Dynamic Performance Management of Big Data Clusters

Boris Zibitsker, PhD, BEZNext; Alex Lupersolsky, PhD, BEZNext; Dominique Heger, PhD, Data Analytica; Yuri Balasanov, PhD, University of Chicago; Mouttayen Manivassakam, Student of University of Chicago

Abstract

Large systems supporting mixed workloads incorporate workload management software allocating and controlling usage of resources and focusing on meeting Service Level Goals (SLG) of each workload [8,9]. When the response time of a workload exceeds SLG, we call it an anomaly. When the increase in response time by a workload happens periodically at the same time, we call it a seasonal peak [3].

Systems administrators are creating Workload Management rules to prevent anomalies and allocate enough resources during seasonal peaks to satisfy SLGs. When usage of resources increases, they allocate more resources, but they are still concerned with the risk of performance surprises.

In this paper we will focus on Big Data environment and will discuss how to apply modeling to find appropriate YARN Scheduler Queue settings to meet SLGs for Data Lakes, ad hoc and batch workloads and how to determine when additional hardware resources will be required.

1. Introduction

Workload management software for complex systems, including Big Data Clusters, Teradata, Oracle and others, use rules defined by Systems Administrators to control the allocation and usage of resources to meet SLGs of each workload [9, 10]. Each workload can be characterized by frequently-changing performance, resource utilization and data usage profiles. Response time is a critical part of SLG. Service time, Queueing time and Delay time are components of the Response Time. In this paper, we will review a methodology of defining YARN Scheduler Queues settings and justifying hardware configuration changes necessary to be sure that all workloads' response time will be below the SLG:

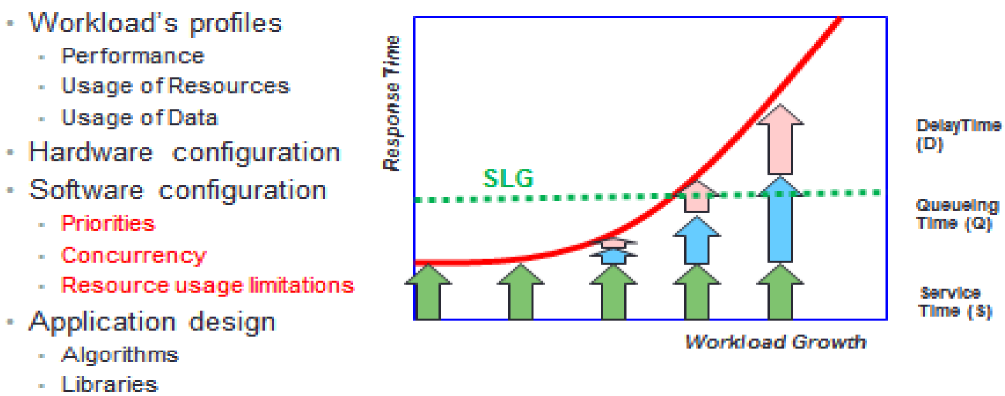

Figure 1. Workload Response Time includes relatively stable Service time, but constantly changing Queueing time and Delay time [9]. When response time exceeds the SLG, we call it an Anomaly.

1.1 Data Lake Resource Allocation & Management in YARN

YARN provides three standard schedulers to manage the tasks and control the execution and resource usage of the various applications that run concurrently on the cluster: Capacity (the default), Fair, and FIFO [11]. When the default scheduler is used, YARN rules are established by the Systems Administrator to assure the jobs are executed in the most effective and efficient manner based on the tasks and the hardware resources at hand.

As depicted in Figure 2, in most scenarios, the resources are decomposed by departments (layer 1) while the departments (parent) divide their allocated resources by actual projects (leaf). It should be pointed out that YARN provides many configuration parameters and options for the Capacity and the Fair scheduler, and there is the opportunity to design elasticity into the YARN rules. So, if resources are available (and not currently used by other projects or departments) a project or a department that has a need for additional resources can allocate them.

