



APPLICATION PERFORMANCE MANAGEMENT (APM) WITH BIGSTREAM XRAY

Introduction

In the realm of data analytics and machine learning (ML), APM has become a key aspect of application development. APM's goal is to provide detailed analysis of big data applications with respect to performance and potential improvements therein. Data scientists and developers as well as operations professionals can use it to understand how to improve platform configuration, application structure and infrastructure for higher performance.

This is becoming relevant as analytics and models leverage larger, more unstructured, and formerly dark data sources. It is also particularly relevant when utilizing emerging acceleration technologies, to understand how and where to apply them.

To meet these ends, Bigstream has developed the xRay APM tool which profiles big data applications, specifically Apache Spark. It works seamlessly with Bigstream Hyperacceleration (see: <https://blog.bigstream.co/resources/hyper-acceleration-with-bigstream-technology>), in a synergistic way providing users with applicable analysis of their production application performance, as well as the potential benefits of applying acceleration technology. xRay is easy to install, provides fully web-based analysis and runs both in the cloud and on-premise. Note that xRay can also be used standalone to assess workload performance, independent of acceleration.

The rest of this paper is structured as follows. We first describe xRay and the answers it provides. We then briefly discuss how Bigstream Hyperacceleration works in tandem with xRay. Finally, explain how to get started with xRay, as well as describe plans for future releases.

xRay Description

The overall goal of the tool is to provide in-depth and useful insight to the user into the performance aspects of her Apache Spark application. It runs as a listener class on the user's production application in their environment, i.e. not a benchmark or surrogate. It provides basic application run information such as configurations and runtimes of the application, stages and tasks. It also does log data signature analysis of the run to provide recommendations to the user as to how to improve performance. Finally, it provides an assessment of the potential in applying Bigstream Hyperacceleration technology to the

application. Information about operators, UDFs, stage structure, data formats and more inform this acceleration assessment.

Given the goal of performance, xRay provides answers to the following questions for the Spark user automatically, among others:

- What are the configurations (i.e. Spark, Yarn etc.) that I am using? Can they be changed to improve performance?
- Where is most of the time being spent in my application (compute, I/O, idling etc.)?
- What is the bottleneck stage of my application? What operators/data is it using?
- Does my data have skew? Do certain stages have straggler tasks?
- Which stages can be accelerated either via software or hardware acceleration? Are these the key stages in my application?
- How much acceleration can I expect?

It is interesting to note that the majority of these questions have to do with basic application performance, and are not specific to acceleration. Therefore, xRay is designed to be a performance tool that works alongside acceleration; there is no requirement for using acceleration to use xRay. We now briefly describe how xRay can be used in conjunction with Bigstream acceleration technology.

xRay and Bigstream Hyperacceleration

The xRay tool works in concert with Bigstream acceleration technology to provide high performance to the user. As stated earlier, there is no requirement for using acceleration to use xRay and vice versa. However, Bigstream Hyperacceleration and xRay are made to work synergistically to maximize performance for the user with zero code change.

