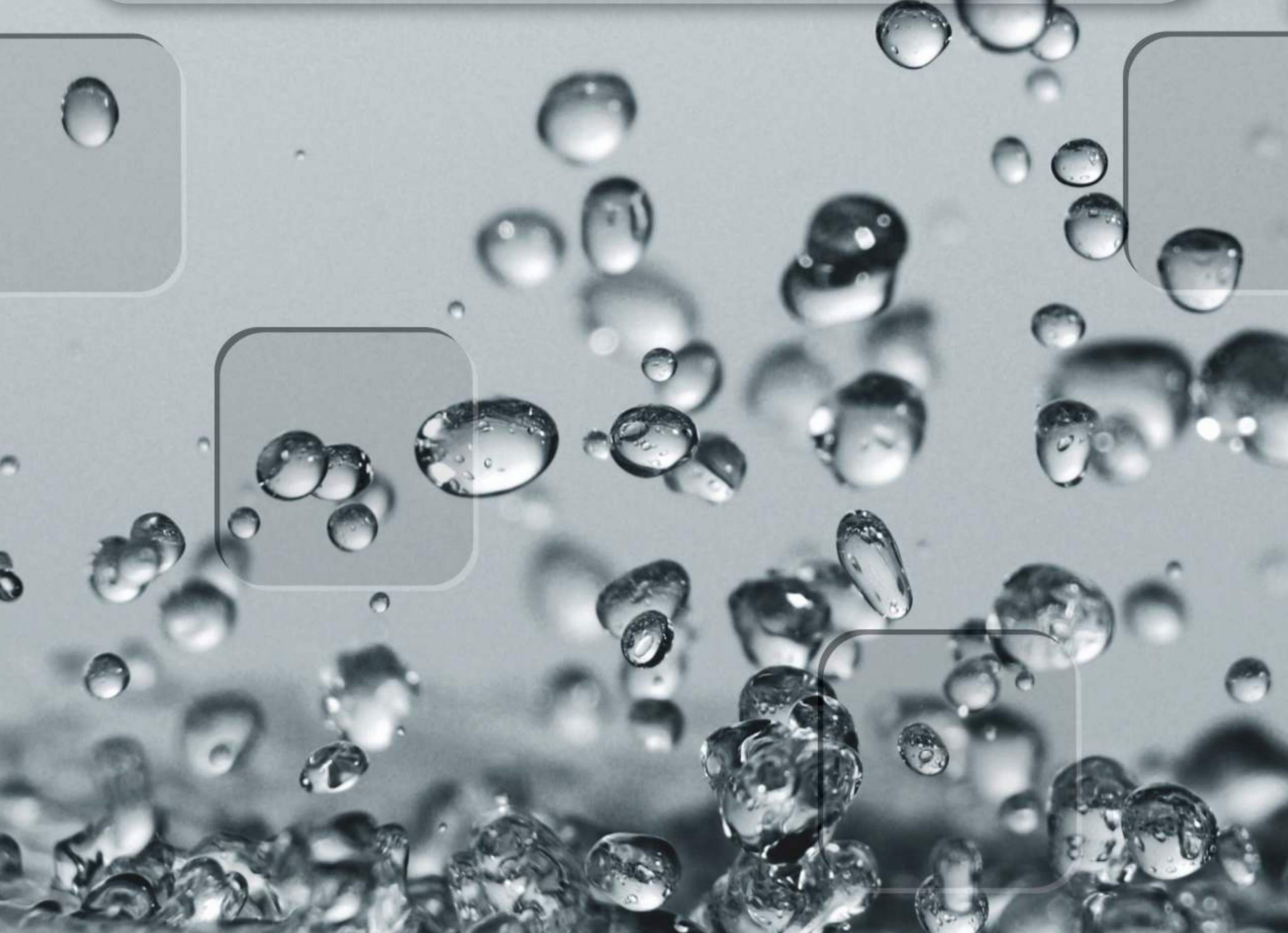
Actian ParAccel Dataflow for Hadoop Cluster Configuration Guide



Introduction

Sizing a Hadoop cluster is important, as the right resources will allow you to optimize the environment for your purpose, saving huge amounts of time, as well as hardware and energy costs, over the lifespan of the cluster. At Actian, helping businesses take action on big data is our specialty. In this paper, we guide you through the many steps to build a Hadoop cluster to serve your specific needs.

About the Actian ParAccel Big Data Analytics Platform

Once you have chosen the right hardware for your environment, you must also choose the right software. We at Actian believe that our ParAccel Big Data Analytics Platform is the best software in the market for high-speed, high-demand analytics work. In this paper, we will provide you with the facts to support these claims.

The ParAccel Big Data Analytics Platform consists of three main components: ParAccel MPPdb, ParAccel SMPdb and ParAccel Dataflow, along with various on-demand integration utilities and other elements that support them.

ParAccel SMP provides high-speed, low-latency interactive SQL style in-database analytics on a single, rapid-deploy-and-go server. ParAccel MPP provides similar high-speed, low-latency interactive SQL style in-database analytics in a large scale distributed database. These two options give you unbeaten analytic database performance, regardless of the scale or complexity of your workload.

The third component is ParAccel Dataflow. Use this option when you need high-speed, high-throughput, advanced analytics, including machine learning, on Hadoop clusters, or when you need integration and data quality cleansing between Hadoop and other data sources, including ParAccel SMP or MPP.

