



# Microsoft SQL Server 2012 Parallel Data Warehouse



A Breakthrough  
Platform for Next-  
Generation Data  
Warehousing and Big  
Data Solutions

# Contents

4	Summary Statement
4	Introduction
6	PDW's Capabilities
6	Scalable, Fast, and Reliable
7	Easy to Use and Manage
7	Platform for Data Mining and Analysis
8	What Customers Are Saying
10	Purpose-Built for Data Warehousing Workloads
10	Designed for Parallel Processing
11	Hardware and Software Designed Together
13	Scalable Solution for Today's Data Challenges
13	Easy to Add Capacity
15	Easy to Load Data Fast
17	Easy to Retain More Data
18	Easy to Consolidate Data

19	Why Queries Run Fast
19	Queries Run on Distributed Data
20	Queries Are Highly Parallelized
22	Data Movement Service Transfers Data Fast
22	In-Memory Clustered Columnstore Indexes Improve
23	Why You Can Trust PDW
24	PDW's Hardware Redundancy
24	High Availability
25	PDW is Easy to Use and Manage
27	Platform for Data Mining and Analysis
28	PolyBase for Hadoop Integration
30	Business Intelligence Integration
30	Summary

# Summary Statement

**This paper explains PDW's thoughtful design so that you can learn more about how PDW achieves breakthrough results, and how upgrading to SQL Server 2012 PDW will benefit your business. PDW's benefits are so remarkable that you might think it all sounds too good to be true! However, PDW's benefits are real and the result of a well-engineered system that is purpose-built for data warehousing workloads, including the growing business needs for data warehousing to integrate the analysis of Hadoop and relational data.**

Microsoft's® SQL Server® 2012 Parallel Data Warehouse (PDW) is a next-generation platform built to run your data analysis fast and to scale storage from a few terabytes to over 6 petabytes in a single appliance. PDW ships to your data center as an appliance with hardware and software pre-installed and pre-configured for maximum performance. With PDW's Massively Parallel Processing (MPP) design, queries run in minutes instead of hours, and in seconds instead of minutes in comparison to today's Symmetric Multi-Processing (SMP) databases. PDW is not only fast and scalable, it is designed with high redundancy and high availability, making it a reliable platform you can trust with your most business critical data. PDW is designed for simplicity which makes it easy to learn and to manage. PDW's PolyBase technology for analyzing Hadoop data, and its deep integration with Business Intelligence tools make it a comprehensive platform for building end-to-end solutions.

# Introduction

If your business relies on data, you know that it is a constant challenge to store, manage, and analyze it effectively as data continues to proliferate. It's also expensive to keep enough data on "hot" storage where it is readily available for analysis. Even when you have the data you need on hot storage, it can take hours or even days to run analysis and reports on today's symmetric multi-processing (SMP) systems. To add more to the challenges, businesses today are struggling to figure out how to add the value of non-relational Hadoop data into their analysis.

As a result, business analysts are held back from making faster and more accurate data-driven business decisions that are needed to compete in today's marketplace. This is the modern data challenge.

If you can relate to today's data dilemma, consider upgrading to SQL Server 2012 Parallel Data Warehouse (PDW), Microsoft's next generation platform for data warehousing and Big Data integration. With PDW's massively parallel processing (MPP) design, queries commonly complete 50 times faster than traditional data warehouses built on symmetric multi-processing (SMP) database management systems.

---

50 times faster means that queries complete in minutes instead of hours, or seconds instead of minutes. With this breakthrough performance, your business analysts can generate more extensive results faster, and can easily perform ad-hoc queries or drill down into the details. As a result, your business can make better decisions, faster.

---

In addition to achieving breakthrough query performance, PDW makes it easy to:

- Grow your data warehouse to anywhere from a few terabytes to over 6 petabytes of data in a single appliance by adding “scale units” to your existing system,
- Trust your data will be there when you need it because of the built-in high redundancy and high availability,
- Solve modern data challenges of loading and consolidating data,
- Integrate Hadoop data with relational data for fast analysis by using PDW’s highly parallelized PolyBase technology,
- Use Business Intelligence tools to build comprehensive end-to-end solutions.

**How is this possible?** Would you like to understand how PDW achieves such remarkable results? This paper details how PDW performs in real-world situations and then explains how PDW’s design is able to achieve these outcomes through a well-engineered data platform that is purpose-built for data warehousing workloads and Big Data integration.

# PDW's Capabilities

## Scalable, Fast, and Reliable

PDW is more than a product, it's a data platform for next-generation data warehousing and Big Data solutions.

The following benefits of PDW are all the result of a well-engineered system that is designed specifically for data warehousing workloads.

