

# Challenges in Cluster Management

A Technical Whitepaper

Kobus Jooste

BixData

kobus{at}bixdata.com

This whitepaper focuses on the major challenges of managing a commodity cluster, and specifically on the features and benefits of using BixData on a cluster of 600 machines.

## About BixData

BixData is a software suite for managing diverse cluster environments and is specifically designed for environments that include a mix of operating systems as well as commodity and higher-end hardware. BixData is able to manage clusters of 1000, or more, machines running any combination of Linux or Windows Operating Systems.

## 1. Commodity Clusters

Commodity clusters are fast becoming the preferred way of running large scale applications and are composed of commodity hardware, usually with large attached storage. Many search engines, online web applications and scientific applications are running on commodity clusters.

The commoditization of cluster hardware and the drastic lower cost of hard disks have opened the market to build larger clusters at a lower cost and more rapidly. Apart from the drastically lower cost, commodity hardware is also more flexible and more open. However this also means there is less standardization on deployed hardware due to fast paced product changes, shorter product availability and a shorter product lifecycles.

Linux is the most popular choice of operating system for clusters. But with so many distributions and versions available, clusters often end up with a mix of different configurations and versions.

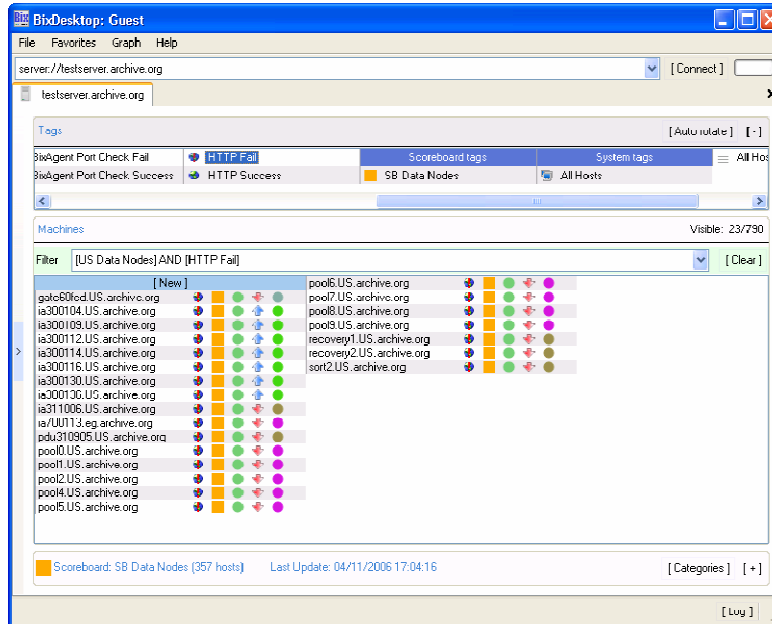
## 2. Challenges and Requirements

### 2.1 Machine registry

Keeping a list of machines and organizing them logically in some representation becomes difficult in an environment with so many machines, which is only made more difficult by machines being constantly added or decommissioned. Most clusters also have different classes of machines that include data nodes, head or controlling nodes and compute nodes.

BixData can import XML files as a source of dynamic machine lists and includes a network scanning tool for discovering new machines.

Through the BixDesktop Situation Room you can see a visual representation of all machines in the cluster. This allows you, among other things, to see if machines are up or down.



**Fig. 1** Situation room filtering hosts by selecting tags

Machines can be organized into smaller logical groups by using tags. Tags are similar to labels, and are represented by colors or icons. Each machine in the registry can be associated with one or more tags. This allows quick layout and organization of machines. Advanced filtering allows the user to combine tags and search text to identify and locate machines easily. Tags are also used to represent the different classes of machines. Therefore it becomes a simple task to see which of the compute nodes are currently down, or which of the data nodes are operating without any problems.