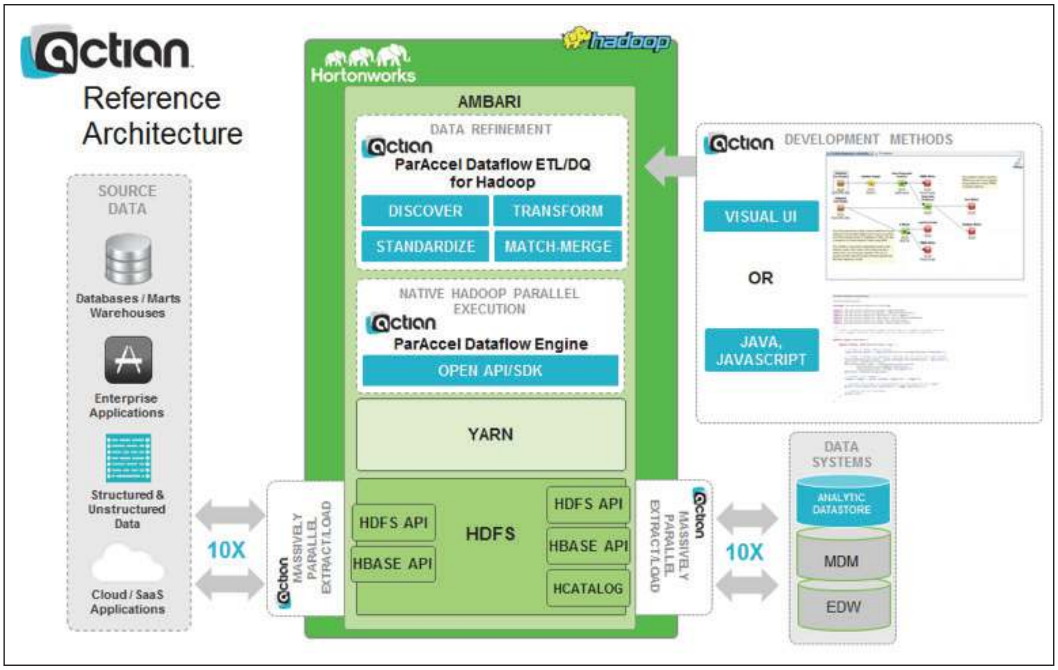
About ParAccel Dataflow

ParAccel Dataflow is a unique, patented parallel programming framework and execution engine that eliminates performance bottlenecks in data-intensive applications. The framework automatically parallelizes and optimizes workflows at runtime using directed acyclic graphs connected by dataflows, to take advantage of any hardware configuration. ParAccel Dataflow has five main advantages for cluster application development.

1. **Automatic Scaling:** Performance of Dataflow applications dynamically scales up with increased core counts and increased nodes. No need to constantly re-build the same applications as hardware needs change.
2. **High Throughput:** Using both pipeline and horizontal data parallelism, ParAccel Dataflow consistently outperforms alternatives by factors of up to 5X with no limits to input data size.
3. **Cost Efficient:** Auto-scaling at runtime means that Dataflow squeezes maximum performance from standard multicore servers, SMP systems, and multi-node clusters, cutting energy and other usage costs by factors as high as 100X and beyond. This means that the same level of performance can be gained from a much smaller cluster, or that larger clusters can handle far more extensive workloads.
4. **Easy to Implement:** ParAccel Dataflow provides a visual interface, as well as a Javascript and Java API for more in-depth development. Complex parallel programming problems, such as threading, collisions, etc. are handled by the framework, vastly reducing development time.
5. **Extensible:** The existing list of ParAccel Dataflow functions and capabilities is quite extensive, but we recognize that there will always be a need for new functionality or refinements of existing functionality. The API for user development is rich and friendly. Knock yourself out.

Actian offers two packaged applications for working on Hadoop clusters with the ParAccel Dataflow Engine: ParAccel Dataflow ETL/DQ for Hadoop and ParAccel Dataflow Analytics for Hadoop. Both include a visual working environment and a wide array of ready-to-use data access, data quality, de-duplication, and data preparation operators, as well as basic data profiling and statistics operators. ParAccel Dataflow Analytics for Hadoop also includes more advanced distributed analytics, and machine learning capabilities.

We'll focus specifically on the ParAccel Dataflow ETL/DQ for Hadoop package in this paper, as it provides the most widely useful capabilities for any large scale data preparation job on Hadoop. Regardless of what other applications, databases or architectures you use, high-speed, parallel data profiling, cleansing, de-duplication, transformation, filtering, merging, sorting, etc. are always going to be essential parts of the process.



- 100% visual development with no knowledge of MapReduce or parallel programming required
- Offers up to a 500% improvement in performance execution over MapReduce
- Open Java API and SDK for custom parallel functions

About YARN

Historically, MapReduce applications were the only applications on a Hadoop cluster that could be managed by the Hadoop resource management capabilities. In the new version of Hadoop, YARN (Yet Another Resource Negotiator) provides containers for all YARN-compliant applications launching on worker nodes. This helps control usage of CPU, memory, disk, and network resources. YARN provides resource negotiation and management for the entire Hadoop cluster, and all applications running on it.

