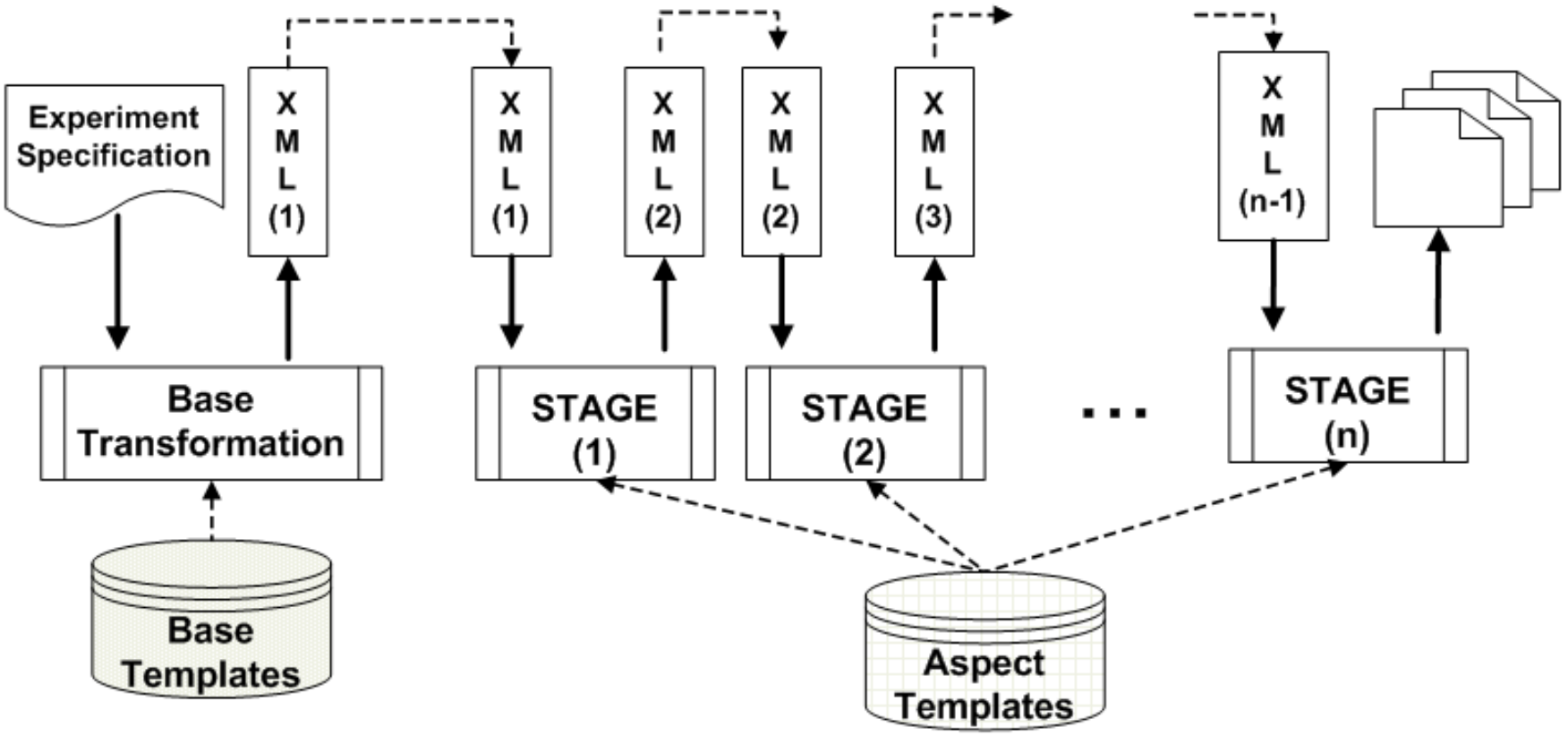
Performance Measurement Workflow



Expertus – Code Generator

- **Idea**: Generate scripts to automatically create, manage and analyze the experiments from user-friendly specification files.
- Key Challenges:
  - **From abstract mapping to concrete scripts**
  - **Heterogeneity of hardware and software components**
  - **Flexible customization needed in experiments**
- Solution platform: XML + XSLT + AOP