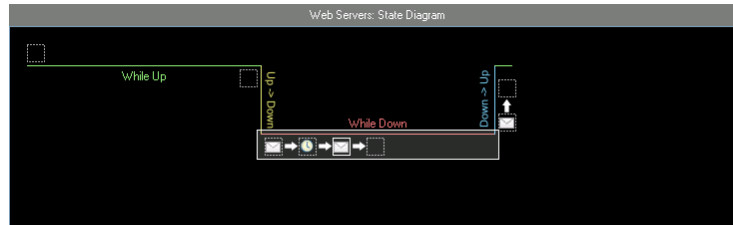
## 2.2 Service monitoring configuration

Each machine has a number of services that requires monitoring. These include network connectivity, HTTP, streaming servers, etc. An important requirement is maintaining the configuration to include new machines or exclude decommissioned machines.

All the services on the cluster can be monitored through Service Checks in BixServer. To enable easy configuration and maintenance, BixData allows the system administrator to configure Service Checks to run against tags. Since tags are dynamic and can be configured to include new machines or exclude decommissioned machines, the Service Check configuration is automatically maintained through the machine registry and

Situation Room. This prevents false Service Check failure notification and ensures that all machines are monitored.

BixData goes one step further by allowing Service Checks to be grouped together as different Notifications that allow different actions to be taken if any Service Check fails. This facilitates complex notification sequences and escalation procedures within an organization. Setup of Notifications and Service Checks are done through a graphical user interface by using a colored state diagram that displays the sequence of actions taken for each of the four states that a Service Check can be in.



**Fig. 2** State diagram showing HTTP notification configuration

### 2.3 Visualization of problems and error messages

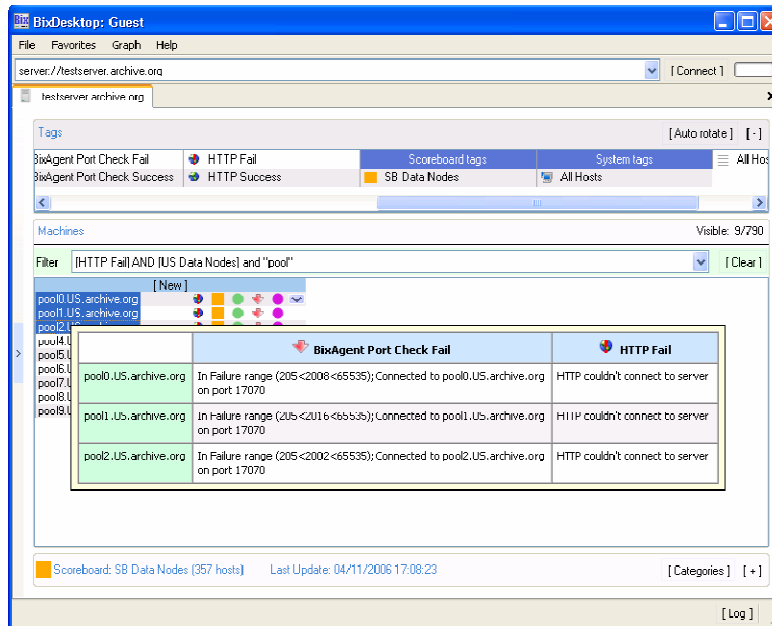
With many machines and services being checked, the cluster requires a visualization of problems and error messages that is simple and obvious.

Instead of presenting a list of machines sorted into a hierarchy, tags can organize machines into a multitude of smaller groups and classes, without presenting a complex tree to navigate. Tag colors and icons can visually show the association of machines with problem areas, or the logical organization, at a glance.

The Situation Room is designed not to present the details of errors and problems with the list of machines shown. This allows for a cleaner visualization and does not clutter the representation with many numbers or huge amounts of text.

A system administrator can browse the tags as well as combine and filter them in any way to drill down into problem areas. Tool tips are used to display more detailed information about a specific machine and its associated Service Checks. Hovering over a tag that represents a Service Check will display formatted error messages and Service Check detail.

Common situations system administrators face are not isolated problems, but segments of a network that goes down, or a set of machines that exhibit the same problem. Intelligent Tooltips provide the solution for comparing and analyzing a set of machines and error messages.



**Fig. 3** Intelligent tooltips comparing selected hosts

By selecting more than one machine, an Intelligent Tooltip will display formatted error messages and Service Check details in a table for easy comparison. The table is even pivoted vertically or horizontally to fit the most information on the screen, depending on the number of Service Checks associated or machines selected.