This means that, once integrated with YARN, non-MapReduce-based distributed applications, such as ParAccel Dataflow, can run as first-class citizens on Hadoop clusters, sharing resources side-by-side with MapReduce-based applications. ParAccel Dataflow is fully YARN compliant and certified. This means that the Dataflow benefits of more rapid development and deployment, as well as automatic scaling and superior execution performance, can be enjoyed without the need for any extra resource management effort.

The Basics

Hadoop and HBase clusters have two types of machines:

- **Masters –** the HDFS NameNode, the YARN ResourceManager, and the HBase Master
- **Workers –** the HDFS DataNodes, the YARN NodeManagers, and the HBase RegionServers

The DataNodes, NodeManagers, and HBase RegionServers are co-located or co-deployed for optimal data locality. In addition, HBase requires the use of a separate component (Zookeeper) to manage the HBase cluster.

Actian recommends separating master and worker nodes for the following reasons:

- Task workloads on the worker nodes should be isolated from the masters.
- Worker nodes are frequently decommissioned for maintenance.

Clusters of 3 or more machines typically use a dedicated NameNode/ResourceManager and all the other nodes act as the worker nodes. For a medium (45-100 node) to large (300+) cluster, we recommend expanding this to **4 master nodes** as follows:

1. NameNode
2. ResourceManager Node
3. Secondary NameNode, HBase Master, Hive Server
   *NOTE – In a half-rack cluster, this would be combined with the ResourceManager.
4. Management Node (Ambari, Ganglia, Nagios)
   *NOTE – In a half-rack cluster, this would be combined with the NameNode.

That leaves the remaining hosts for Worker nodes, each running a DataNode, NodeManager, and HBase Region Server. As an option, clients may be installed on these nodes to facilitate data and processing integration between the Hadoop cluster and legacy systems within the environment. This can be seen in the Assign Workers and Clients step of the Ambari Cluster Install Wizard.

Balanced Value and Optimum Price/Performance

A Hadoop cluster is different from common enterprise IT server infrastructure. First, a workload or job within a Hadoop cluster will have little or no idle time during its life. Second, Hadoop requires a “balanced value” which guides you to select hardware configurations for master and worker nodes where consumption of CPU, memory and disk resources all peak together. The result of a balanced value design is optimum price/performance of your Hadoop cluster.

Hadoop workloads are different from common IT applications

The Hadoop workload consumes data, creates new data sets and is typically limited by CPU, memory and disk bandwidth. If you compare this to a typical IT workload you find the Hadoop workload spends very little time in an idle state, and wastes relatively few resources. Hence, a Hadoop workload is typically a single execution task. Running multiple Hadoop workloads (in version 1.0) on a common set of hardware often results in saturation of the CPU, memory and/or disk I/O resources. The end result is that overall job loads lose efficiency due to bottlenecks. We recommend running master and worker nodes on physical hardware for the best efficiency and price/performance optimization.

A “shared nothing” architecture promotes scalability

In Hadoop cluster design, the “shared nothing” concept says that a master or worker node should not share any hardware resource with any other node, which highlights the performance issues that will result of using SANs for disk storage and unified I/O fabrics found in blade servers. Share resource infrastructure assumes that the average of the demand from the jobs will not exceed the total bandwidth of the resource. The performance issues come from the non-interactive job profile of Hadoop workloads. Since all Hadoop jobs are demanding max performance, the shared resources will become saturated and result in a performance bottleneck.

The benefit of a “shared nothing” architecture is particularly clear when nodes are added to a cluster. Since each node is not sharing any resource with other nodes, the addition of a new node results in added CPU, memory and disk I/O resources. When we couple this with the ability of a Hadoop cluster to break a larger job into multiple independent jobs, the result is a near linear scalability of the cluster. If you need to double the capacity of the cluster you can easily double the number of nodes. ParAccel MPP and ParAccel Dataflow also have a shared nothing architecture with near linear scalability that expands readily as your processing needs expand, making this software ideal for inclusion in well-designed clusters.

Balance: the key to high performance & affordable Hadoop Clusters

When building Hadoop clusters, make sure all components are balanced in capacity, cost and performance. With a balanced cluster, we minimize performance bottlenecks and avoid excess unused capacity. Since there are many more worker nodes than master nodes, most of our discussion will revolve around worker node design considerations.

Common Cluster Component Guidelines

If we look at a node design for master and worker nodes, we see that we have control over selection of CPU, memory and disk I/O. We have secondary considerations of power, cooling and physical space. As the size of the cluster scales up, it acts as a multiplier, and these secondary considerations can very quickly make our design impractical to implement.

- **CPU and memory selection**
  Always start with CPU selection. Here we need to look at price/performance of our CPU choices. Picking the current top of the line CPUs will result in a very expensive cluster as you multiple out this cost by the number of nodes. Going the other direction picking underpowered CPUs results in your cluster not being able to meet its performance goals. Since there is strong innovation happening in the CPU market, the sweet spot will move quarter to quarter. Right now we find the Intel Sandybridge hex and octo core processors to be good value for master and worker nodes respectively. We expect that the hex core processors will become more competitive. Always check price/performance ratio when doing your design.