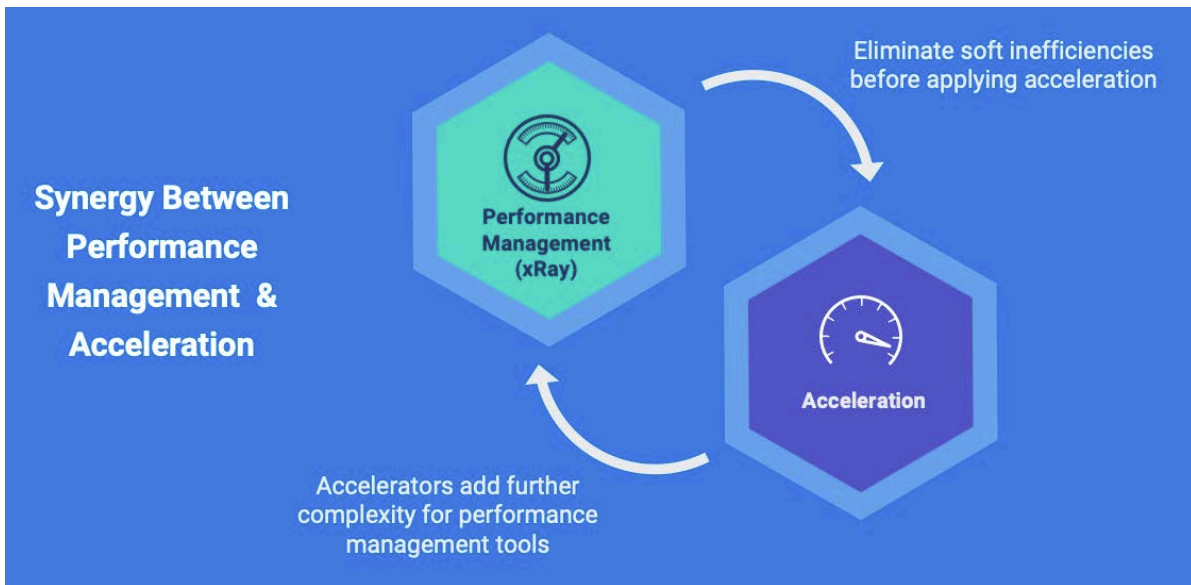


FIGURE 1. xRay/Acceleration Synergy

Figure 3 illustrates the synergy between xRay and acceleration when applied to optimize applications. A priori, it provides information about applicability of acceleration, and estimates of performance that will be reaped. For example, Bigstream accelerates the Dataframes/Datasets API (i.e. SparkSQL), and xRay analysis can ensure that these APIs are the ones used by the application. After this, the user applies acceleration and then re-runs xRay to understand the performance gains made. We now describe how to get started.

Running and Analyzing with xRay

This section describes how to get started with xRay for the reader’s Spark application, and gives an example of the analysis it provides. This example is a subset of the functionality provided; full documentation and functionality are available directly from the website, insight.bigstream.co

Running an application with xRay

After signing up, instrumenting and running is a simple 3-step process, illustrated in Figure 4. This figure is taken directly from the website..