## 2.4 Storage of operational data

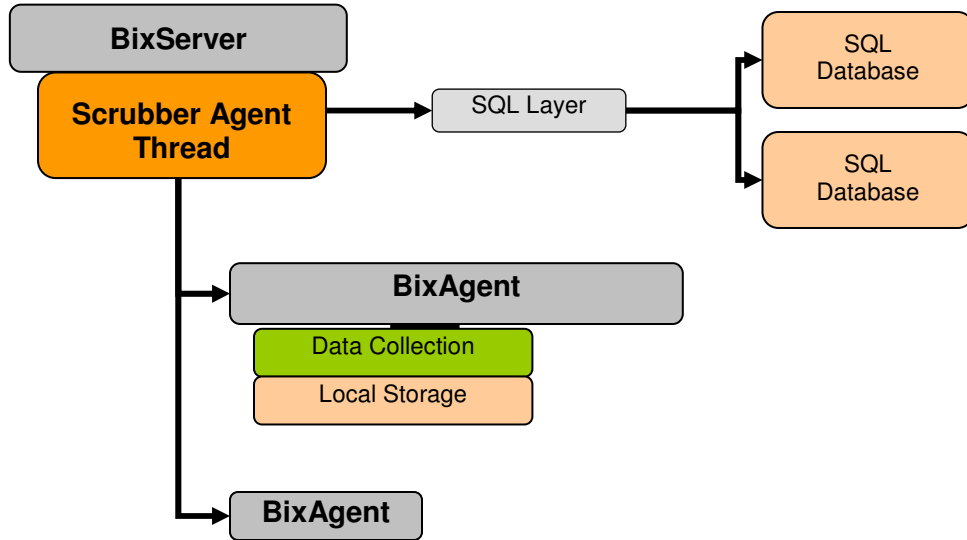
Many of our customers operate with a philosophy of storing and archiving information on the performance of their clusters and usually have archives of Operational Data about their cluster for a number of years. Data stored include CPU and Memory information as well as SMART hard disk information. Most of this information is stored in text files and comprises almost half a terabyte of information.

Analyzing the information stored in text files poses a number of challenges.

- 3 Text files do not present a data definition that can be used to verify that data stored at different times are the same format.
- 4 Text files sometimes do not store accurate timestamp information, other than the files own timestamp.
- 5 Text files contain no additional information about the data source, or method of data collection, such as version or data collection settings.
- 6 Text files need to be parsed and stored in a database or spreadsheet to be useful for further processing into reports or graphs or study.

BixData addresses the challenge of collecting, storing, and analyzing Operational Data through its Operational Data Store and Advanced Reporting.

The first part of the challenge is to reliably collect and store the information, without impacting running systems in terms of system performance or network bandwidth.



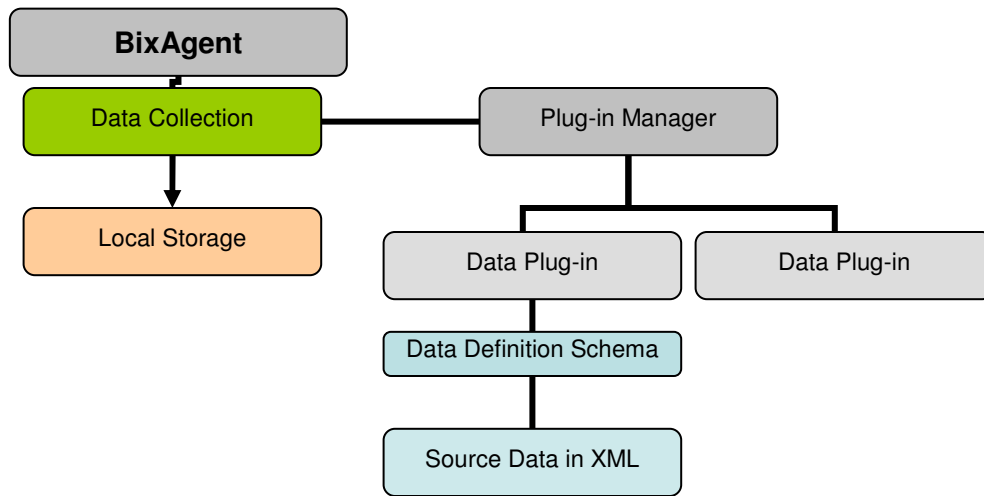
**Fig. 4** BixServer data collection from agents into SQL storage

BixServer contains an Operational Data Store that is built to support storage of data in any number of SQL databases. BixServer can be setup to connect to a number of SQL database servers and use them as storage for collected data, reports, and graphs. To add more storage or remove storage from the Operational Data Store is as simple as adding or removing a SQL database server connection.