### **Scalable**

PDW achieves breakthrough scalability to expand storage from a few terabytes to over 6 petabytes in a single appliance. Unlike SMP systems that require purchasing a new server and storage in order to add more processing and storage capacity, PDW expands by adding "scale units" to the existing appliance. After adding capacity, PDW does all the work of redistributing data and balancing I/O to the additional scale units.

### **Fast**

PDW's Massively Parallel Processing (MPP) design can load data in real-time and finish complex queries up to 50 times faster than today's Symmetric Multi-Processing (SMP) systems. Queries complete in seconds instead of minutes, and minutes instead of hours compared to today's SMP databases such as SQL Server 2008 R2, SQL Server 2012, Oracle, or DB2. Some queries that are impossible to run on today's SMP systems can complete on PDW. Companies no longer need to settle for canned reports and summarized results in exchange for quicker response times. Ad-hoc analysis and detailed reporting are possible and on a sustained basis.

### **Reliable**

PDW is a platform you can trust to keep your data safe and available. It is well-engineered with high-redundancy and high availability built-in across hardware and software components. User data is always stored and managed by SQL Server 2012, which has an impeccable reputation for keeping data safe.

# Easy to Use and Manage

PDW is designed for simplicity. The complexity is already engineered into the appliance so that you don't have to handle the details. For example, PDW handles all the details of distributing the data across the appliance nodes, performs all the extra steps required to process queries in parallel, and manages the low-level hardware and software configuration settings.

PDW arrives at your data center pre-configured and pre-tested; simply plug PDW in and configure it for your networking environment. You can start using it the same day it arrives!

As a result, the learning curve is minimal. You can easily retrain your staff without hiring new PDW experts. Your SQL Server DBAs can easily transfer their SQL Server knowledge to PDW.

PDW Simplicity. No need to manage...

- disk or database subsystems
- storage space
- parallel queries
- distributed data
- software configuration
- hardware configuration

# Platform for Data Mining and Analysis

## **Integrated with Hadoop**

PDW's PolyBase technology enables you to query Hadoop data and merge it with your relational data by using Transact-SQL and without learning a host of new skills. By using PolyBase's Transact-SQL interface you can do more in-depth data mining, reporting, and analysis without acquiring the skills to run MapReduce queries in Hadoop. For example, queries can combine Hadoop and PDW data in a single step, Hadoop data can be stored as relational data in PDW, and query results can be stored back to Hadoop.

PDW's PolyBase is an easy way to use the power of PDW's MPP architecture to analyze Hadoop data fast. PolyBase gives you the flexibility to structure the Hadoop data you need, when you need it, as you bring it into PDW for analysis. With PolyBase, queries that run too slow in Hadoop can now run quickly in PDW. This opens up a new world of data integration and analysis possibilities.

# What Customers Are Saying

## **Integrated with Business Intelligence Tools**

PDW's deep integration with Business Intelligence (BI) tools makes PDW a comprehensive platform for building end-to-end data mining and analysis solutions. PDW integrates with the Microsoft BI Stack including Reporting Services, Analysis Services, PowerPivot for Excel, and PowerView. But, PDW also integrates with a growing list of leading non-Microsoft BI platforms, such as Business Objects, Cognos, SAP Data Integrator, Tableau, MicroStrategy, QlikView, Oracle Business Intelligence, and TIBCO Spotfire.

PDW customers are experiencing fantastic results. The following results from our customers are similar to what most people are saying about PDW.

### **Queries Are Fast**

- Our queries are completing 76 times faster on PDW. This was after PDW compressed 1.5 TB to 134 GB.
- Our queries are completing in less than 1.5 seconds on a month of data. In comparison, using SQL Server 2008 R2, the same queries completed in 2-3 minutes on 7 days of data.
- Our queries run 25 times faster on PDW. The queries complete with an average of 23 seconds per query on PDW. On our SMP system, the queries completed with an average of 5 minutes and 36 seconds.
- Our queries complete 18 times faster on PDW. The queries complete with an average of 53 seconds per query on PDW. In comparison, on our SMP system, the queries complete with an average of 16 minutes per query.
- With a heavy load on our PDW, our individual query times only slowed down by 16 percent.

### **Scales Past SMP**

- PDW made it possible for us to complete queries that could not complete our current system. Even though we ran our queries on a DL980 with 40 cores, 2 terabytes of RAM, and a sizeable storage array, the queries could not finish without running out of memory. After loading our data onto PDW, we ran queries on 7 years of data with 200 users on the system. These queries finished with an average time of 689 seconds per query.
- We had 40 databases sharded among 10 servers because of SMP performance limitations. With PDW, we now have all of the databases consolidated into one database. The data is much easier to maintain and load. Even better, we are able to run aggregation queries on a single database.