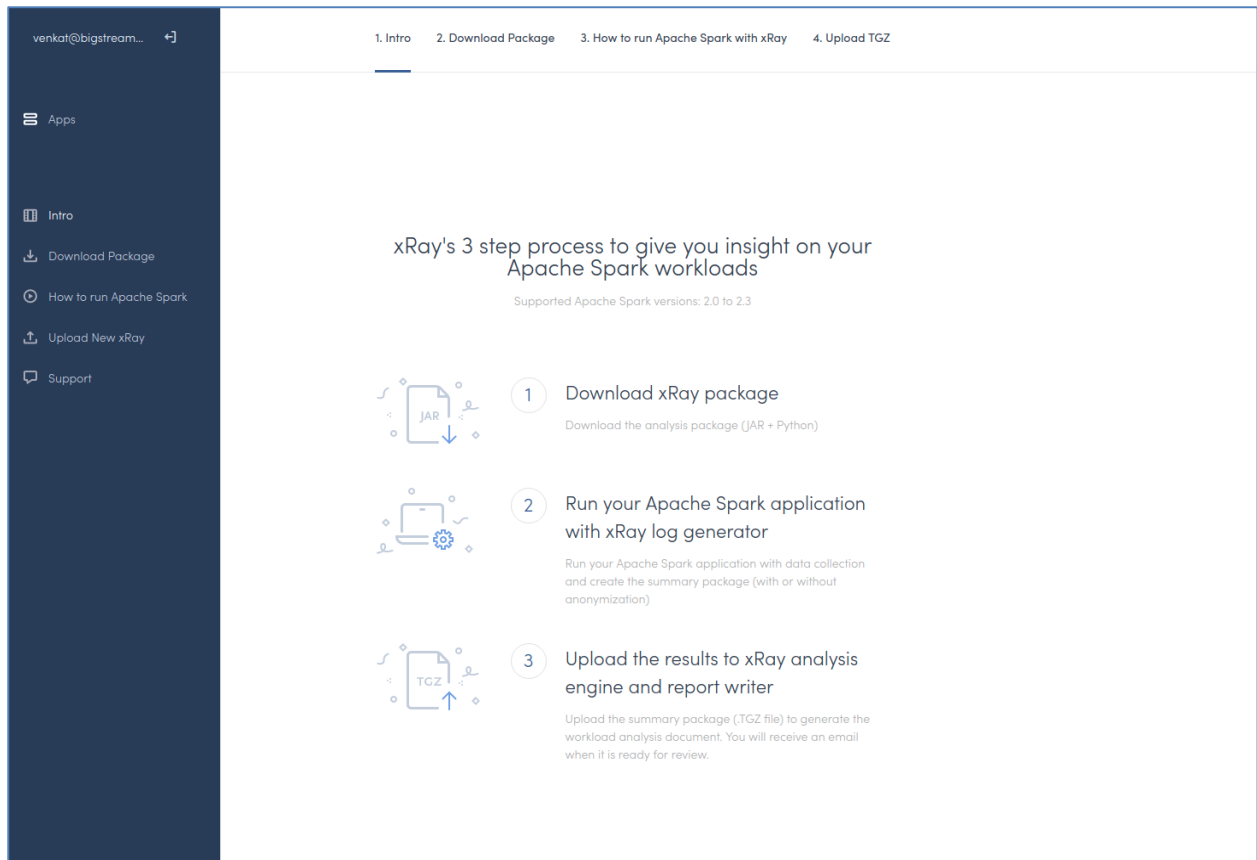


FIGURE 2. User Start Page

Referring to the list of links on the left side of Figure 2, the user can click on the “Download Package” link to select the xRay package suitable for the OS version being run on. Currently, four Linux types are available: RedHat, Amazon EMR, Ubuntu and CentOS. Check the download page for exact versions to match the readers’s infrastructure. These packages work on both cloud and on-premise infrastructure equally.

Once the appropriate package is unpackaged on the master node of the cluster, the Spark application can be run with xRay by providing exactly two additional flags to the Spark run command (i.e. command line). Details are in the “How to run Apache Spark” link shown in the left of Figure 2.

After running the Spark application, the last step is to upload the resultant produced logs to the xRay website via the “Upload New xRay” link. There is an anonymization function provided by the package, again documented on the link.

After this, in a few minutes, the user will find a web-based, clickable report in their dashboard on the xRay site, ready to do analysis. We examine some aspects of this report next.

xRay Report Analysis Example

We present an example analysis that is possible with xRay on a Spark workload. This example is intended to be illustrative; it gives the look and feel of the xRay web interface for a report. There are many more capabilities that xRay has for APM, which we list after this expose.