BixData employs an agent that natively runs on each machine on the cluster to reliably collect and store data for storage in the Operational Data Store. Collecting data from 1000 or more machines means that at any time any number of machines may be offline due to network, hardware, or software failure.

The most reliable strategy is to collect data locally on the machine, independent of a network connection and store data locally until a central component, the BixServer, collects the data for reliable central storage.

Local data collection through an agent also enables more detailed information to be collected because the agent has local access. Reliable timestamp information and synchronized timestamps across all machines is also a critical requirement, if any reasonable comparisons are to be made between machines in the cluster.



**Fig. 5** BixAgent data plug-in architecture

The BixAgent consists of different plug-ins to collect data from hardware or the Operating System. Each plug-in exposes a data definition in XML with relevant version and schema information. This allows BixAgent to be flexible and handle different versions of Operating Systems and environments. Each set of data produced by BixAgent is sufficiently detailed to contain information about the data source, the data types as well as version information. Once data is collected by BixServer and stored in the Operational Data Store, this extra detailed information allows BixServer to make meaningful and accurate comparisons between running machines in the cluster.

Data collection by the BixServer is an automated and low impact operation that compresses and optionally encrypts data from BixAgents. Storage in a SQL store is automatically done by BixServer which can create tables and data storage schemas on the fly, depending on the data collected.

The following table provides some statistics on the performance of Data Collection and storage. An entire data collection and storage operation for 412 hosts with 554 data points per host, takes approximately 2 min 29 seconds.

Hosts	Time	Total Size	Total SQL Records	Size and Records
412	149 seconds 360 ms/host	16,898 KB 113 KB/s	228,341 1532 records/s	41 KB/host 554 records/host

The following table provides some statistics on the performance of the Reporting and Graphing capabilities that use the Data Collection and Operational Data Store.

Hosts	Time	Report or Graph
412	9 seconds	Hard Disk inventory report that includes manufacturer, model, version, revision and serial number for 1646 hard disks
412	7 seconds	Report that shows CPU utilization, idle time, load averages, memory and swap file information for each host
412	5 seconds	Graph showing the load average for all hosts

## 2.5 Hard disk failure study

One of our customers is studying current and historical SMART data, to determine a method of predicting hard disk failure more reliably.

BixData provides a benefit in studying current and future Operational Data through advanced reporting and graphing.

Reports and graphs can be created by BixServer through XML queries to the Operational Data Store.

```

<TableQuery>
  <namespace>Common_SmartInfo</namespace>
  <version>1</version>
  <schema>ATA</schema>
  <instance>*</instance>
  <key>*</key>
  <field>model</field>
  <field>serial</field>
</TableQuery>

```

**Fig. 6** XML query to retrieve hard disk inventory from stored SMART data

Data output by BixServer is in XML format and can be formatted into HTML or other data formats by using XSL style sheets or into graphs by using XML graph templates.

```

<xsl:template match="TableData">
  <xsl:for-each select="Instance">
    <tr class="key_{position() mod 2}">
      <td>
        <xsl:value-of select="../../../key" />
      </td>
      <td>

