Modeling Expands Value of Performance Testing for Big Data Applications

Boris Zibitsker, PhD
BEZNext
Chicago
bzibitsker@beznext.com

Alex Lupersolsky, PhD
BEZNext
Chicago
alupersolsky@beznext.com

ABSTRACT
Performance testing of Big Data applications is performed typically on small test environment with limited volume of data. The results of these types of tests do not take into consideration differences between test and production hardware and software environment and contention for resources with many applications in production environments. In this paper we will review application of the modeling for extending the results of performance testing, predicting how new application will perform in production environment. We will review how modeling results can be used to evaluate different options and justify decisions during design, development, implementation and performance management of the production environment.

Key Words
Performance Engineering; Performance Assurance; Big Data Applications; Big Data Infrastructure; Benchmark; Performance Testing; Performance Models; Performance Prediction.

1. INTRODUCTION

Causes of Performance Surprises
The selection of machine learning algorithms and their implementation, workload management, performance management and capacity planning decisions can affect usage of resources, performance and scalability of new Big Data applications.

Objective of performance testing of new applications is to identify and fix potential problems and minimize risk of performance surprises. Preparing and running performance tests is time consuming, often has a lot of limitations and does not take into consideration the complexity of production environment.
In [1, 2, 3, 6] authors describe goal, history and road map for Software Performance Engineering as a proactive approach using quantitative techniques to developing software systems that meet performance requirements.

In this paper we will review role of Performance Engineering and application of descriptive, diagnostic, predictive and prescriptive analytics during new Big Data application life cycle. Performance Engineering is a part of Performance Assurance (Figure 1), which also includes Dynamic Performance Management and Long Term Capacity Planning [22,23]. Performance Engineering includes data collection, workload characterization, workload forecasting, and performance prediction. Workload is a group of applications supporting specific line of business.

Performance prediction models [4] are built on measurement data collected during test of new applications and measurement data collected in production environments. Modeling

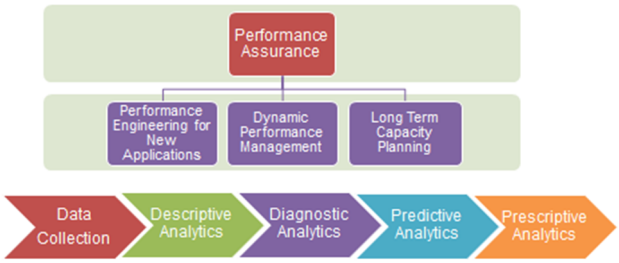

Figure 1. Performance Engineering Functions and Process focus on new Big Data applications design for performance

answers many questions including how the new applications will perform in production environment and how new applications will affect performance of existing applications.
Modeling enables evaluation of the different options [7, 8, 10, 21] to justify proactive actions necessary to continuously meeting Service Level Goals (SLGs) and reduce risk of performance surprises.

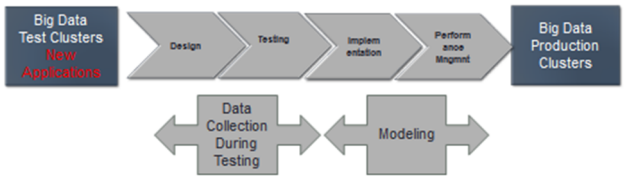

Figure 2. Performance Engineering Solutions for New Applications include Data Collection During Testing and Modeling

2. DATA COLLECTION IN TEST AND PRODUCTION ENVIRONMENT
In test and production environments we continuously collect measurement data about new and existing applications using our Linux, YARN, Kafka, Spark, Storm and Cassandra agents.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ICPE '17 Companion, April 22-26, 2017, L'Aquila, Italy
© 2017 ACM. ISBN 978-1-4503-4899-7/17/04…$15.00
DOI: http://dx.doi.org/10.1145/3053600.3053624

Typically Big Data clusters use Linux. From Linux /proc virtual directory and basic Linux commands (whatever is more effective for specific metrics) we get information about the node configuration and general CPU, memory, network, storage activity and disk space usage per node.

Going deeper into /proc subdirectories individual for each Linux process existing at the sample moment we get resource consumption by the process (CPU, memory, IO activity). Analysis of the process command arguments allows us to recognize the Big Data subsystem which the process (and its child processes) belongs to.