Code Generation Pipeline



Template Types

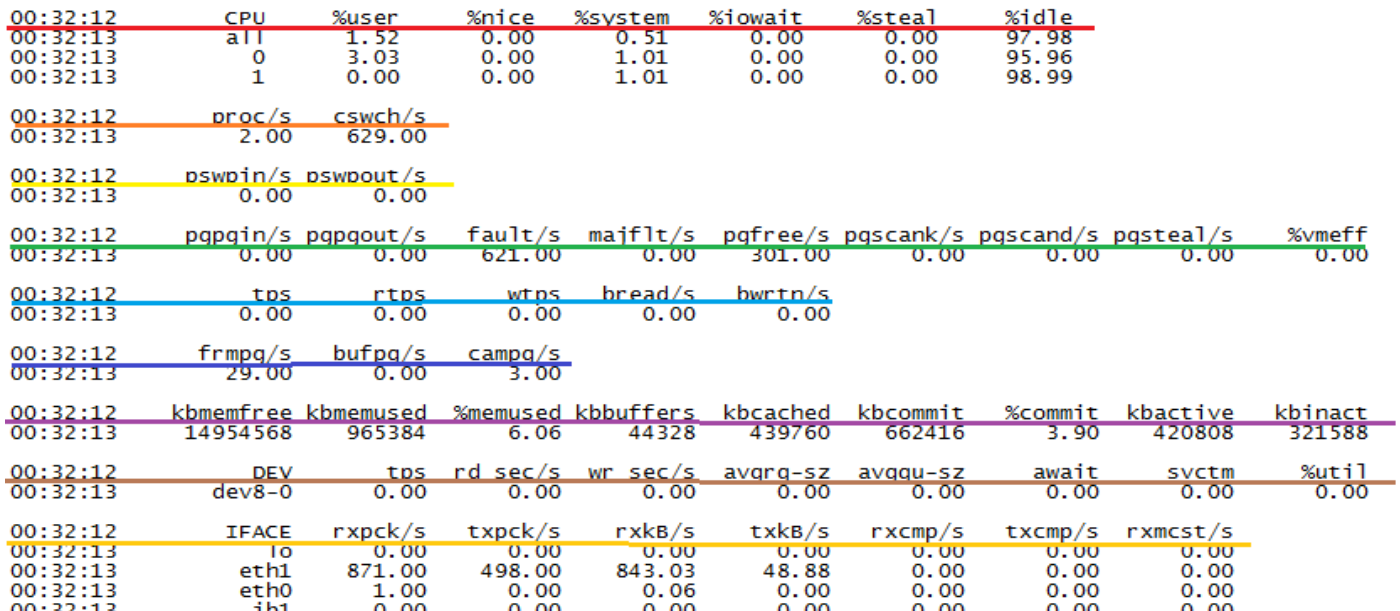
- Base Templates
  - To generate OS, cloud, and user independent resources.
  - A template for each possible action (e.g., deploy-tomcat) and resource (e.g., httpd.conf).
  - Created by identifying output variances (each of which becomes an aspect).
- Aspect Templates
  - Customize to meet application, cloud, and user needs.
  - Can contain one or more advices.
  - Nested pointcuts – an advice can add zero or more pointcuts.