### **Loading Is Fast, from hours to minutes**

- We must load one hour of data in less than an hour. One hour of data contains 6 billion records and we are able to load this onto PDW in less than an hour.
- Our daily load now takes 5.5 minutes on PDW whereas it used to take 2.5 hours without PDW. The load is 27 times faster on PDW.
- We can load 520 million rows into PDW in 13 minutes, whereas our old system took 16 hours to load the same data.
- PDW performs well with our mixed workload; our users continue to run their queries while PDW loads occur in the background.

### **High Data Compression Rates**

By using in-memory clustered columnstore indexes to store data on disk, PDW achieves high data compression rates that save storage costs and improve query performance. These are typical examples of customer compression rates.

- 1.5 TB (uncompressed) compressed to 134 GB (7x compression).
- 5.5 TB (uncompressed) compressed to 400 GB (14x compression).
- 120 GB (uncompressed) compressed to 2 GB (60x compression).

### **Customer Reference**

In a case study

([http://www.microsoft.com/casestudies/Case\\_Study\\_Detail.aspx?CaseStudyID=710000002669](http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?CaseStudyID=710000002669)), the sales and marketing service firm, **CROSSMARK**, needed faster and more detailed insight into terabytes of information about product supply and demand. With PDW they achieved this and as a result, **CROSSMARK** was able to improve service, create reports at least 50 percent faster, improve employee efficiency, and save money.

# Purpose-Built for Data Warehousing Workload

“We started using PDW the same day it arrived at our data center. The appliance arrived with the hardware already configured and the software already installed.”

PDW is an appliance solution designed specifically for data warehousing workloads. From the ground up, the hardware and software were carefully designed together to achieve high scalability, performance, reliability, and high availability. Whereas, SQL Server 2012 works great for online transaction processing (OLTP) workloads that update thousands or millions of records on a daily basis, PDW is designed to be great at loading, storing, and analyzing the OLTP or Hadoop data that your business generates on a daily basis.

# Designed for Parallel Processing

Instead of simply extending the SQL Server 2012 Symmetric Multi-Processing (SMP) design, PDW uses a Massively Parallel Processing (MPP) design to achieve breakthrough query results and other performance capabilities discussed in this paper.

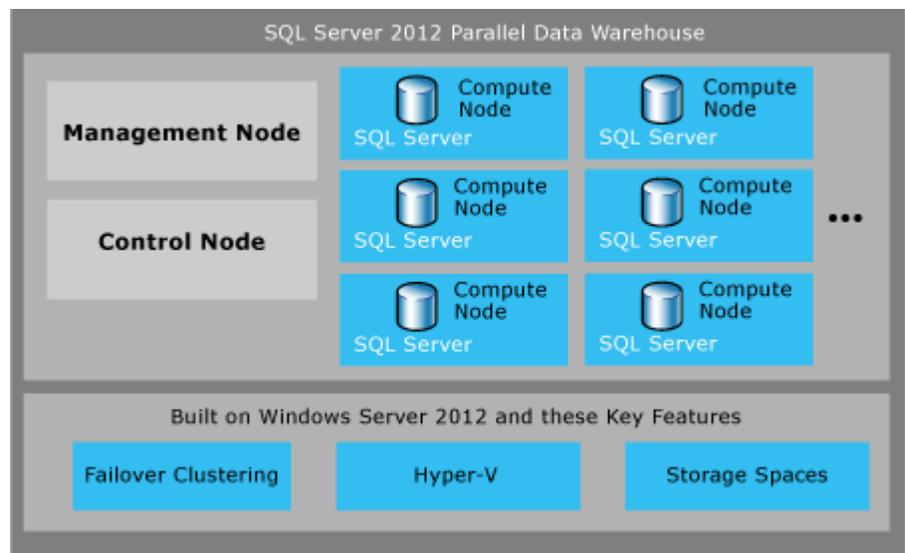


Figure 1: SQL Server 2012 PDW uses an MPP design to achieve high performance and scalability.

# Hardware and Software Designed Together

Query processing is highly parallelized. User data is distributed across processing and storage units called Compute nodes. Each Compute node has its own direct attached storage, processors, and memory that run as an independent processing unit. The Control node is the brains of PDW and figures out how to run each user query in parallel across all of the Compute nodes. As a result, queries run fast!

SQL Server 2012 is foundational to PDW and runs on each Compute node to store, manage, and run queries on distributed user data. PDW uses a special build of SQL Server 2012 that uses updateable in-memory clustered columnstore indexes for high compression rates and fast performance on the individual Compute nodes. Plus, by using SQL Server 2012, PDW has the backing of SQL Server 2012's impeccable reputation to keep your data safe.