Remote interfaces of the subsystems (REST API, JMX) give us additional information about the applications running: users, throughput, elapsed time per unit of work, sometimes resources consumed (like vCore seconds and memory seconds in YARN), priorities and resource usage limits (YARN queues), internal parallelism (like tasks, executors in Spark).

Auto-discovery agent detects changes in hardware and software configuration. Collected data are used for workload characterization, diagnostics, root cause analysis, workload forecasting and performance prediction.

We aggregate measurement data in hourly performance, resource utilization and data usage profiles for each business workload or subsystems like YARN, Zookeeper, Spark, Storm, MapReduce, Tez, HBase, Cassandra, Kafka, etc.

We select representative time intervals to build Analytical Queueing Network Models [7, 10, 13, 15, 19] for evaluation of design and development decisions, infrastructure options (Figure 3).

Data collected during load testing in test environment are used to build models of the new applications [5, 6]. Data collected in production environment are used to build models characterizing performance and resource consumption of the production workloads.

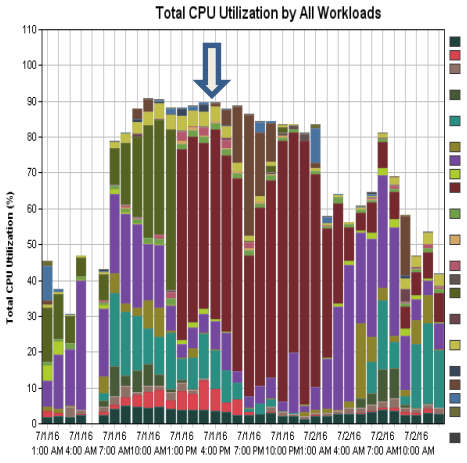

Figure 3. Selection of the Modeling Intervals (Business Workload Names on all Graphs are not shown intentionally)

Modeling results show the predicted impact of the expected workload and volume of data growth on performance of each workload. BEZNext modeling technology performs automatic model calibration [16]. Prediction results are used to develop proactive workload management, performance management and capacity planning recommendations to ensure that all applications will meet SLGs (Figure 4). The uniqueness of presented performance prediction is in approach enabling aggregate all applications in workloads, having performance, resource utilization and data usage profiles. It enables modeling complex Big Data multi-tier, distributed; parallel processing virtualized environments with mix workloads. Another unique aspect of the described models is in ability to predict the impact of the workload management changes. For example, ability to predict the impact of changing YARN rules, including priorities, concurrency, resource allocation to different workloads. After implementation of the recommendations the actual results are compared with the expected

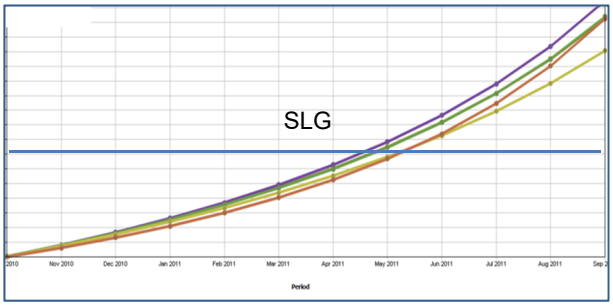

Figure 4. Predicted impact of workload and volume of data growth and determining when SLGs will not be met

Modeling results help to determine the minimum configuration which will be required to support expected growth and meet SLGs of current and new workloads.

3. PREDICTING NEW APPLICATION IMPLEMENTATION IMPACT

Resource consumption for new application in test environment is recalculated for production environment taking into consideration the difference in node types, number of nodes and software parameters (Figure 5).

Queueing Network Models [4, ,9, 11, 16, 18] were used to model test and production environment. Modeling results show how change of algorithms and design decisions will affect the performance of new application in production environment.