Expertract - Automated Data Extractor

- Performance logs from various monitors, semi-structured
  - Potentially, a custom data parser for each experiment
- Log file format with many variations
  - Monitoring tools (e.g., dstat, sar, o-profiler …) and parameter settings
- ETL (Extract, Transform, Load) tools insufficient by themselves
  - Need to figure out the actual log layout

Most Common File Formats

- One header
- Multiple headers with sequentially corresponding data
- Multiple headers with non-sequential corresponding data
- Multiple headers appear randomly in the file and data is entirely non-sequential

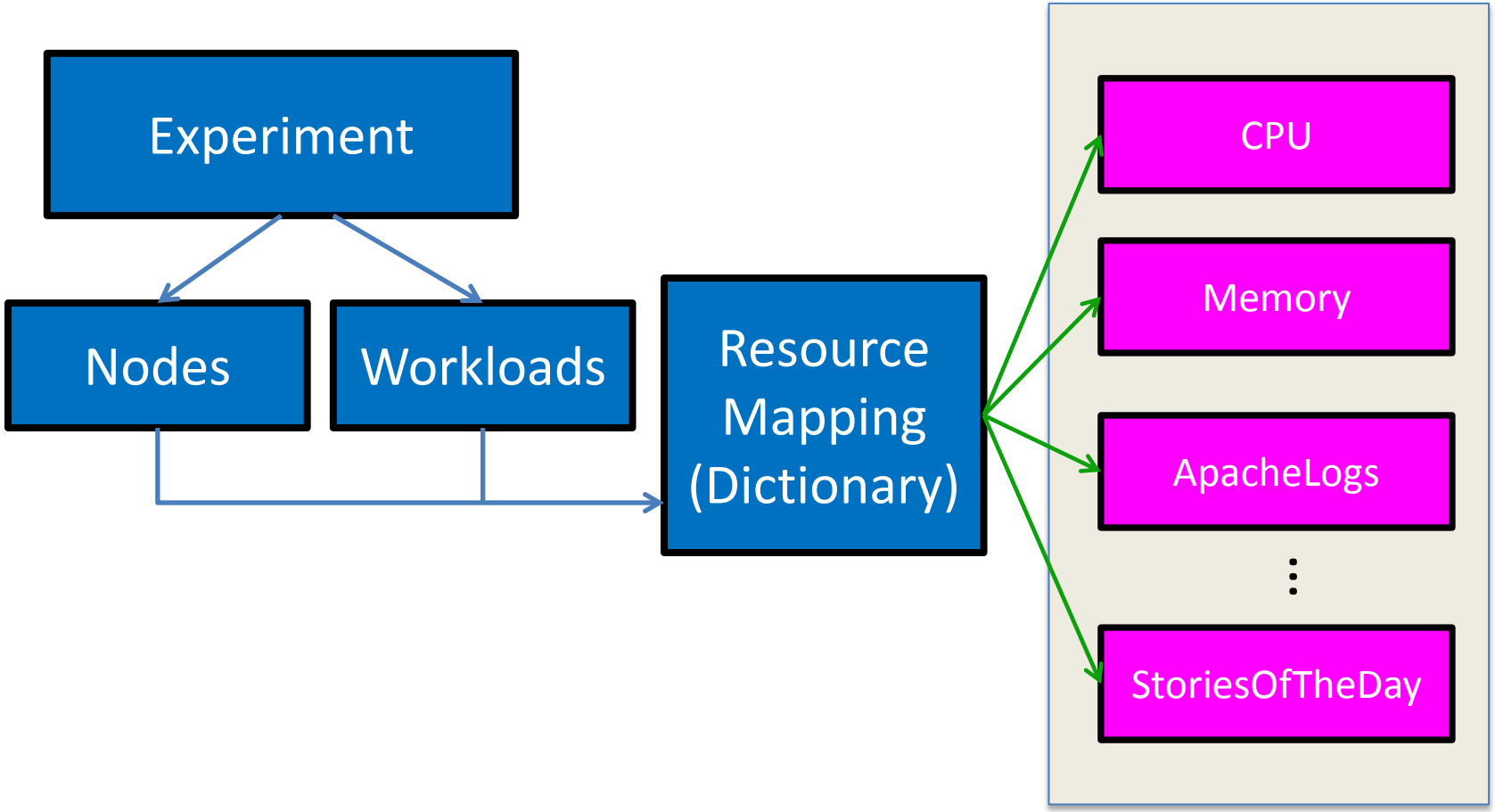


```
00:32:12        CPU     %user     %nice   %system   %iowait    %steal     %idle
00:32:13        all      1.52      0.00      0.51      0.00      0.00     97.98
00:32:13          0      3.03      0.00      1.01      0.00      0.00     95.96
00:32:13          1      0.00      0.00      1.01      0.00      0.00     98.99

00:32:12       proc/s   cswch/s
00:32:13         2.00    629.00

00:32:12     pswpin/s pswpout/s
00:32:13         0.00      0.00

00:32:12     pgpgin/s pgpgout/s   fault/s  majflt/s  pgfree/s pgscank/s pgscand/s pgsteal/s    %vmeff
00:32:13         0.00      0.00    621.00      0.00    301.00      0.00      0.00      0.00      0.00

00:32:12          tps      rtps      wtps   bread/s   bwrtn/s