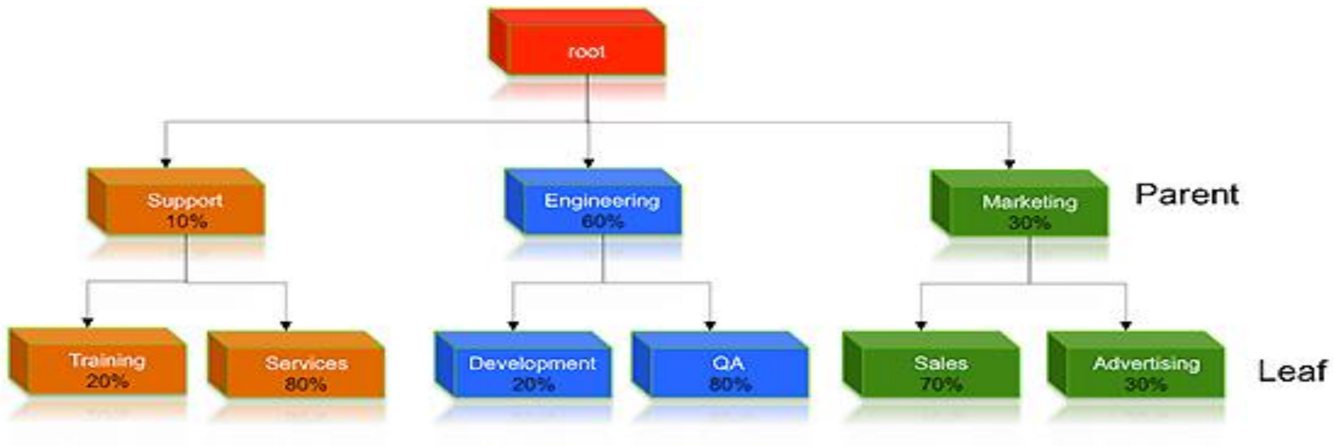

Figure 2: Cluster Capacity Scheduler Configuration

Below is an example of Configuration Parameters in YARN Capacity Scheduler leaf queues (capacitySchedulerLeafQueueInfo). To use common Workload Management terminology, we will refer to

Priority as YARN “absoluteCapacity” – share of the whole cluster resources, 0-100.
Resource Usage Limitation as YARN “absoluteMaxCapacity” – limit of the whole cluster resources usage, 0-100
Concurrency Limitation as YARN “maxApplications” – concurrency limit per queue.

An example of YARN Capacity Scheduler Queues Settings is shown on Figure 3.

Increase of **Priority, Concurrency Limitation** or **Resource Usage Limitation** for some of the workloads can improve their performance but negatively affect queuing and delay components of the response time of others.

When Systems Administrator decides that there are not enough resources and YARN Scheduler cannot support SLGs for some of the workloads, he or she increases the number of nodes and releases them later when resource utilization is reducing.

2. Workload Characterization in Big Data Environment

Aggregation of performance measurement data into workloads processed in YARN Scheduler Queues, auto-discovery of cluster hardware and software configuration is done every hour. Each workload has performance, resource utilization and data usage profiles.

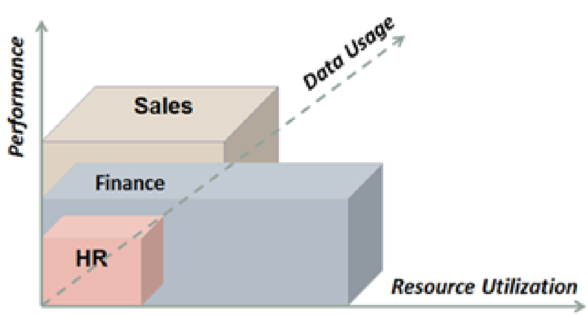

Figure 4. Each workload can be characterized by performance, demand for resources, and data usage profiles. Each workload has SLG and its tasks are processed in different YARN Scheduler queues