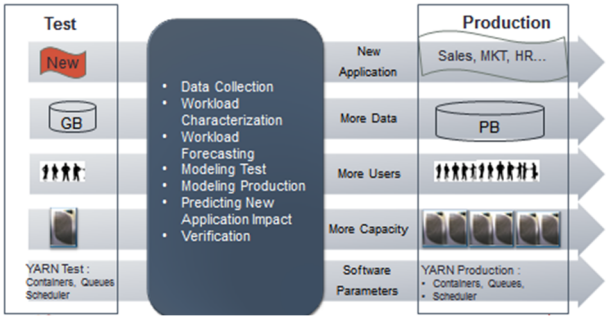

Figure 5. Predicting the impact of new Applications implementation and development proactive measures necessary to meeting SLGs for all workloads

Performance Measurement data collected during tests characterize resource consumption (Figure 6), response time (Figure 7), memory, storage and network utilization of new application.

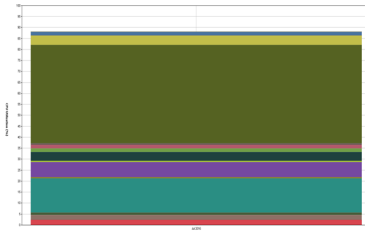

Figure 6. CPU Utilization by Different Applications in Test Environment

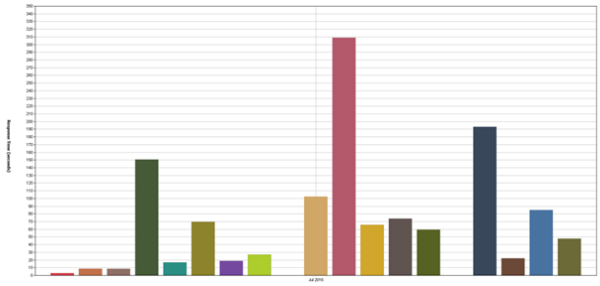

Figure 7. Response Time by Different Applications in Test Environment

Modeling results show what will be the impact of the new application implementation on performance of existing production workloads and determine when infrastructure will not be able to support SLGs as it shown on Figure 8.

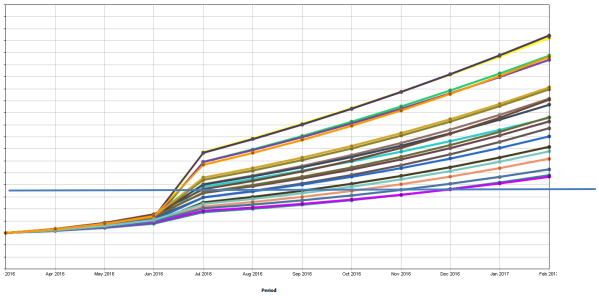

Figure 8. Predicted impact of new application implementation and determining when system will not be able to meet SLGs

Modeling also predicts what will be the impact of the new application implementation on resource consumption. Figure 9 shows how implementation of new application will affect the cluster's CPU utilization by different workloads.

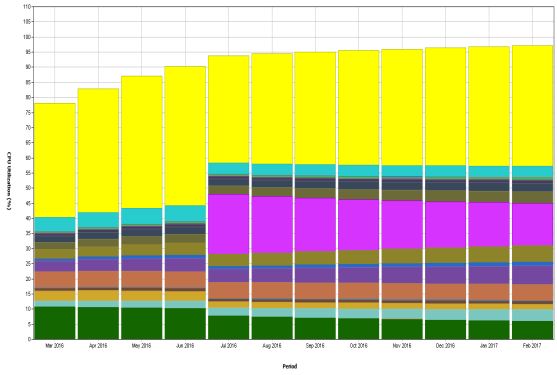

Figure 9. Predicted CPU Utilization after implementation of the new application in production

The same models generate predictions on how implementation of the new application will affect response time components for existing production workloads.

Predicted values of the largest components of the response time for each workload (Figure 10) are used to find current and future bottlenecks and justify proactive workload management and performance management actions.

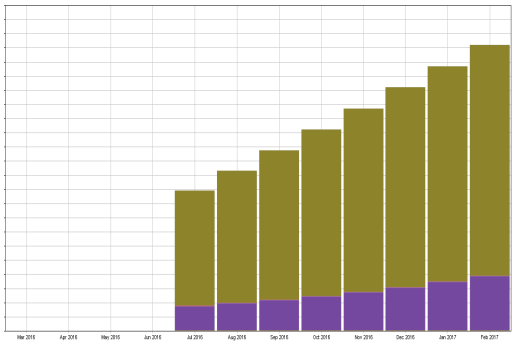

Figure 10. The biggest component of the Response Time for new workload in production will be "Waiting for Disk"

Each production workload has different SLGs, different performance, resource utilization profiles and priorities and will be differently affected by new application implementation. For example as it shown on Figure 11 for workloads "C*" the performance bottleneck will be CPU Wait Time.