00:32:13         0.00      0.00      0.00      0.00      0.00

00:32:12      frmpg/s   bufpg/s   campg/s
00:32:13        29.00      0.00      3.00

00:32:12    kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit   %commit  kbactive   kbinact
00:32:13     14954568    965384      6.06     44328    439760    662416      3.90    420808    321588

00:32:12          DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
00:32:13         dev8-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00

00:32:12        IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst/s
00:32:13           lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
00:32:13         eth1    871.00    498.00    843.03     48.88      0.00      0.00      0.00
00:32:13         eth0      1.00      0.00      0.06      0.00      0.00      0.00      0.00
00:32:13          ib1      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

Experstore - A Flexible Data Warehouse

- Tables are created *on-the-fly* based on the data.
- Why not static tables? (global schema too big)
  - Several monitoring programs
  - Many different parameter settings: e.g., 2 core vs. 4 cores
- Why not column based tables?
  - Would be too many tables (over 20000 tables per experiment).
  - (# Workload) * (# Nodes) * (# Resources).
- Our solution – A hybrid approach:
  - Create small tables to store related data, for example a table to store CPU data that consists of user, sys, idle etc…

Static and Dynamic Tables



Data to Schema Mapping

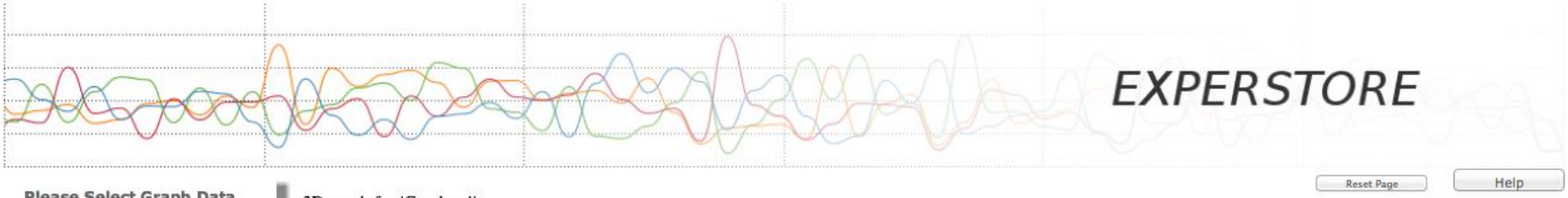
- Mapping performance data to a table (to-be-created) in the data warehouse.
  - Which columns (row) to read ?.
  - How to format (e.g., datetime) ?.
  - What to include/exclude ?.
  - Which parser to use ?.
- Specifying resources for a given node.
  - e.g., CPU1, CPU2, network, IO etc …
- Mapping result directories to workloads.
  - e.g., “2009-11-29@11-25-40”→ 1000-RO
- Mapping log files to a node/resources.
  - e.g., mod_jk.log → request processing time at Apache.

Data Analysis - Web Portal

- Aid the analysis of large amount of data.
- Identify patterns, trends and relations.
- Control the way in which data is represented.
- Data from seemingly unrelated sources could be easily compared against each other.

Web Portal-2D

Web Portal-3D



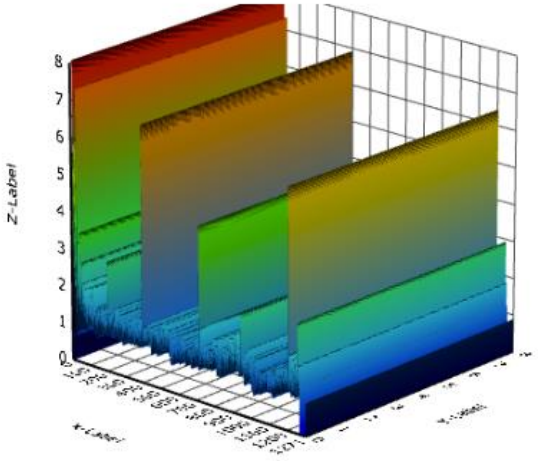
Please Select Graph Data

Select Graph:	Graph - 4
Experiment Name:	rubbos-1-2-:
Iteration:	1
Node Name:	Tomcat2
Resource:	CPU0
Series:	user
Range:	1 to 1271
Step Size:	step
Type of Graph:	Line
Advanced Graph:	3-D Graph
Graph Title:	Title
X-Axis Label:	X-Label
Y-Axis Label:	Y-Label
Z-Axis Label:	Z-Label
Function for plot:	x

Please Select Graph Operation

Operations: --Select--

3D graph for 'Graph - 4'
Z: 1:1
Y: 1:10
X: 1:1

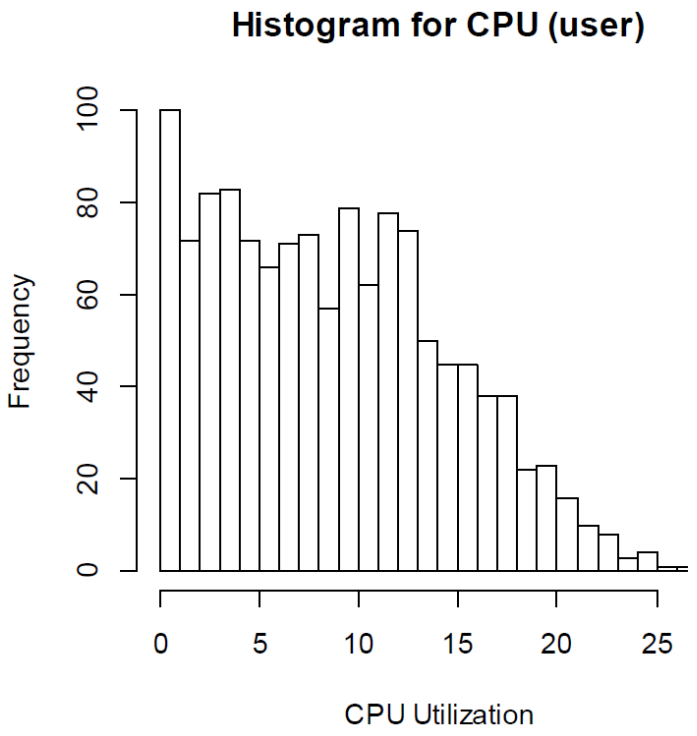


Support for *R* Framework

Generated Script