- **Disk selection**
  The amount of disk on each worker node will define the amount of data your cluster can hold. The speed of the disks will be one of the bottlenecks that will constrain the worker node performance. So our disk selection is going to be a trade-off between performance and capacity. The highest capacity disks use a SATA interface. SAS drives have a more sophisticated command set which will give you better performance and reliability, but SAS drives cost more. We have found that near line SAS drives give us a good combination of both capacity and performance. A near line SAS drive is the marriage of a SAS interface to a SATA disk mechanism. So we get the capacity of SATA with some of the performance of SAS. We do lose on reliability and life span but since Hadoop is designed to accept drive failures this is an acceptable trade-off. This kind of design decision is typical of what you will do as you iterate through your cluster design.

› **Networking**

Networking for Hadoop clusters is simple since most of the data traffic is taking place inside the worker nodes. The network is used for control, bringing in new data and moving out the results of jobs. For each rack of servers there will be a 1G rack level switch with a 10G uplink. That connects to the nodes in the rack. For small to medium clusters we can daisy chain these switches together. When you go over 5-7 racks you will then need 10 GB spine switches to interconnect the racks.

Our trade-off decision in this case is 1 GB Ethernet versus 10 GB Ethernet. This decision will be made on cost and infrastructure considerations. While the incremental cost of 10 GB Ethernet interfaces may be reasonable, you must ask yourself whether the interconnect infrastructure can handle it.

Later, we will talk about cluster isolation, and if we have isolated the cluster, this may be a good decision. From a balanced value point of view, right now 1 GB Ethernet between nodes in a rack with a 10 GB Ethernet backbone will meet most current designs.

› **Redundancy in cluster components**

In the design of an Apache Hadoop cluster we apply redundancy sparingly. From a cluster point of view the built-in redundancy of Hadoop software allows a cluster to lose a drive, a worker node or, if the scale is large enough, a whole rack of servers. So as your cluster grows, the loss of any component diminishes in impact. This is part of the beauty of the Hadoop design.

The important place to apply hardware redundancy, (i.e. extra power supplies, multiple network paths, etc.) is with the master nodes. The loss of a master node will affect the whole cluster, so employ enterprise-grade server hardware to minimize the chance of losing a master node.

The exact opposite happens with the worker nodes. Since data is replicated through the cluster, the loss of a disk drive or a worker node is managed by Apache Hadoop. In a large enough cluster replacement of failed hardware is a scheduled maintenance task, and Apache Hadoop manages replacement of hardware so there is no cluster downtime. It is important to realize that worker hardware failure is not a priority-one issue. It can be resolved the next time there is a hardware replacement maintenance event.

Network redundancy changes as you go from small to large clusters. In a small and medium cluster, loss of a switch has a larger impact than in a large cluster. We do not recommend redundant switches but we do recommend redundant paths so you can minimize the impact of a switch failure. Consider physically splitting the cluster into two equal halves, so if one cluster has a failure, the other cluster can still process jobs.

Master Node Design Guidelines

The master nodes are key components of your Hadoop cluster. Since there are only a few of them, we can relax some of our cost constraints. We want to optimize these nodes for availability. We recommend using enterprise-quality hardware since a master node failure will impact the cluster. Redundant power supplies and network connections will help increase the availability of the node.

- **Memory**
  Memory demand for a master node is based on the NameNode data structures that grow with the storage capacity of your cluster. We find that 1 GB per petabyte of storage is a good guideline for master node memory. You then need to add on your OS overhead, etc. With Intel Sandybridge processors, 32 GB is more than enough memory for a master node.

- **Local disk configuration**
  The local disks on a master node should be configured using RAID 10 with hot spares. This configuration is picked to give good performance and fast recovery on loss of a disk. We recommend 6 x 1 TB drives. Four are allocated to the RAID 10 with two hot spares.

- **NameNode guideline**
  An important resource for NameNodes is memory to hold working data structures. For NameNode memory, you need 1 GB of RAM for every petabyte of allocated storage space.

Worker node design guidelines

The design of worker nodes is dominated by the fact that there are very many of them. Anything you do - good or bad - is multiplied by the number of nodes. Look to balance the CPU, memory and disk I/O configuration such that they all reach their peak limits around the same point.

- **Memory**
  If we follow the Intel Sandybridge processor selection we find that a ratio of 4 GB of RAM per physical core is a good starting point for RAM. When using the Dataflow engine, memory can be used very heavily during intra-node processing. Remember that memory is cheap but when you multiply the cost of the number of worker nodes, the extra money spent on RAM could allow you to buy a few more worker nodes.

- **Disk**
  Disk selection is limited by the I/O bandwidth of the processor chip set and the disk controller card. Based on our Intel Sandybridge recommendation we find that 6 x 1 TB disks configured as JBOD will do well. A JBOD configuration keeps the disk controller out of the way and allows for full speed I/O with the processor.

- **Disk Controller**
  Having a disk controller or controllers that can provide enough bandwidth to all the drives in the system is key. This will ensure that the connection between the CPU and the disks does not become a bottleneck.

- **Core to Disk Ratio**
  The more drives a server contains, the more efficiently it can service I/O requests because it reduces the likelihood of multiple threads contending for the same drive, which can result in interleaved I/O and degraded performance.

- **Memory configuration**
  Servers running the worker node processes should have sufficient memory for either Dataflow, HBase and/or for the amount of MapReduce Slots configured on the server. The Intel Xeon E5-2470 has 3 memory channels per processor. When configuring memory, always attempt to populate all the memory channels available to ensure optimum performance.