After determining when workload will not be able to meet SLGs various scenarios / options can be evaluated based on the same model to determine the most effective workload management, performance management and capacity planning measures which should be implemented to continuously meet SLGs for each workload and when. For example, change priorities after implementing new workload, adding 4 nodes in January 2017 and 2 nodes in January 2018 (Figure 12) will be required.

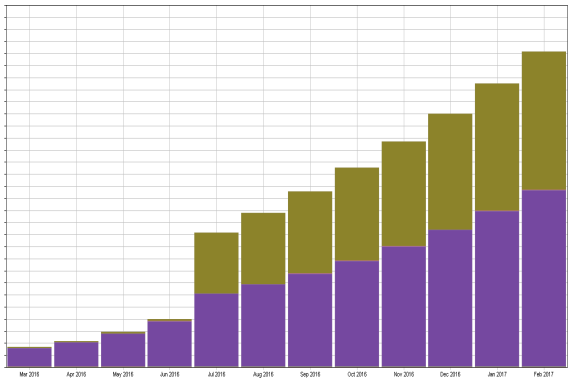

Figure 11. CPU Wait Time will be the largest component of the response time for existing production workload

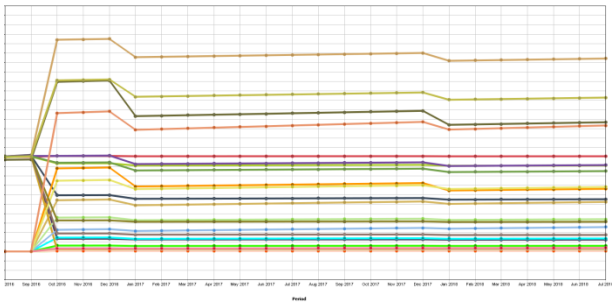

Figure 12. The modeling results for several options determine the most effective plan of proactive actions.

Predicted results show how planned hardware upgrade will affect the CPU utilization by each workload (Figure 13).

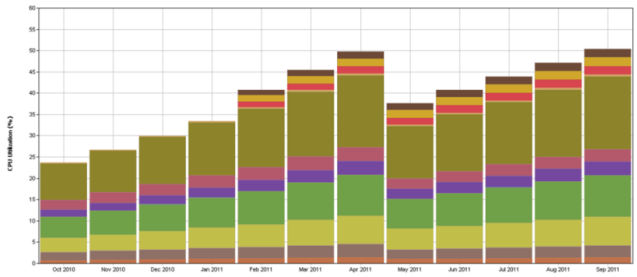

Figure 13. Predicted impact of the hardware upgrade

Machine learning Algorithms [5, 22] are used to determine Anomalies and their Root Causes after collecting measurement data during load tests after each build.

4. VERIFICATION

Verification of Results

Results of comparing of the actual measurement data (random values) with expected results (shaded area) are shown on Figure 14.

For this example the cause of difference between measured values and expected was a modification of application. It increased the CPU utilization and affected the response time.

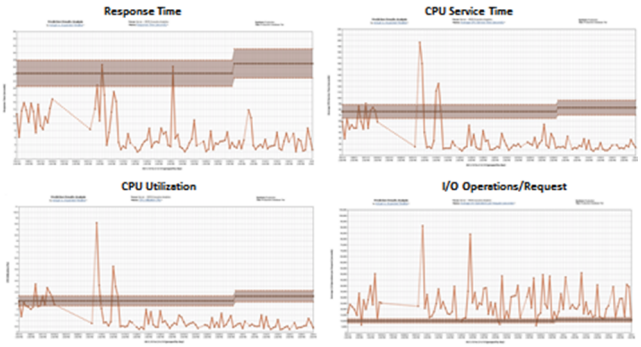

Figure 14. Comparison of the actual results with expected is a base for diagnostics and root cause analysis

Automated comparison of the Actual performance and resource utilization for each workload vs. Expected values enables diagnostics of anomalies. Root cause analysis and adjustment of the models and performance prediction scenarios are used to generate new / corrective proactive actions. It is a foundation for organizing continuous Performance Engineering and Performance Assurance process.