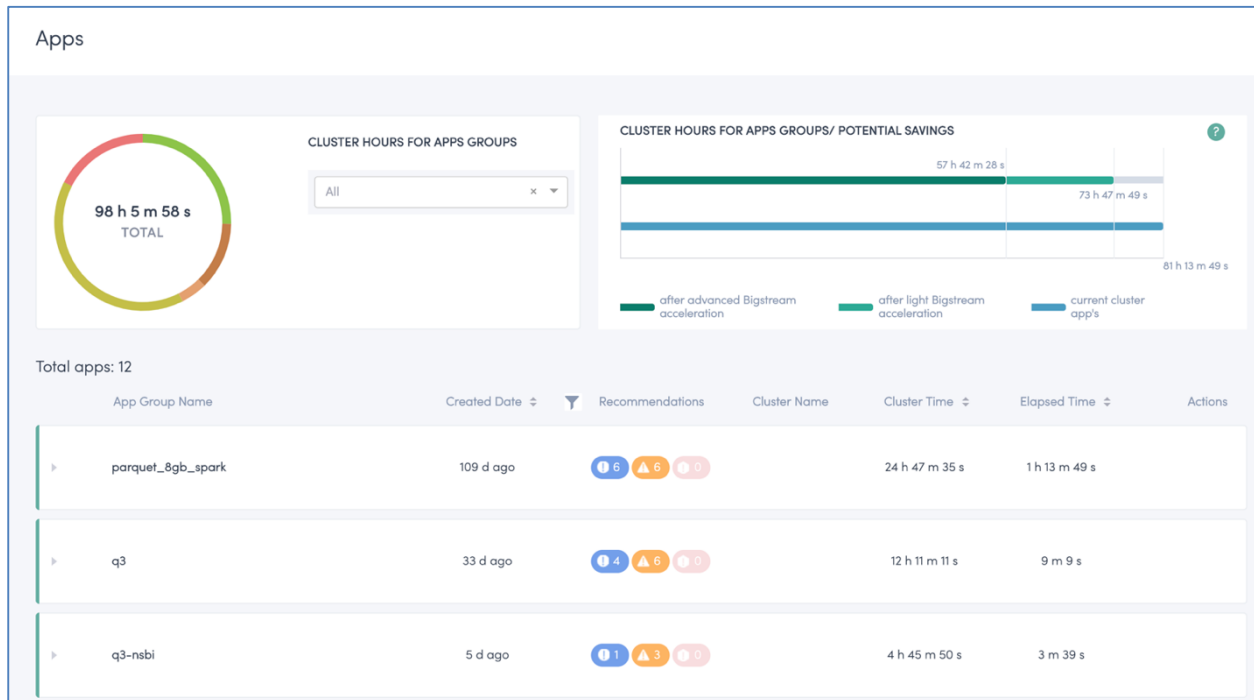


FIGURE 3. Dashboard View

Figure 3 shows the top portion of the user dashboard, with several application log outputs uploaded. In the upper left of the dashboard, summary information is presented including aggregate runtime for all of this users' applications. The list of applications is below this, organized into "App Groups" which are groupings of apps defined by the user. These groups are user-defined and are a convenient way to organize related application runs' information. Groups are clickable to drill down into specific application information, as needed. In addition, each application group has colored icons indicating recommendations, where Blue="Info", Orange="Warning", and Red="Critical", provided by xRay's analysis of the application logs. Clicking an application group provides access to individual application run reports.

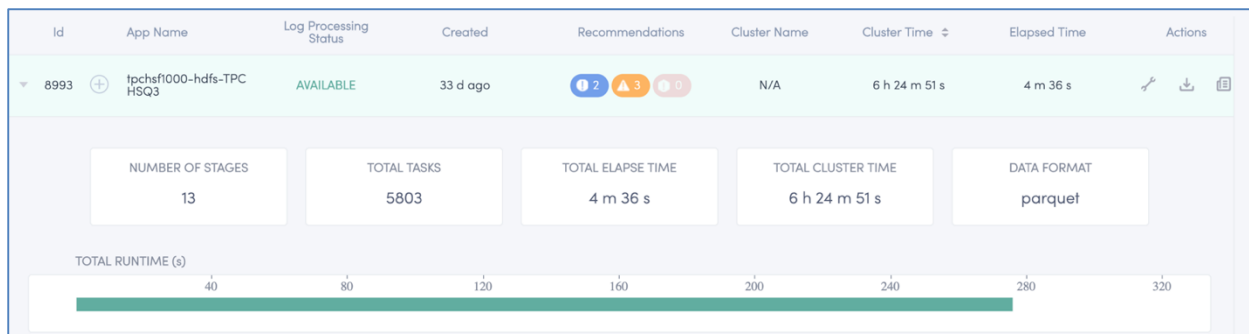


FIGURE 4. Application Overview

Figure 6 shows the result of clicking on a particular application report, in this case named “tpcdsf1000-hdfs-TPCHSQ3”. This overview gives information about the application in aggregate, including run time, cluster time used, data format and timeline. Again the blue, orange, and red issue buttons are clickable. They give access to analysis performed via log signature detection, along with recommendations for this particular application.

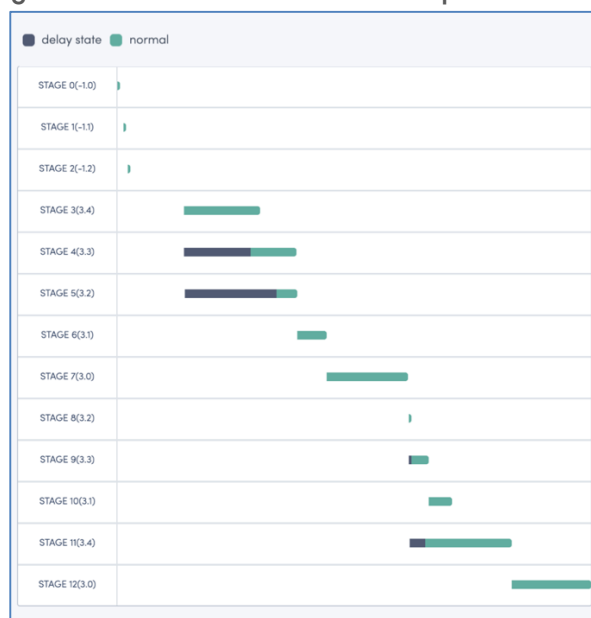


FIGURE 5. Stage-by-stage View

Clicking on the green timeline bar in Figure 4 leads the user to a stage-by-stage timeline of execution shown in Figure 5. The stage runtimes are shown, in order of execution. The dark parts of the individual stage timelines represent periods of time when the stage was runnable, but waiting for other inputs. Green portions show actual execution. In this example, the users eyes are drawn to stage 11, which appears to have the longest running time.