```

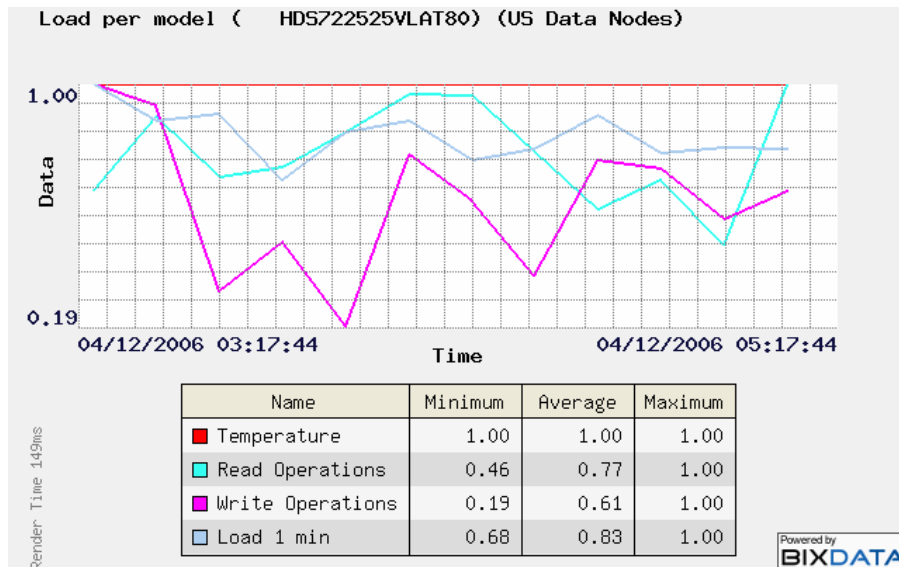
```

                <xsl:value-of select="value" />
            </td>
            <xsl:for-each select="Key[not(starts-with(value, 'cpu'))]/Fields/*">
                <td>
                    <xsl:value-of select="node()" />
                </td>
            </xsl:for-each>
        </tr>
    </xsl:for-each>
</xsl:template>

```

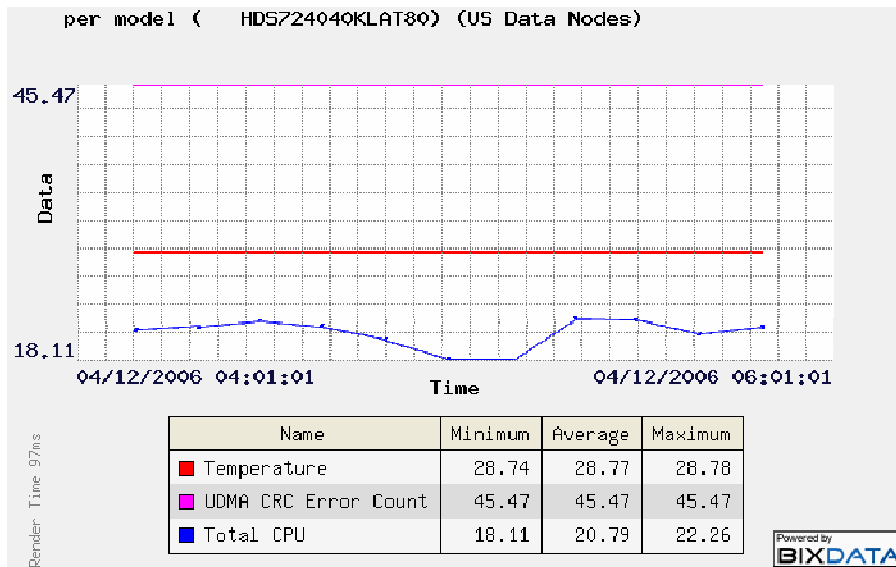
**Fig. 7** XSL Style sheet for outputting data in HTML format

BixServer provides near real-time data collection and graphs for SMART hard disk data. The advanced graphing capabilities allows the study of Hard Disk temperatures and error rates correlated to machine CPU usage, machine load or read and write activity.