```
drv <- JDBC("com.mysql.jdbc.Driver",
      "C:/mysql/mysql-connector-java-5.1.7-
       bin.jar",
      identifier.quote="`")

conn <- dbConnect(drv,

       "jdbc:mysql://elbafs.cc.gatech.edu:3313/elba
       ",
      "elba", "elba")

d   = dbGetQuery(conn, "select user from
     TAB1333977005752002l9B_CPU0
     where dictionaryid='334'")

hist(d$user, breaks=20, col="white",
    xlab="CPU Utilization",
    main="Histogram for CPU (user)")
```

Generated Graph



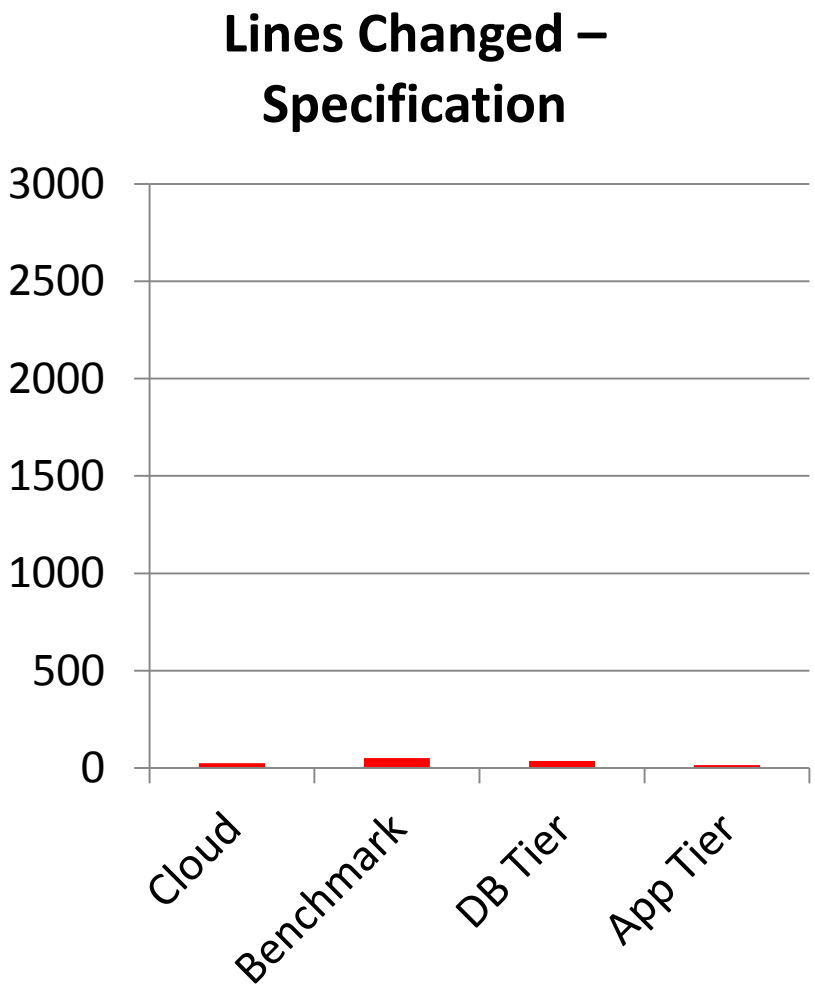
Wide Applicability

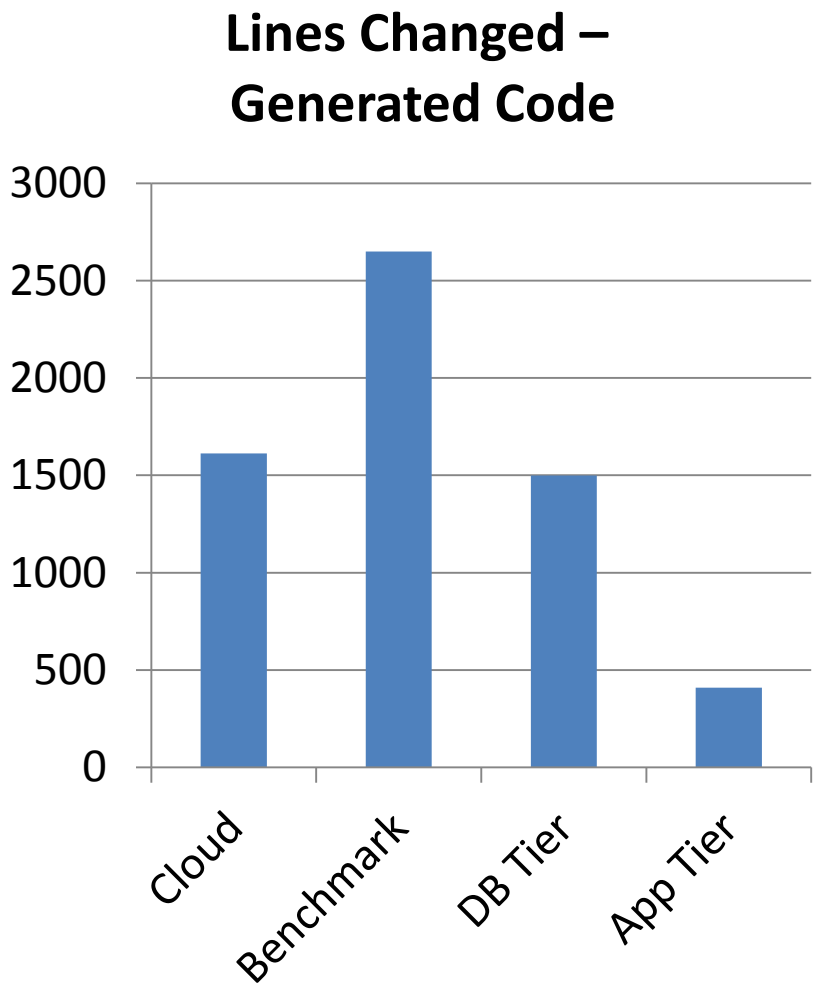
- Over 500 different hardware configurations.
- Over 10,000 software configurations.
- Over 100,000 nodes.
- Many clouds (e.g., Emulab, EC2, OpenCirrus, Georgia Tech cluster, and Wipro).
- Many representative applications (e.g., RUBBoS, RUBiS, CloudStone, and over 10 OLTP benchmarks).

High-Level Summary