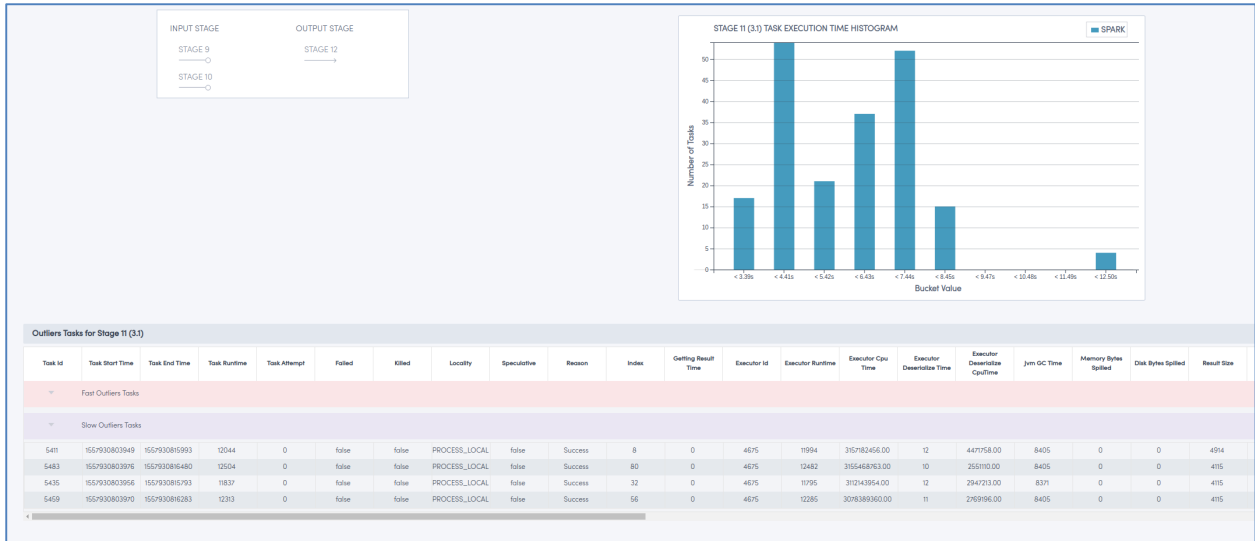


FIGURE 6. Stage 11 Analysis (part 1)

By clicking on the stage 11 timeline in Figure 5, the user has access to stage-by-stage information. The view is shown in Figure 6 and 7, where we have split up the view for clearer explanation. In this case, the user has identified stage 11 as the long running stage and is interested in the distribution of task runtimes. On the right of Figure 6, we see a histogram of task runtimes for the stage which in turn shows a group of outliers i.e. “straggler” tasks (the rightmost blue stack in the histogram).



FIGURE 7. Stage 11 Analysis (part 2)

Figure 7 is the other half of the stage analysis and provides information about the root cause of the outlier problem of stage 11. On the left is the Spark physical plan of stage 11 in isolation. This provides the user information about the operators that are causing the slowdown of the tasks. On the right, we see the specific straggler tasks (in red), their number and the length of those outliers. At this point the user has information to take action. This

could include approaches such as examining the application keying structure, memory allocation, employing Bigstream acceleration to shorten task times, or any subset of these approaches.

This example showed how with a few clicks, xRay can identify performance issues and provide knowledge and suggestions aimed at improvement. This example was a small subset of the analysis capability, the user is encouraged to visit the xRay site at insight.bigstream.co to explore further functionality including:

- Issue-by-issue recommendations for performance slowdown mitigation.
- Side-by-side, stage-by-stage comparison of multiple runs, useful for comparing like application runs.
- Acceleration potential estimates, based on operator usage and historical analysis

Future Releases

This We have introduced and shown examples of analyses that are provided by the xRay tool. The tool is usable in 3 easy steps and is runnable on the cloud or in the customer premise, even in a production environment.

In the future, we intend to expand the purview of the tool by adding functionality such as:

- Online, while running analytical capability.
- Additional optimization recommendations.
- Streaming application statistical analytics.
- Automatic “upload-less” report creation directly from the customer premise.
- Operating system level analytics..

In summary, xRay is an easy-to-use tool to analyze the performance of a huge spectrum of Spark applications. It can also point the way to utilizing seamless Bigstream Hyper-acceleration to address any user performance needs. We encourage the reader to get started with xRay today by signing up in minutes at insight.bigstream.co.



ACCELERATING INTELLIGENCE

Bigstream Solutions, Inc. 1975 W. El Camino Real, Ste 300, Mountain View, CA 94040

All rights reserved. All trademarks and registered trademarks are the property of their respective owners.