**Fig. 8** Hard disk temperature normalized against read/write operations

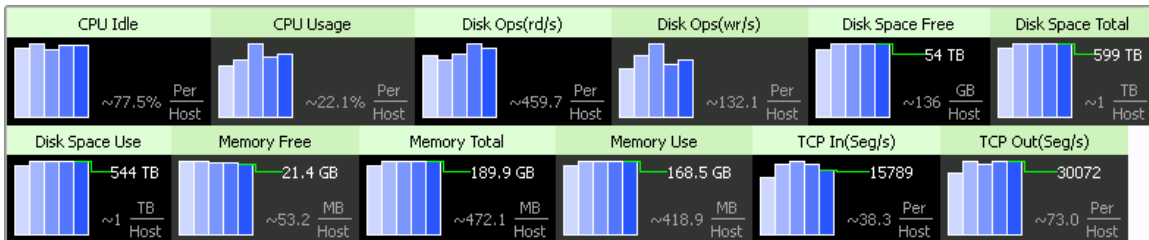
BixServer advanced graphing capabilities include the ability to pivot graphs and reports based on a specified dimension in the Operational Data Store. This allows the study of hard disk SMART information such as temperatures and error rates for a specific manufacturer or model of hard disk. Or to graph disk activity for installed logical disks that allows you to study the activity of a boot partition versus a purely storage partition.



**Fig. 9** Hard Disk temperature, CPU usage and CRC error rate, per model

## 2.6 Large screen display

The Situation Room supports a demo mode that automatically rotates through the machines in the cluster and shows detailed information about the cluster environment. This is suitable for large screen displays to show information to technical staff and visitors. Scoreboards can show cluster wide critical information such as total processing power and total storage usage for the cluster and are also automatically updated. Each graph shows the last five values for total and average per host for each of the information categories.



**Fig. 10** Scoreboard showing total & average critical values

### **3. BixData Technical Advances**

BixData is designed as large scale distributed software with the focus on being flexible and low impact, as well as easy to deploy.

BixData achieves this through a number of technical advances.

#### **3.1 Cross platform, no dependency**

All BixData components are built using our own cross platform runtime. This allows BixData to run natively on Linux and Windows, with more planned Operating System support for the near future. Even when running exclusively on Linux, there are a number of challenges related to various levels of Operating and System software, such as kernel versions and varying levels of libc compatibility. BixAgent is compiled and linked statically for different operating systems, with no external dependencies, to maximize the ease of deployment.

#### **3.2 Different kernel versions, proc file systems and security privileges**

BixAgent solves challenges due to different kernel versions and proc file systems by employing a plug-in system, and including different plug-ins that support a multitude of Operating Systems and versions. BixAgent requires no special security privileges, and can access all monitoring values in read only mode. SMART information can even be gathered by parsing output from a command line utility instead of through the supported native plug-in.

#### **3.3 Data schemas and data storage**

Open data exchange is very important to integrate with different environments and software systems, especially in IT. All data schemas and data storage in BixData is in XML format that can easily be integrated into 3<sup>rd</sup> party applications or accessed directly from the underlying SQL database servers.

#### **3.4 Low impact TCP/IP communication and Binary XML**

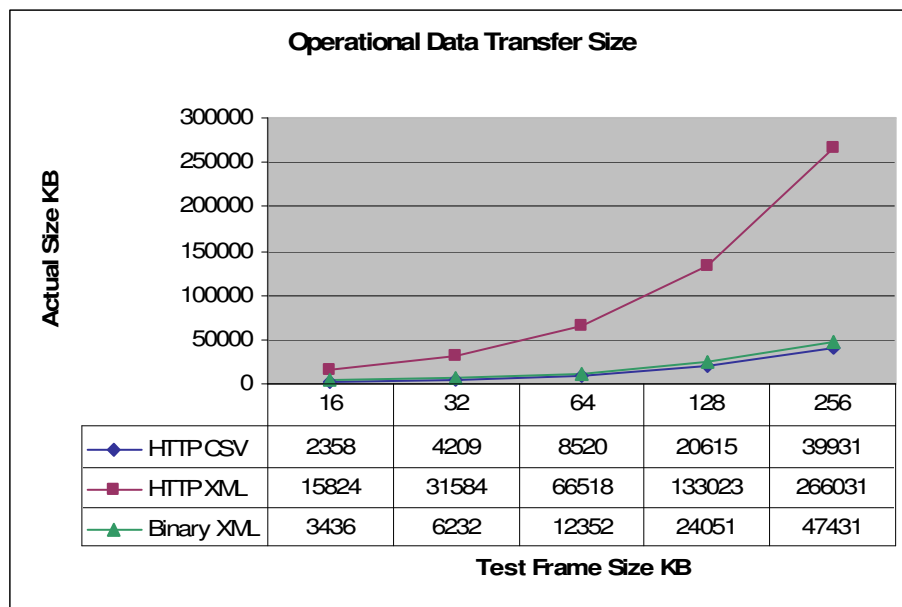
The BixData cross platform runtime includes its own low impact TCP/IP communications library built for performance. BixServer running on a low end Pentium 4 desktop computer can easily balance more than 1000 open connections.