Analysis of workloads’ profiles and SLGs is used to set up YARN Scheduler Queues

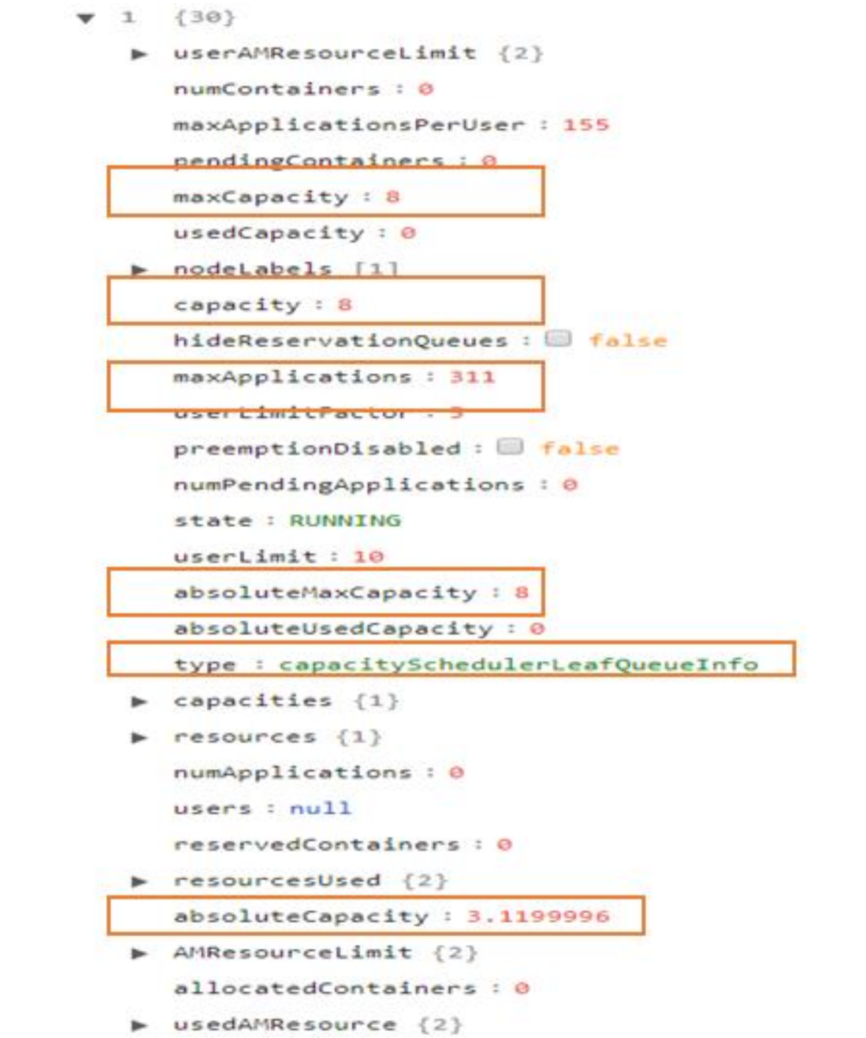

Figure 3. Example of YARN Capacity Scheduler Queues Rules

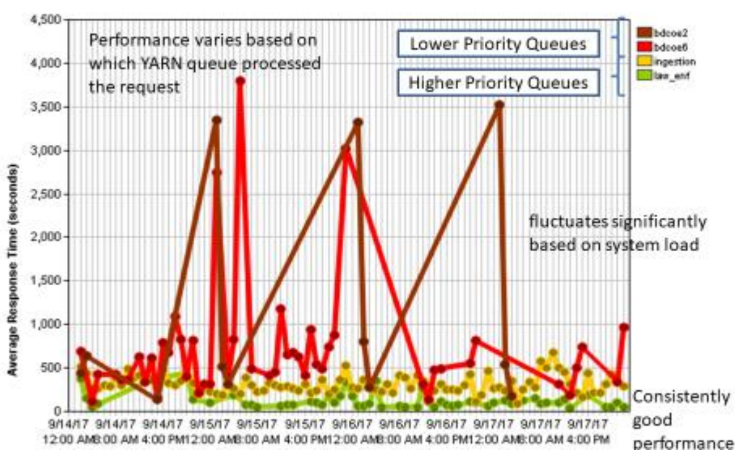

Figure 5. In this example Hive tasks (green and yellow) running in those Queues that have higher Priorities have better Response Time than tasks (red and brown) running in Queues that have lower Priorities

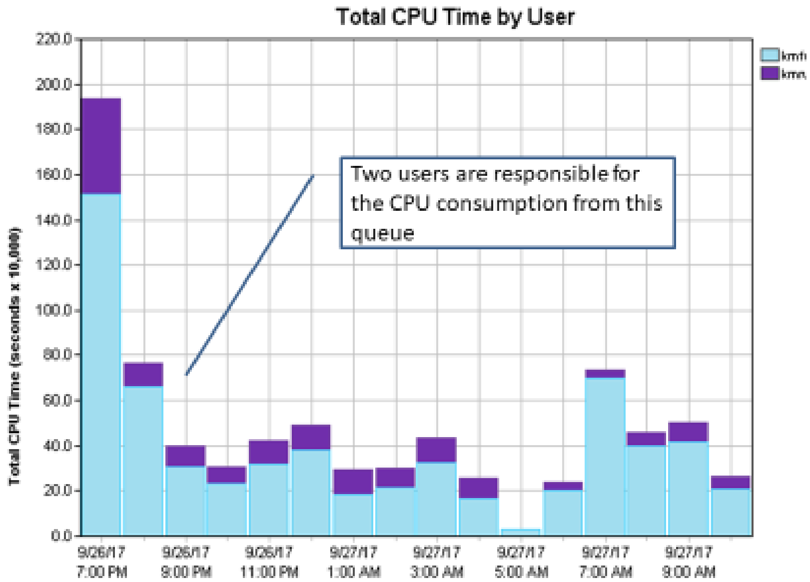

Figure 6. User with name “Light blue” consumed most of CPU Time allocated to a Queue

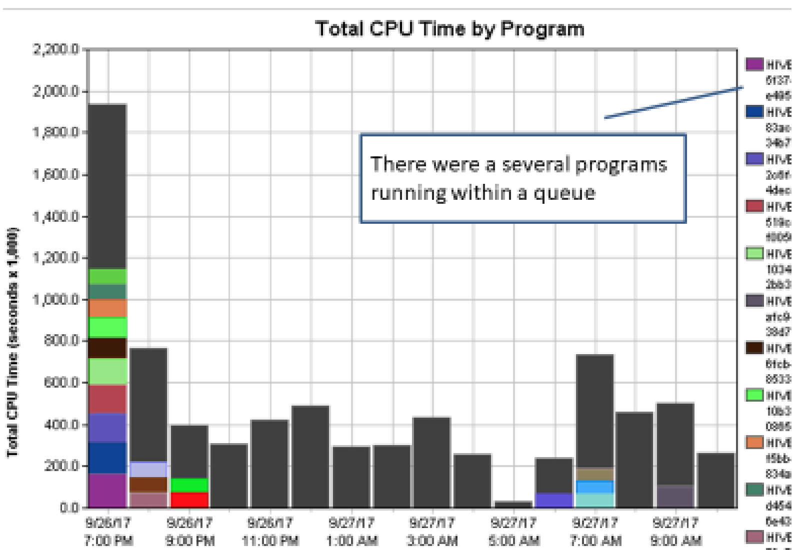

Figure 7. “Black” Program consumed most of the CPU Time allocated to a queue

Measurement data are used for determining seasonal peaks, anomalies and their root causes. They also are the input to models and prescriptive analytics to compare different options and generate recommendations how to change workload management rules and, if necessary, the hardware configuration to be able to liquidate anomalies and continuously meet SLGs for each workload.