Cluster Design Tradeoffs

We classify clusters as small (up to 3 racks), medium (4-10 racks) and large (above 10 racks). What we have been covering so far are design guidelines, and part of the design process is to understand how to bend the design guidelines to meet your goals.

In the case of small, medium and large clusters things get progressively more stringent and sensitive when you bend the guidelines. A smaller number of worker nodes allows you greater flexibility in your decisions. One of the advantages of using ParAccel Dataflow, rather than MapReduce, is that you can get far more performance out of less hardware. This allows you to keep cluster size small, and still get the power you need.

There are a few guidelines you don't want to violate, though, like isolation, regardless of the size of your cluster. When you get to a medium size cluster the number of nodes will increase your design sensitivity. You also now have enough hardware that the physical plant issues of cooling and power become more important. ParAccel Dataflow really shines in larger clusters where efficient hardware usage can vastly improve overall cluster power usage. Your interconnects also become more important. Hortonworks' experience has allowed them to develop deep expertise in this area and we at Actian strongly recommend you work with them if you want to build Internet-scale clusters.

Growing Your Cluster

Once you build your initial successful cluster, demand to increase your cluster size will quickly grow.

If your cluster is larger than a couple of racks, we recommend adding to your cluster in increments of racks of nodes. This keeps things simpler since you just need to replicate racks. You will probably be expanding in your existing data center so try to keep the new racks near each other. You will need to directly connect the new racks to the cluster.

As you grow from a small to medium cluster, watch the bandwidth of the rack switches. At this point the racks are daisy chained together. When you exceed 5-7 racks you may need to introduce a spine switch that interconnects the rack switches. Growing the cluster one rack at a time is a good strategy because it will let you spot issues as they emerge versus attempting to pinpoint issues after a large change in the cluster. Otherwise, as you reach the 10-14 rack range, the cluster should grow nicely.

When you grow from medium to large clusters, your physical infrastructure - space, power, cooling, etc. - will be the driving functions. Just as server best practices dictate - test as you build. Integrate and test each rack or row. Make sure you are adding a solid and stable resource to your cluster. If you are planning a major expansion, please engage your Hortonworks or Actian team as we have the knowledge and tools to help make a smooth transition.

One of the biggest advantages of ParAccel Dataflow is that it will easily grow with your Hadoop cluster. Dataflow workflows can be built via a graphical user interface on a client machine (laptop, desktop, etc.), tested on a small cluster or pseudo-distributed environment, then moved to production with no change to the design. The ParAccel Dataflow Engine automatically detects hardware resource availability and optimizes the workload at runtime, so, unlike with MapReduce, there is no need to constantly re-write the same applications as the cluster grows.

Isolation

A Hadoop cluster is an integrated supercomputer cluster. As such, issues like network interconnect latency are important. It is strongly recommended that your Hadoop cluster be implemented on an isolated network with a minimal amount of network complexity. When Hadoop cluster servers are scattered through the data center, you may experience network instability throughout the whole data center. Isolate and localize the Hadoop servers from the rest of the data center.

Heterogeneous Worker Nodes

Your first cluster build should be homogenous - all the worker nodes should be the same or quite similar. Over time, technology changes, and you will likely buy different node configurations as you go along.

When you have a mix of faster and slower nodes, Apache Hadoop does make some adjustments in the job stream. One feature of Apache Hadoop is speculative execution, which means a node is behind in processing its share of a job. The resource manager will start another copy of that task on another node just in case the first one fails. It will then take the results of whichever node finishes first. This works out fine at the job level.

At the cluster level, this results in some inefficiency because you are running a task twice. When there is a large performance differences in nodes, problems might result. Some things to watch for are things like SATA I/SATA II disks, 1 GB/10 GB Ethernet, difference in spindle count, ext3/xfs file systems, etc. These things make a difference in node performance. Your cluster can develop a hot spot where a server or set of servers become a bottleneck in getting jobs done. In general, keep worker nodes balanced so that any node can run a task equally well.

The ParAccel Dataflow Engine has an exceptional ability to conform to a heterogeneous cluster. Since it detects the available resources at runtime, it can adapt workflow optimization more efficiently than MapReduce. This allows Dataflow to appropriately take advantage of the specific hardware of any cluster configuration without creating bottlenecks at weaker machines.

Small Hadoop Cluster Design Example

A small Hadoop cluster could consist of a two-level architecture built with rack-mounted servers housed in one or two racks. With a high density of disks (6-12 per node) on the worker nodes, Hortonworks does recommend two racks to start in order to best support the load, power and cooling specifications of most data centers as well as the expansion that most customers undertake after their initial Hadoop implementations. These constraints / plans should be reviewed before choosing a single rack. A single rack of servers should be interconnected using a 1 Gigabit Ethernet (GbE) switch. For purposes of this cluster planning document, we are assuming a single rack is 42U.