All data exchanged through communications uses an indexed form of Binary XML. This allows the most flexibility for exchanging typed data, as well as high information density when compared with XML.

A number of tests has shown the lower transfer times and higher transfer rates achieved through using Binary XML for data exchange when compared to other forms of semi structured text data or XML.

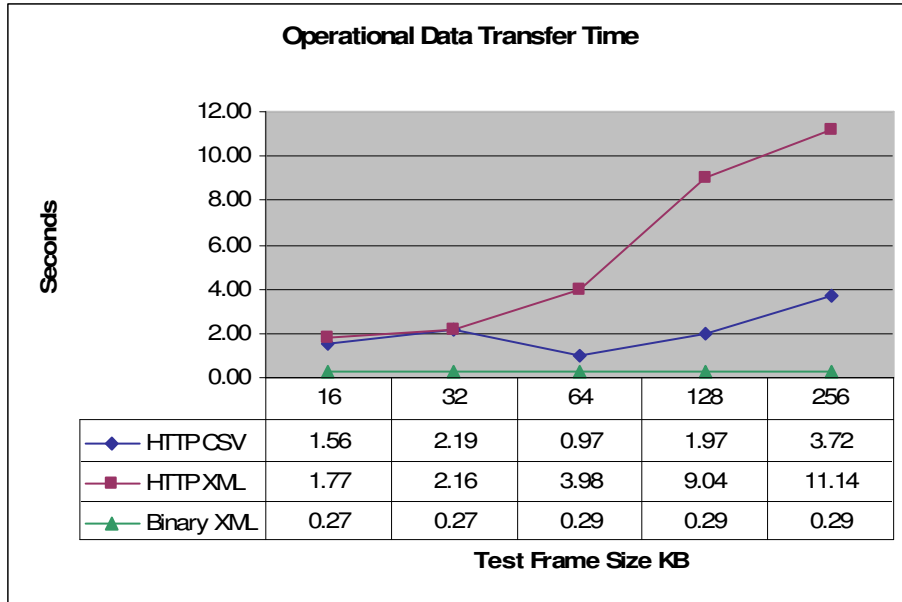
Test operational data was gathered in different size of 16, 32, 64, 128 and 256 KB. Each set of test data was converted into compressed CSV (comma separated values) format and XML format. The data in XML format was directly converted into Binary XML.

Fig. 11 shows that, on average, Binary XML was 5 times smaller than XML. Most operational data includes a lot of numeric data and storing data values in their original numeric format achieves a 2 or 3 fold reduction in size. CSV is slightly smaller than Binary XML, but includes no type information or data structure.



**Fig. 11** Transfer size of HTTP CSV vs. HTTP XML vs. Binary XML

Many system management applications and script based data collection, such as Perl, uses HTTP for transferring data from a source. Tests were done to compare transfer of data over HTTP using Apache versus Binary XML through a dedicated TCP communications library.



**Fig. 12** Transfer time of HTTP CSV vs. HTTP XML vs. Binary XML

Fig. 12 shows that Binary XML benefits from the size reduction, but is still a lot faster than CSV over HTTP. Using a dedicated TCP communications library is a lot more efficient than transferring data over HTTP. Transfer times for Binary XML includes making a connection, generating and parsing the Binary XML, where transfer times for XML over HTTP was a simple static transfer of a file.

### 3.5 Interactive graph and real-time monitoring

BixDesktop can connect directly to any BixAgent and receive real-time data. Any number of data points from data plug-ins and data sources within BixAgent can be displayed using the interactive graph.

## 4. Conclusions and Future Work

Cluster management continues to present many challenges as clusters scale out and provide more storage and computational power. BixData is focused on providing solutions for managing diverse environments, simplifying system management and developing innovative cross-platform software.