3. Anomalies and Root Cause Determination

The hour when the average response time for a workload becomes greater than SLG for this workload is a start of Anomaly. The hour when workload’s response time becomes lower than the SLG is the end of Anomaly. The difference between start time and end time of anomaly provides the duration of the anomaly.

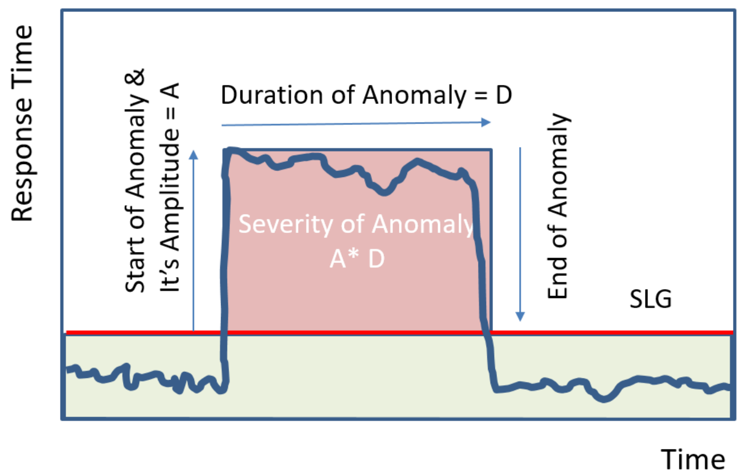

Figure 8. The Anomaly can be characterized by start and end, Anomaly Amplitude A, Duration of Anomaly D and Severity of Anomaly S = A * D.

Anomalies can be determined not only for Response Time, but also for Throughput, CPU utilization, I/O rate, Memory and Network Utilization.

workload name	timestamp	parameter name	root_causes program	root_causes workload	root_causes parameter	root_causes user	current datetime	workload value
HelpDesk3	95	RESPTIME	JD24	HelpDesk3	MEANCPUTIME	A59826D	1/6/2016 19:00	27662
HelpDesk3	196	RESPTIME	QR1	HelpDesk3	MEANIOOPS	A1D6886	1/11/2016 0:00	18926.67
HelpDesk3	242	RESPTIME	QR1	HelpDesk3	TOTALEXECCOUNT	86E3A08	1/12/2016 22:00	35385.38
HelpDesk3	345	RESPTIME	QR1	HelpDesk3	TOTALEXECCOUNT	595D7A0	1/17/2016 5:00	29783.08
HelpDesk3	390	RESPTIME	JD24	HelpDesk3	TOTALEXECCOUNT	E6A2208	1/19/2016 2:00	20710
HelpDesk3	400	RESPTIME	QR1	HelpDesk3	TOTALEXECCOUNT	95A57D0	1/19/2016 12:00	25925.16
HelpDesk3	433	RESPTIME	JD24	HelpDesk3	TOTALEXECCOUNT	59A4E01	1/20/2016 21:00	19005.13
Dev2	241	RESPTIME	TAB	Dev2	TOTALEXECCOUNT	LMERWIE	1/12/2016 1:00	49925.92
Dev2	251	RESPTIME	REFRESH3	Dev2	TOTALEXECCOUNT	LEEMIWR	1/12/2016 11:00	20792.93
Dev2	253	RESPTIME	SQLX1	Dev2	TOTALEXECCOUNT	ELWIRME	1/12/2016 13:00	29840.42
Dev2	255	RESPTIME	REFRESH3	Dev2	TOTALEXECCOUNT	IMLEEWR	1/12/2016 15:00	61304.62
Dev2	262	RESPTIME	REFRESH3	Dev2	TOTALEXECCOUNT	A09226C	1/12/2016 22:00	48259.96
Dev2	265	RESPTIME	TAB	Dev2	TOTALEXECCOUNT	IMWLERE	1/13/2016 1:00	18049.09
Dev2	271	RESPTIME	SQLX6	Dev2	TOTALEXECCOUNT	KALCBDA	1/13/2016 7:00	53420.59
Dev2	272	RESPTIME	TAB	Dev2	TOTALEXECCOUNT	MRILEEW	1/13/2016 8:00	69742.47

Table 1. Example of the determining Anomalies with the Response Time and their Root Causes. Workload HelpDesk3 Anomalies' Root Causes were high CPU time by user A59826D running program JD24 and high I/O rate caused by user AD6886 running program QR1

During certain periods, like the end of the month, the processing for some of the workloads takes more time. It is an expected anomaly. Let's call such repeatable anomalies as "seasonal peaks." Other anomalies happened at random time and they are unpredictable.

We have developed an algorithm to find the seasonal peaks. For expected seasonal peaks, we can proactively change the YARN Scheduler Queues settings.

An example of finding the seasonal peaks based on analysis of the historical data it shown in Table 2.