Recommended hardware specifications for a half-rack cluster are as follows:

Rack 1: Small Cluster	
Master Nodes **2 Servers**	2 - Intel Quad core Sandybridge processors 8 - 8 GB DDR3 rDIMMs (Sparing, ECC) 1 – Perc S100 (RHEL6/xfs – RAID 10) 6 – 1 TB, NL-SAS disks Built-in dual GbE Dual Power Supply
Worker Node **8 x 2U Servers**	Intel Octo core Sandybridge processor 16 - 4 GB DDR3 DIMMs 2 – Internal 2.5" SATA drives for root (RHEL6/xfs) 12 – 1 TB, SATA disks (SATA/ahci) Built-in dual GbE
Rack-level Switch: 1U	24 – ports GbE

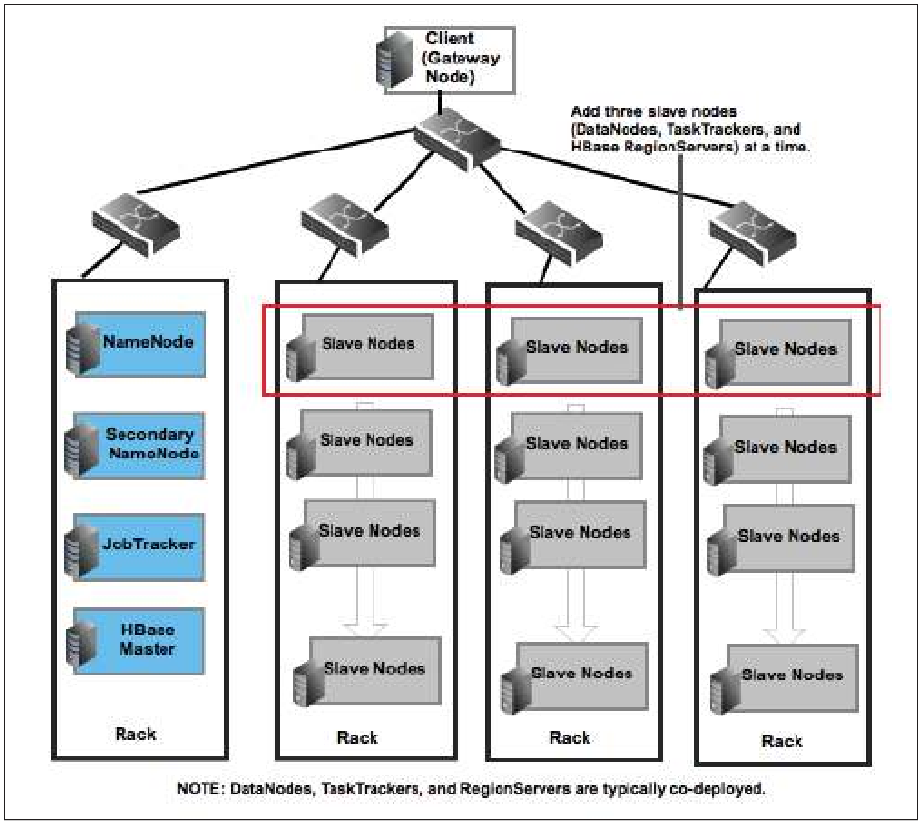
This half-rack cluster is sized for 96 TB of Raw Storage. To calculate how much usable disk space that provides we typically plan for 2 replicas for redundancy purposes with 20% headroom on each disk. This breaks down to 26.6 TB usable disk across this 8 node cluster.

The calculation is as follows:

Usable Disk = Raw Storage / (3 * 1.2)
Usable Disk = 96 TB / 3.6
Usable Disk = 26.6 TB

Typical Hadoop Cluster – Multiple Racks

Hortonworks recommends starting with a multiple rack configuration for ease of expansion. Typically, a large Hadoop cluster consists of a three-level architecture built with rack-mounted servers. Each rack of servers is interconnected using a 1 Gigabit Ethernet (GbE) switch. Each rack-level switch is connected to a cluster-level switch (which is typically a larger port-density 10 GbE switch). These cluster-level switches may also interconnect with other cluster-level switches or even uplink to another level of switching infrastructure.



The Master Nodes and Worker Nodes would be typically deployed in two racks, interconnected through a cluster-level switch (which is typically a larger port-density 10 GbE switch). For purposes of this cluster planning document, we are assuming a single rack is 42U.

Recommended hardware specifications for a large cluster are as follows:

Rack 1: Cluster	
Master Nodes **4 x 2U Servers**	2 – Intel Sandybridge hexa core processors 8 – 4 GB DDR3 rDIMMs (Sparing, ECC) 1 – Perc S100 (RHEL6/xfs – RAID 10) 6 – 1 TB, NL-SAS disks Built-in dual GbE Dual Power Supply
Worker Node **8 x 2U Servers**	Intel Quad core Sandybridge processor 8 - 4 GB DDR3 DIMMs 2 – Internal 2.5" SATA drives for root (RHEL6/xfs) 6 – 2 TB, SATA disks (SATA/ahci) Built-in dual GbE Dual Power Supply
Rack-level Switch: 1U	24 – ports GbE 1 – 10 GbE SFP (optional)
Cluster-level Switch: 1U	24 – ports GbE
Rack 2-n: Cluster Data Nodes	
Data Nodes **20 x 2U Servers**	Intel Sandybridge Octo core 16 - 4 GB DDR3 DIMMs 2 – Internal 2.5" SATA drives for root (RHEL6/xfs) 1 – 12x3.5" Drive Bay Option 12 – TB, SATA disks (SATA/ahci) Built-in dual GbE Dual Power Supply
Rack-level Switch: 1U	24 – ports GbE per rack

This multi-rack cluster is sized initially for 240 TB of Raw Storage. To calculate how much usable disk space that provides, we typically plan for 2 replicas for redundancy purposes with 20% headroom on each disk. This breaks down to 66.7 TB usable disk across a 20-node cluster.