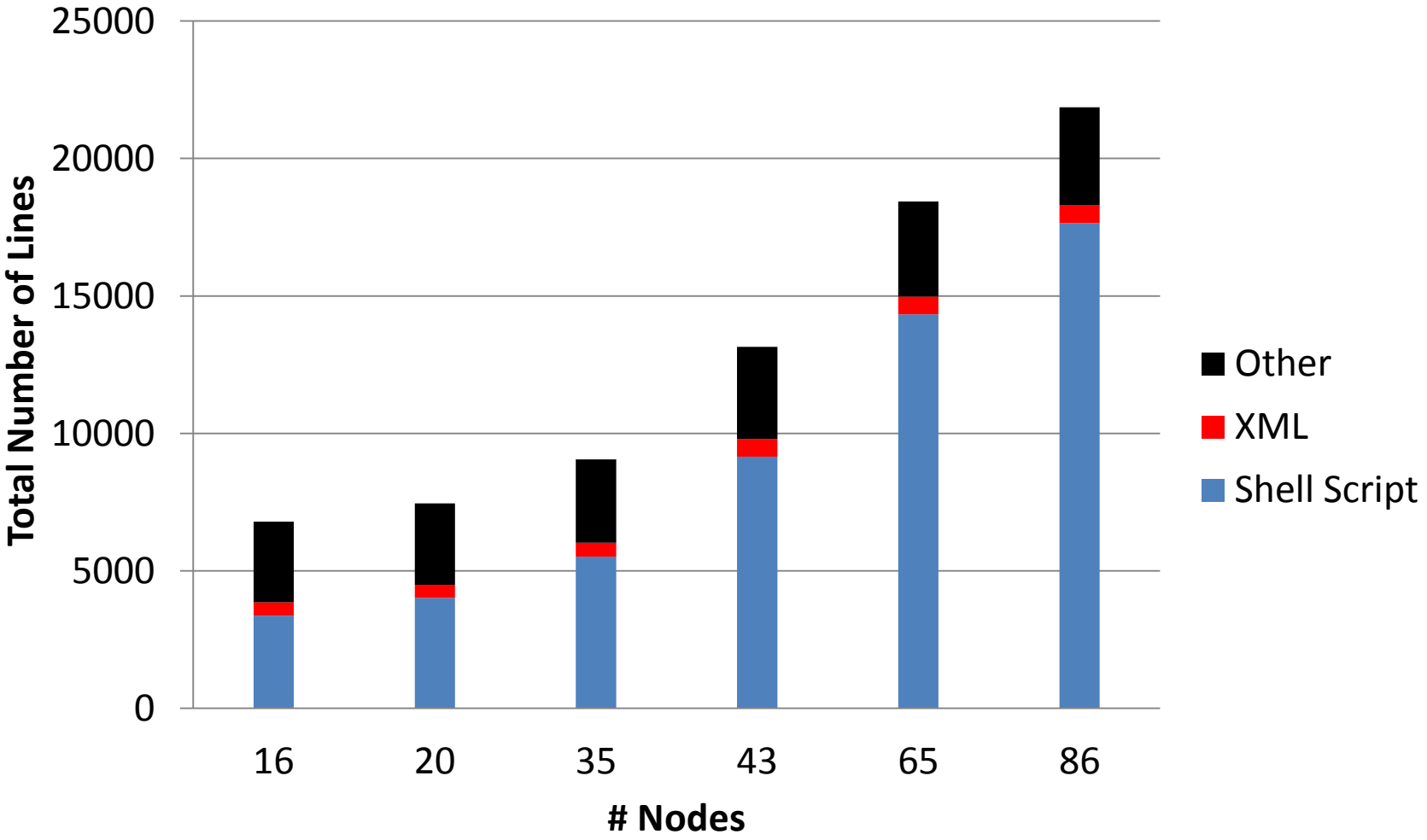
Type	Emulab	EC2	OpenCirrus	Elba	Wipro
Experiments	9215	1436	480	3567	120
Nodes	102687	25848	4987	9865	430
Configurations	392	86	28	163	8

Specification Changes vs. Changes in Generated Code





Number of Nodes vs. Generated Code



2013 Elba Publications

- **Q. Wang, Y. Kanemasa, et al**, Impact of DVFS on n-Tier Application Performance. In *TRIOS 2013*
- **D. Jayasinghe, et al**. An Infrastructure for Automating Large-scale Performance Studies and Data Processing. In IEEE BigData 2013
- **Q. Wang, Y. Kanemasa, et al.** "Detecting Transient Bottlenecks in n-Tier Applications through Fine-Grained Analysis", In *ICDCS 2013.*
- **Q. Wang, Y. Kanemasa, et al.** "An Experimental Study of Rapidly Alternating Bottlenecks in n-Tier Applications", In *IEEE Cloud 2013*
- **D. Jayasinghe, Fabio Oliveira, et al.** "AESON: A Model-Driven and Fault Tolerant Composite Deployment Runtime for Clouds", In *SCC 2013*
- **Junhee Park, Y. Kanemasa, et al.** "Variations in Performance Measurements of Multi-Core Processors: A Study ", In *SCC 2013*
- **Jack Li, et al.** "Performance Overhead Among Three Hypervisors: An Experimental Study ", In *IEEE BigData Congress 2013*
- **Y. Kanemasa, Q. Wang, et al.** "Revisiting Performance Interference among Consolidated n-Tier Applications", In *SCC 2013*

Ongoing Work

- Extending the data parser to support additional data formats.
- Extending the data warehouse to use No-SQL databases.
- Extending the visualization tool to support more customizable graphing capabilities.

Conclusions

- Help researchers efficiently creating, storing and analyzing performance measurement data.
- Open new opportunities and enable large-scale experiments above and beyond manual application testing.