WORKLOAD NAME	PARAMETER NAME	PERIOD	MEAN DURATION	MEAN AMPLITUDE
Accnt1	RESPTIME	1	3	72493.71
Accnt2	RESPTIME	1	2	0
Admin2	RESPTIME	1	3	41674.11
Dev1	RESPTIME	1	5	8712.97
HR4	RESPTIME	1	1	37763.97
Load0	RESPTIME	1	2	4137.35
Load1	RESPTIME	1	1	202333.27
Load3	RESPTIME	1	1	210877.68
Load4	RESPTIME	1	1	177527.29
Load7	RESPTIME	1	1	393990.27
Load8	RESPTIME	1	1	22012.14
Other	RESPTIME	1	3	8821.97
Prod1	RESPTIME	1	1	84995.01
QA0	RESPTIME	1	1	83520.66
QA2	RESPTIME	1	2	67362.99
TechSupport2	RESPTIME	1	1	42580.35

Table 2. Example of determined Daily Seasonal Peaks

4. Methodology of Adaptive Control

Adaptive control of distributed computing environment is a challenging multidimensional problem that at this point may not be ready to be easily solved by pure analytical approach, nor by traditional methods of optimal control. One of the reasons is high dimension of state space of workloads and their performance characteristics and highly cardinal nature of possible controls [7].

The proposed adaptive control method is a combination of analytical modeling of processes in multi-node cluster and machine learning applied to history of measurements data of cluster performance, including response time for vector of workloads and root causes of occurred anomalies [4.5].

Machine learning algorithms are applied to analyze patterns of anomalies and root causes to understand their major types and frequencies [1,2].

Analytical model uses history of anomalies and root causes and applies to them multiple combinations of control factors to collect enough evidence for selecting a better combination of rules for each anomaly scenario. The analytical model step of the approach may require additional data analysis and machine learning techniques, like reduction of dimensionality of anomalies and experimental design reducing the number of combinations of control factors.

When Workload Management changes alone cannot satisfy SLGs, modeling determines the hardware upgrade required to meet SLGs.

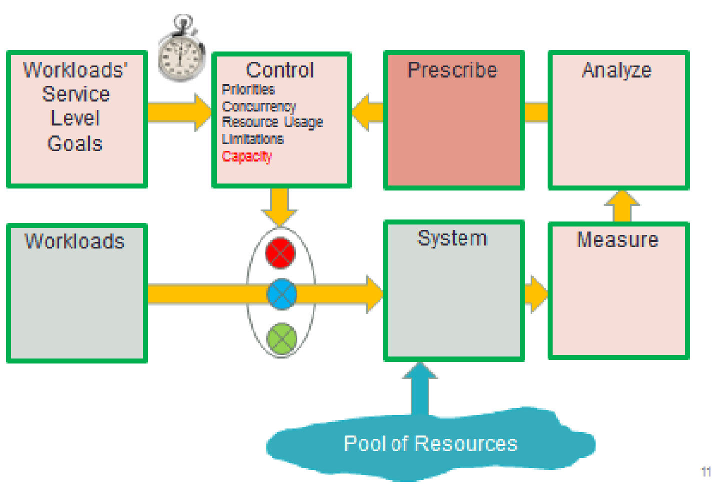

Figure 9. When workload management changes can't satisfy SLGs, the queuing network model Queueing Network Models (QNM) is used to determine the appropriate hardware upgrade required to Meet SLGs

4.1 Application of QNM for Dynamic Performance Management and Capacity Planning

Smart Search algorithm and QNM are implemented to evaluate options of YARN Scheduler Queues settings to meet SLG for every workload.

The idea of the algorithm is following:

1. Find a workload with the biggest positive difference between predicted response time and SLG.
2. Find the most probable cause of SLG violation based on the longest response time component (CPU service or queuing time, IO service or queuing time, network service or queuing time, etc.) for this workload predicted by QNM.
3. If it is service time, then application tuning or hardware upgrade is required
4. If itis a queeing time issue, then one of these options is used:
    a. Increasing the priority of the problematic workload waiting for resource or reduce concurrency for other workloads using excessive amount of resources. It usually improves performance of this one workload but worsens performance of all other workloads.
    b. Find another workload that utilizes most of the critical resource. Reduce either its Priority or Concurrency limit or Resource utilization limit. Reducing any of these worsens the performance of the selected workload but improves the performance of all other workloads.
    c. If all reasonable changes of the YARN parameters did not help satisfy all workloads' SLG, additional resources are required
5. Estimate the performance of all workloads for the new YARN parameters and different hardware configurations using the QNM.
6. If all workloads' SLG are satisfied, stop the analysis. If not return to #1.

Notes

1. Workload priority here means the relative share of available resources assigned to the workload.
2. It seems it'\is better to reduce the workload's priority rather than reducing resources or concurrency limit because the latter is unconditional: the limited workload cannot use more, even if the resource is available.
3. The numeric value of the proposed change is defined by the algorithm based on the difference between the predicted workload's response time and its SLG.

The algorithm determines required changes and provides expectations of the response time and usage of resources for each workload It also enables verification of recommendations by comparing the actual measurement data with expected. Here are two examples.

Example 1. Applying QNM to Predict Anomalies and Determine YARN Scheduler Queues parameters changes necessary to meet SLGs for all workloads

Figure 10. According to prediction results, Response time for three workloads (green, brown and blue) will not meet SLGs. Can changes in YARN Scheduler Queues required be determined to meet SLGs for all workloads?