The calculation is as follows:

Usable Disk = Raw Storage / (3 * 1.2)
Usable Disk = 240 TB / 3.6
Usable Disk = 66.7 TB

Additional capacity can be added to this multi-rack cluster by half-rack (10) or full-rack (20) increments of worker nodes. A half-rack of worker nodes would add 120 TB of additional Raw Storage (33.4 TB of additional Usable Disk). A full-rack of worker nodes would add 240 TB of additional Raw Storage (66.7 TB of additional Usable Disk).

Conclusion

The greatest trait of Apache Hadoop cluster design is, the larger it gets the more resilient it gets. When you have a large numbers of nodes, the loss of a disk, server or even a whole rack, diminishes. When you make your component selections, we have found more nodes are better than more powerful nodes. Also the less you pay for each node the more nodes you can buy. This directly translates into how much work you can process with your cluster.

Key Takeaway: A recipe for configuring a clean, stable Hadoop cluster
We leave you with a set of guidelines and suggestions on how to configure a clean, stable infrastructure on which to install your Apache Hadoop cluster. These guidelines were developed by engineers with years of Hadoop experience and present their preferred default configuration. Following these guidelines will help you avoid 80% of the throughput bottlenecks in a cluster. If this is your first cluster, you should start here, get the cluster stabilized and then start modifying it for what you are doing. This will get you a known stable baseline system.

- RedHat Enterprise Linux 6.3 – there are improvements in this release for memory page cache BDI flusher performance. Hortonworks Data Platform 1.2 or higher – picks up fadvise/drop-behind and the 4 MB readahead for file system performance
- Use XFS file system on all HDFS LUNs – as of 3/10/13 there is a write bug in ext4, ext4 read performance is fine
- For 2 TB disks – create a 250 GB temp and a 1.75 HDFS partition, this will isolate HDFS temp directory churn, on a 12 drive node this gives you 3 TB of temp which is sufficient
- Use XFS on the temp directory partitions also
- Do not create /hdfs or HDFS temp directory on the root volume
- Slots: For Intel processors a dual processor hex core system will show up as a 24 CPU system. However half of the “CPUs” are due to the Intel Hyper Threading feature. To account for this, set the Apache Hadoop map slots to 2/3 of the number of cores.

As noted above these suggestions are all implemented before you install Apache Hadoop. The next step after this is to implement Hortonworks Data Platform pre-installation checklist.

Apache Hadoop Cluster design is a serious platform-engineering project and design decisions need to be well understood. What you can get away with in a small cluster may cause issues as the cluster grows. This guide has given you the basics of Hadoop Cluster design, which will enable you to easily design a small cluster. We strongly recommend you engage Hortonworks or Actian if you need to build a large Internet-scale cluster. Our proven experience in big data allows us to help you avoid common design issues and deliver a stable, working Hadoop cluster.

Analytics database as a Hadoop cluster complement

Legacy RDBMSs and data warehouses are well-suited to their traditional roles, but they were not designed to deal with the very complex, high-volume analytical workloads of today. Hadoop has stepped in to fill that gap, but it, too, has its strengths and weaknesses. A database specifically designed for high volume, low latency analytics can complement a Hadoop deployment, and tackle the things that Hadoop doesn’t do as well, such as interactive, fast response, low latency, ad-hoc querying.

The ParAccel Big Data Analytics Platform intelligently manages workloads and can gather and process data from a wide variety of possible sources, not just Hadoop, but also the cloud, web and other external sources such as log data, unstructured data, applications, data warehouses and data marts. Like Hadoop, the ParAccel Platform does not replace existing enterprise information management architecture, so much as supplement it. It enables cooperative analytic processing by accelerating the processing of very demanding analytical workloads and facilitating interactivity among different data sources and platforms. While in this paper we focus on the capabilities of ParAccel Dataflow in conjunction with Hadoop, Dataflow is only one component of the ParAccel Big Data Analytics Platform.

The other two components are ParAccel MPP, which provides extreme performance for interactive, fast response in-database analytics on cluster scale data sets; and ParAccel SMP, which delivers extremely fast performance – rapid-deploy-and-go interactive in-database analytics and business intelligence – in a single server configuration. The ParAccel Platform also has unique, on-demand integration to fit seamlessly into existing enterprise systems.

To these features, ParAccel Dataflow adds: data provisioning, data cleansing, data discovery, workflow orchestration and a very rich advanced analytic processing capability, which can be built in a codeless manner and implemented directly on Hadoop clusters without moving the data. Or, if needed, ParAccel Dataflow can provide a powerful, high-performance data pump between the various parts of the enterprise information architecture.