5. RELATED WORK

A lot of papers discuss application of Performance Engineering, [1, 2, 3, 5, 6, 12] and describe automation of performance testing and use of open source software. Several papers describe application of modeling and performance prediction to complement the performance testing results [6, 3]. Applications of ML predictive analytics described in [4, 12] and application of Control theory for automation of Performance Management in [14].

6. CONCLUSION AND FUTURE WORK

In this paper we reviewed applications of modeling during Big Data Applications life cycle.

We illustrated the application of Advanced Analytics for the justification and verification of decisions during Design, Development and Implementation of new Big Data applications The setting of realistic expectations and verification of results by comparing of the actual results with expected increases the confidence in decisions and reduces a risk of performance surprises. It is a first step in implementing collaborative Enterprise Performance Assurance process [17]

A future work focus is on incorporating optimization, control analytics [14] and Prescriptive Analytics for expanding role of Performance Engineering and Performance Assurance for Big Data applications by

7. ACKNOWLEDGEMENT

Special thanks go out to Dominique Heger from DHT Technologies, Alex Podelko from Oracle and our colleagues from BEZNext.

8. REFERENCES

1. Connie U. Smith, Performance Engineering Services "Software Performance Engineering Then and Now: A Position Paper"
2. Daniel A. Menasce "Software, Performance, or Engineering?" WOSP '02 Proceedings of the 3rd international workshop on Software and performance Pages 239-242
3. André B. Bondi Foundations of Software and System Performance Engineering: Process, Performance Modeling, Requirements, Testing, Scalability, and Practice Kindle Edition
4. Max Kuhn, Kjell Johnson, Applied Predictive Modeling, Springer, 2013
5. Alexander Podelko, Multiple Dimensions of Load Testing, CMG 2015
6. Connie U. Smith, Lloyd G. Williams, Performance Solutions. A Practical Guide to Creating Responsive, Scalable Software.
7. B. Zibitsker, IEEE Conference, Delft, Netherlands, March 2016, Big Data Performance Assurance
8. B. Zibitsker, Key note presentation "Role of Big Data Predictive Analytics" Big Data Predictive Analytics Conference, Minsk 2015
9. B. Zibitsker, Workshop on "Big Data Predictive Analytics", Big Data Predictive Analytics Conference, Minsk 2015
10. B. Zibitsker, "Application of Predictive Analytics for Better Alignment of Business and IT" Big Data Conference, Riga 2014,
11. B. Zibitsker, Key Note Presentation on "Big Data Advanced Analytics", Big Data Advanced Analytics Conference, Minsk 2016,
12. Dominique A. Heger "Big Data Predictive Analytics, Applications, Algorithms and Cluster Systems" ISBN:978-1-61422-951-3
13. B. Zibitsker, Strategic Performance Management for the Real Time Enterprise, Part 1 and 2, Measure IT, August – 9/2003
14. Dr. Joseph Hellerstein, *Microsoft Corp,* Yixin Diao, *IBM* Engineering Performance Using Control Theory
15. B. Zibitsker, Proactive Performance Management for Data Warehouses with Mixed Workload, Teradata Partners, 2008, 2009
16. B. Zibitsker, A. Lupersolsky, Modeling and Optimization in Multi-Tier Virtualized Distributed Oracle Environment, OOW 2009
17. B. Zibitsker, T. Jung, Teradata Partners, 2012, "Collaborative Capacity Management"
18. J. Buzen, B. Zibitsker, CMG 2006, "Challenges of Performance Prediction in Multi-tier Parallel Processing Environments"
19. B. Zibitsker, CMG 2008, 2009 "Hands on Workshop on Performance Prediction for Virtualized Multi-tier Distributed Environments"
20. Big Data Benchmark for Big Data https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench
21. BigBench: Toward An Industry-Standard Benchmark for Big Data Analytics http://blog.cloudera.com/blog/2014/11/bigbench-toward-an-industry-standard-benchmark-for-big-data-analytics/
22. B. Zibitsker, A. Lupersolsky, CMG 2016, "Performance Engineering for New Big Data Applications
23. B. Zibitsker, CMG 2016, "Performance Assurance for Big Data Applications"