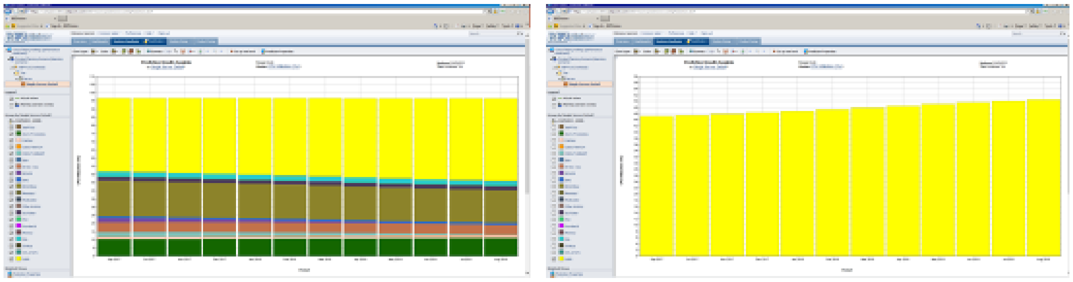

Figure 11. Yellow workload “User” uses about 50% of CPU resources

Figure 12. Reduce Priority for “Yellow” workload will meet SLG for “Blue” workload, but “Green” and “Brown” workloads will not meet SLGs

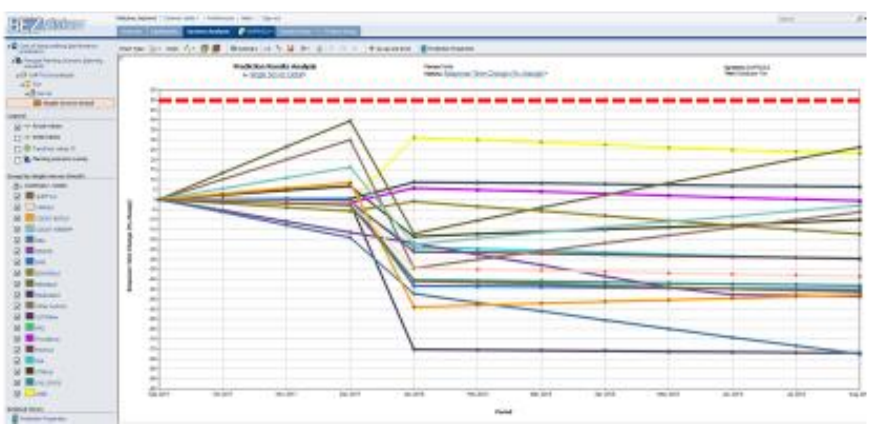

Figure 13. Reducing Priority for “Yellow” workload and increase for “Green” and “Brown” is sufficient to meet SLG for all workloads

Example 2. When all workloads violate SLGs, additional resources are required. This example illustrates use of QNM and Capacity Planning Recommender to justify minimum upgrade required to meet SLGs for all workloads

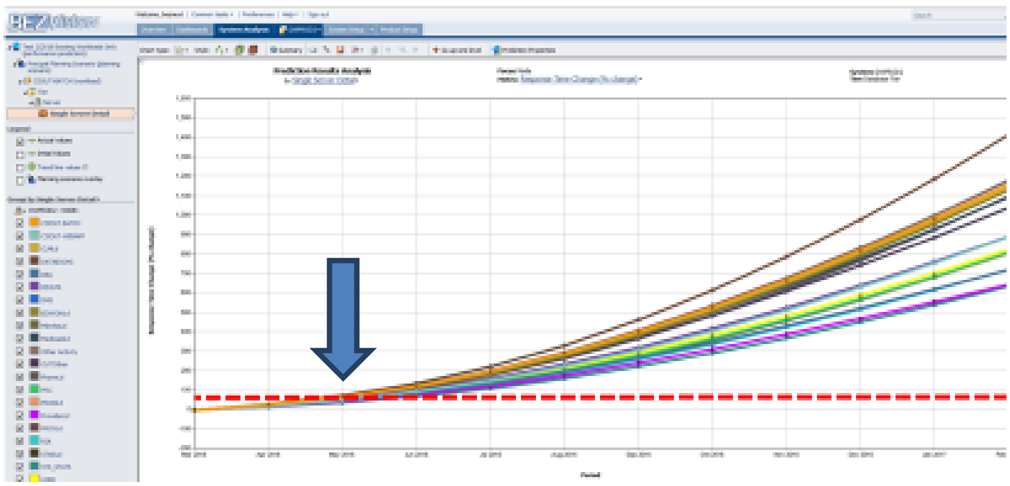

Figure 14. According to prediction results all workloads will violate SLGs in 2 months and workload management alone will not be able to solve the problem. Additional nodes will be required.