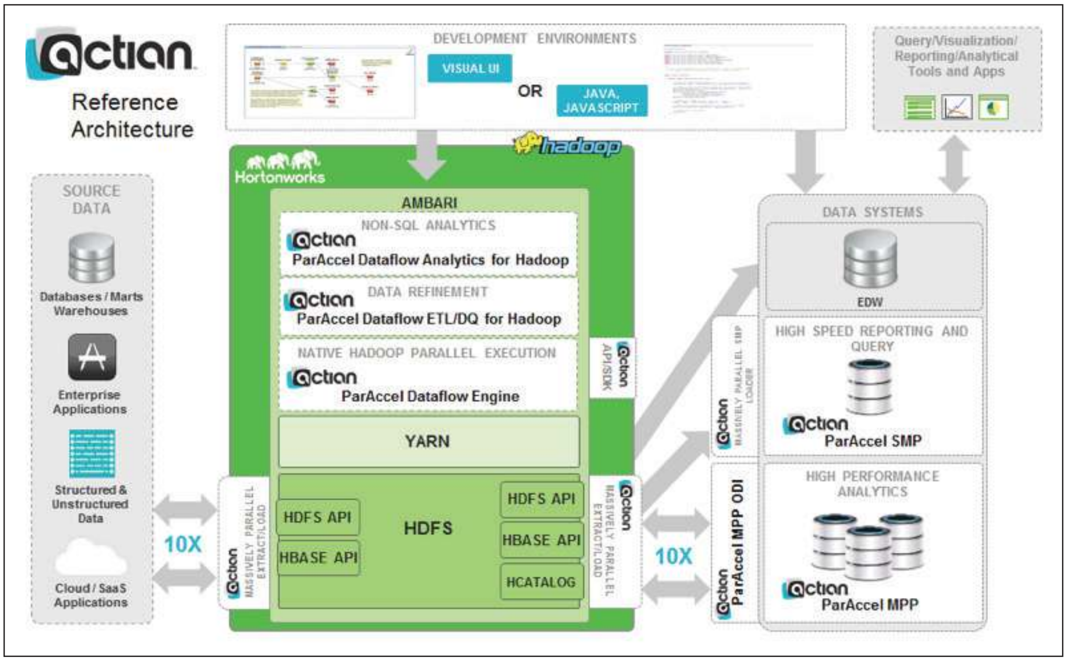
Taken together, these components can implement high-performance analytic capabilities across multiple systems, minimizing the movement of data and providing the muscle when and where it is needed to process complex analytics at extreme scale.

Benefits of the ParAccel Big Data Analytics Platform

ParAccel SMP can be easily downloaded, quickly mastered, and rapidly placed into production for new applications. For companies that want to embed analytics in their application or support a group of divisional or mid-market analysts, ParAccel provides the right-sized solution. It nicely complements small clusters. It can be deployed in a single-node instance, and is designed from the ground up for speed. We vectorize our workloads and take advantage of the latest server, memory, and chip technology in ways that make a single server seem like many.

ParAccel MPP is a high-performance analytic database, designed from the ground up for speed. We have surrounded this database with an extensibility framework that embeds 600 analytic functions and supports on-demand integration of a variety of data and analytic results, right at the point of execution. It scales without limits to handle any size of workload, just like Hadoop, making it an excellent complement to medium to large scale deployments.

ParAccel Dataflow provides ETL, data quality, and advanced analytics for Hadoop clusters. ParAccel is the only platform that provides a high-performance engine that allows you to extract value from data anywhere along its journey. ParAccel Dataflow was built from the ground up to speed the iterative process used by the smartest analysts. Use our KNIME-based design interface to drag and drop a thousand operators to capture, prepare, enrich, and analyze data within or beyond Hadoop. Or dig down deep into the API with your choice of JVM languages: Java, Javascript, Scala, Groovy, Jython, JRuby, etc. There's no limit to what you can accomplish, with no need to worry about the complexities of parallel programming, since the Dataflow framework handles parallel workload optimization for you automatically.



For more information, please visit the following links:

Actian ParAccel Big Data Analytics Platform: http://www.actian.com/platform/paraccel-platform
Hortonworks Hadoop Cluster Size-O-Tron: http://hortonworks.com/resources/cluster-sizing-guide/

Actian Corporation
500 ARGUELLO STREET
SUITE 200
REDWOOD CITY
CALIFORNIA 94063
USA
PHONE: +1.650.587.5500

Actian Austin
12365-B RIATA TRACE PKWY
AUSTIN
TEXAS 78727
USA
PHONE: +1.512.231.6000

Actian Europe Limited
217 BATH ROAD
SLOUGH BERKSHIRE,
SL1 4AA
UNITED KINGDOM
PHONE: +44 (0) 17.5355.9500

Actian Germany GmbH
OHMSTRASSE 12
63225 LANGEN
GERMANY
PHONE: +49 (0) 6103.9881.0

Actian France
IMMEUBLE GABRIEL VOISIN
79 RUE JEAN-JACQUES
ROUSSEAU
92150 SURESNES
FRANCE
PHONE: +33 (0) 1.80.03.11.50

Actian Australia
LEVEL 8, SUITE 1
616 ST. KILDA ROAD
MELBOURNE, VICTORIA, 3004
AUSTRALIA
PHONE: +61 3 8530.1700

Contact Us:

WWW.ACTIAN.COM
+1.888.446.4737 (TOLL FREE)

ACTIAN.COM
FOR MORE INFORMATION, CONTACT ACTION_TEAM@ACTIAN.COM
TO VIEW ADDITIONAL OFFICES: ACTIAN.COM/ABOUT/WORLDWIDE-OFFICES

CG1013US