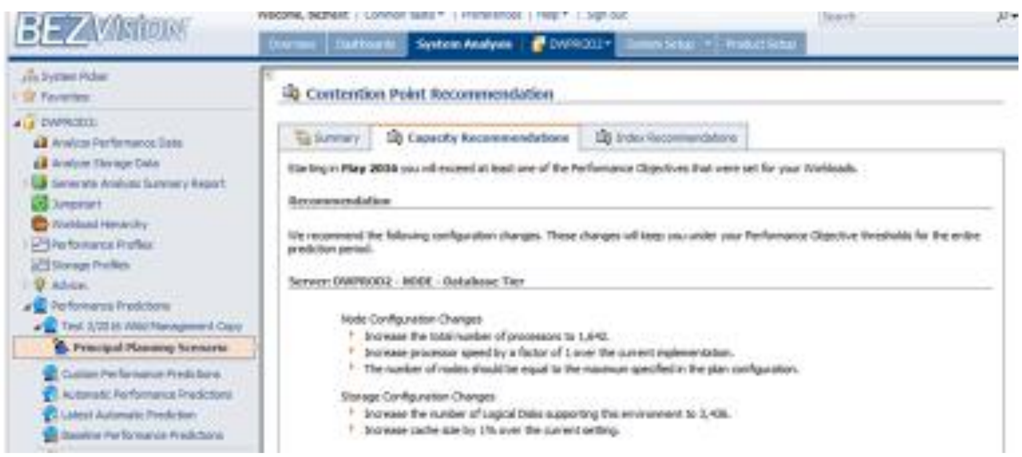

Figure 15. According to Capacity Planning Recommender increase from 1456 to 1642 CPUs and increase from 3432 to 3436 disks in a Cluster are required to meet SLGs for all workloads during next 12 months

4.2 Application of ML algorithms for Dynamic Performance Management

After determining Anomalies, Root Causes and Seasonality (Tables 1 and 2), the ML algorithms for Clustering of Anomalies, Mapping Anomalies to Control Parameter Changes [6] is applied to identify Control Parameters Changes which helped in the past to satisfy SLGs.

Mapping Anomalies to Control Parameter Changes

Identified anomalies are traced back to the control parameter changed at that time .

```
> head(diff(equiv_priority_matrix),3)
   3rdPTYLd ACTUA Actuarial APPL       APPLDEV     CAPNLd CHS        CIIld      CIILd CIIOUT-BATCH CIIOUT-WEBAPP         CIS        CISLd
X2        0     0         0    0  0.0077647058  0.006453476   0 -0.008250853 -0.01482353   0.00000000   0.004941176 0.009882353 -0.005090909
X3        0     0         0    0 -0.0076764705  0.001555431   0  0.008250853  0.00000000   0.01482353   0.000000000 0.000000000  0.009882353
X4        0     0         0    0  0.0008529412 -0.012950084   0 -0.014823529  0.01235294  -0.01482353   0.000000000 0.008641711 -0.009882353
       CLINICAL       CLMLd       COALd COCV20Ld CompHlthSn DATAEXCHG        DBA          DBC       DEVLPR          DMS     DntlvnLd EAS
X2 -0.0007058824 0.004941176 -0.004941176        0          0         0 0.01976471  0.009345882 -0.009882353  0.000000000 -0.006636105   0
X3 -0.0041978609 0.002470588 -0.004941176        0          0         0 0.00000000 -0.003778547  0.009882353  0.004941176  0.001694929   0
X4  0.0086042780 0.007411765  0.003705882        0          0         0 0.00000000 -0.002397924  0.000000000 -0.004941176 -0.014823529   0
   ECRLd      EDLR1Ld     EDLR1LOAD EDMLd   EDWARDLOAD    EDWHstLd EDWOthLd       EF5500      EF5500Ld        EGDLd     HELPDESK HR
X2     0  0.007411765 -0.0050160425     0  0.009882353 -0.007071444        0 -0.003364706  0.000000000  0.019764706  0.009882353  0
X3     0  0.012256684 -0.0007058827     0 -0.009882353  0.003487889        0  0.000000000  0.000000000 -0.014823529 -0.001411765  0
X4     0 -0.009786096  0.0089306724     0  0.009882353  0.002441523        0  0.000000000 -0.009882353 -0.004941177 -0.003245172  0
         HRALd        Load        LOAD     MbrshpLd          MCP MCPLd   MedicaidLd    OplSoln        OPS Other Activity         OUT OUTOther
X2  0.004707594 0.002546154 0.000000000  0.009882353 -0.014823529     0  0.007411765  0.0129148  0.0129148          0.024 0.009882353        0
X3 -0.008103530 0.000000000 0.000000000 -0.009882353  0.019764706     0 -0.012278075 -0.0129148 -0.0129148          0.000 0.000000000        0
X4  0.009882353 0.000000000 0.009882353  0.000000000 -0.003051903     0  0.002395722  0.0000000  0.0000000          0.000 0.000000000        0
          PBM      Pharm      PharmLd          PRC      PRODLd   ProviderLd       PROVLd        RDSLd        RHILd          RIA        RVNULd
X2 0.00000000 0.00000000 -0.009882353  0.009882353  0.00000000 -0.001230190  0.000000000  0.000000000 -0.009882353  0.009882353  0.0009331650
X3 0.00000000 0.00000000  0.009882353 -0.002964706  0.00000000  0.000326904  0.000000000  0.009882353  0.000000000  0.000000000 -0.0001249998
X4 0.01510777 0.01510777 -0.009882353  0.002964706 -0.01482353  0.002765102 -0.007411765 -0.007411765  0.009882353 -0.004941176 -0.0049411765
         STARLd SVC    SYS_STATS     Unknown         USER
X2  0.005558823   0  0.004941176 0.001970588 -0.004941177
X3 -0.009882353   0 -0.004941176 0.002545454  0.004941177
X4 -0.004941176   0  0.004941176 0.002395722  0.035786096
```

Figure 16. Control Parameter Priority Change Matrix

The matrix above is a snapshot of the complete matrix that captures only data for four hours. Here the columns represent the workloads and the rows represent the differencing of adjacent time periods. For example, Row X2 is the changes done between HR1 and HR2, Row X3 represents the change done between HR2 and HR3 for individual workloads.

Similarly, the change matrix for CPU Utilization Limitation and Concurrency settings for all workloads are built.

Identification Significant Control Parameter for each Workload

In this step, the control parameters which have the strongest impact on the Workload's Performance, whether it is concurrency limitation, priority or CPU utilization limitation are determined. The objective here is to identify both negative and positive correlations. For example, reducing the CPU Utilization Limitation for one of the workloads is likely to degrade the system's overall response time for other workloads.

Results of applying ML algorithms to justify YARN Scheduler Queues settings will be a part of the presentation.

4.3 Application of ANN for Data Lake Capacity Planning

In most Data Lake setups, the YARN rules are not dynamically changed, so they depict a rather static entity that may need to be fine-tuned periodically. Further, the statement can be made that most Big Data clusters are vastly over-configured and hence an actual resource shortage is rather unusual.

In scenarios where the cluster resources are being highly utilized and there is a potential for over-committing the resources, one of the authors of this report has been working on incorporating dynamic capacity planning based on artificial neural networks into the cluster maintenance cycle [12].

It must be pointed out, however, that the objective is not to adjust the YARN rules *per se,* but to determine how many additional nodes should be moved into the cluster to stay within the SLGs (taking advantage of the great horizontal scalability of the Apache Big Data projects).

This process works both ways, so if a downward resource demand is projected, the node count in the cluster can be reduced. In practice, this approach has proven to be more effective than dynamically changing the YARN rules. To reiterate though, any cluster or YARN rule changes are only necessary if the cluster resources are being fully utilized, a scenario that is rather unusual when Big Data clusters are vastly over-configured.

5. Summary

This paper includes a methodology and several examples illustrating application of modeling for Dynamic Performance Management of Big Data Clusters and challenges of data collection, workload characterization, anomaly detection, root cause determination, identification of seasonal peaks and justification of changes in YARN Scheduler Queues Control Parameter necessary to continuously satisfy SLGs for all workloads.

Related Work

Many papers discussing Autonomic Computing and application of the Control Theory were presented at IEEE conferences ICCAC. Some of them focus on challenges of the workload and resource management in Big Data environment. Most of them focus on batch Map Reduce and Tez batch workloads. Just a few papers cover issues of autonomic computing in real time environment based on Spark, Storm and Kafka.

Future Work

We are working on applying Prescriptive Analytics based on Machine Learning and AI to justify Performance Engineering, Performance Management and Capacity Planning decisions during Big Data applications life cycle. We will continue collaboration with several Universities, IBM and research organizations to develop Recommenders for Performance Assurance

Key Words

Big Data Dynamic Performance Management; Workload Management; Capacity Planning; Autonomic Computing; Self-Aware Computing; Anomaly Detection; Problem Prediction; Root Cause Analysis; Prescriptive Analytics

Acknowledgments

Special thanks go out to our colleagues from BEZNext.

References

1. Daniel A. Menasce “Software, Performance, or Engineering?” WOSP '02 Proceedings of the 3rd international workshop on Software and performance Pages 239-242
2. Max Kuhn, Kjell Johnson, Applied Predictive Modeling, Springer, 2013
3. B. Zibitsker, IEEE Conference, Delft, Netherlands, March 2016, Big Data Performance Assurance
4. B. Zibitsker, Key note presentation “Role of Big Data Predictive Analytics” Big Data Predictive Analytics Conference, Minsk 2015
5. B. Zibitsker, Key Note Presentation on “Big Data Advanced Analytics”, Big Data Advanced Analytics Conference, Minsk 2016,
6. Dominique A. Heger “Big Data Predictive Analytics, Applications, Algorithms and Cluster Systems” ISBN:978-1-61422-951-3
7. Dr. Joseph Hellerstein, *Microsoft Corp,* Yixin Diao, *IBM* Engineering Performance Using Control Theory
8. B. Zibitsker, Proactive Performance Management for Data Warehouses with Mixed Workload, Teradata Partners, 2008, 2009
9. J. Buzen, B. Zibitsker, CMG 2006, “Challenges of Performance Prediction in Multi-tier Parallel Processing Environments”
10. B. Zibitsker, CMG 2008, 2009 “Hands on Workshops on Performance Prediction for Virtualized Multi-tier Distributed Environments”
11. Heger, D., “Introduction to Apache YARN Schedulers & Queues”, Data Analytica, www.mlanalytica.com, 2016
12. Heger, D. "Machine Learning in the Realm of Big Data Analytics", Fundcraft Publication, ISBN 978-0-578-19095-2, March 2017.