PDW is an appliance solution with hardware and software designed together to achieve maximum performance and scalability. PDW arrives at your data center with the software and hardware already installed, configured, and tested. No tuning is required because the appliance is already built and tuned to balance CPU, memory, I/O, storage, network, and other resources. Simply configure PDW for your own network environment, and it is ready to use.

The first rack of the PDW appliance is called the base rack. Every appliance has at least one base rack with 2 or 3 Compute nodes, depending on vendor hardware.

As your business requirements change, you can expand PDW by adding scale units to the base rack. When the base rack is full, PDW expands by adding additional racks, called expansion racks, and adding scale units to them.

The base rack has two InfiniBand and two Ethernet switches for redundant network connectivity. A dedicated server runs the Control node and the Management node. A spare server ships in the rack for failover clustering. Optionally, you can add a second spare server.

The number of Compute nodes in the base rack varies according to the hardware vendor. For example, HP has 2 Compute nodes per scale unit and Dell has 3.

Pictured below is an HP base rack plus 3 scale units for a total of 8 Compute nodes. Dell's base rack scales to 9 Compute nodes.

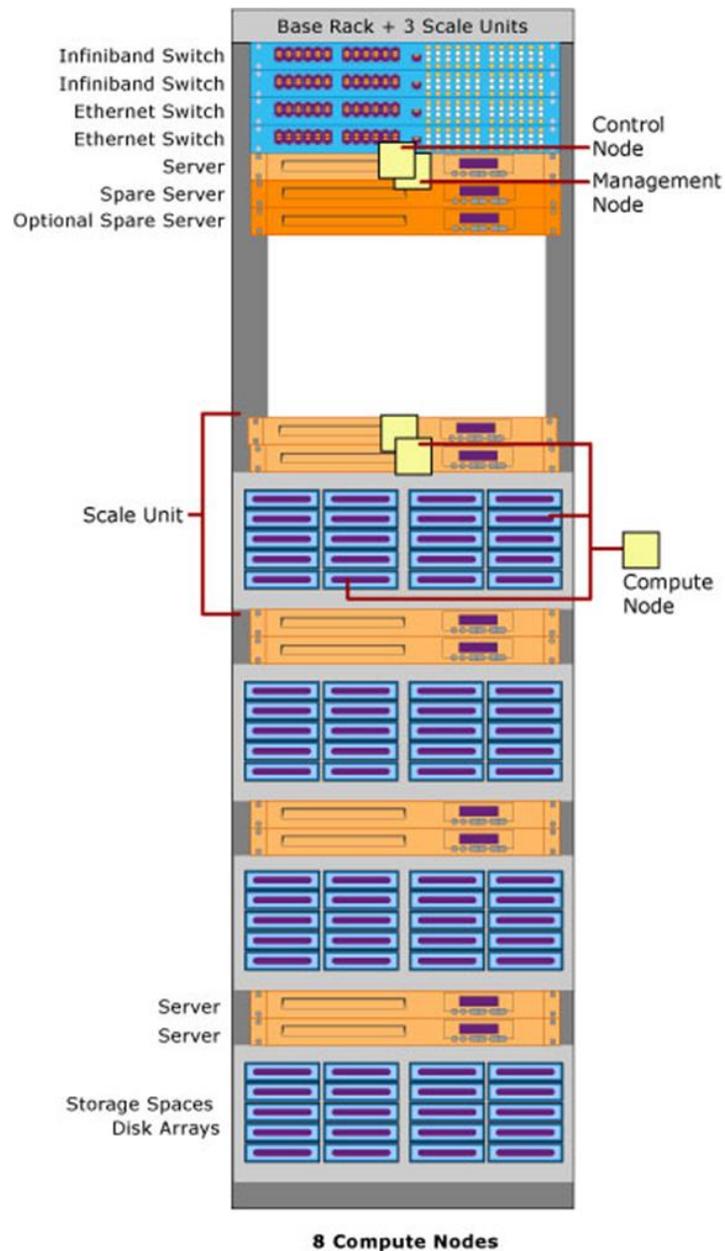


Figure 2: Software and hardware are designed together for parallel processing and scalability.

### High-Speed InfiniBand Network

Networking is also very important for the appliance to achieve breakthrough query performance. The hardware ships with dual InfiniBand networks, and supports fast data transfer up to 56 Gb/s by using FDR InfiniBand and Mellanox ConnectX-3 FDR InfiniBand network adapters. Each rack has redundant Ethernet switches for connecting to your corporate network.

For fast loading and database backups, we recommend connecting your own loading and backup servers to the appliance's InfiniBand network.

## Scalable Solution for Today's Data Challenges

---

“We are able to consolidate our data from multiple databases onto PDW where we can run analysis queries that simply weren't possible before on our old databases.”

---

PDW does more than achieve breakthrough query performance and scalability. The thoughtful design of PDW results in a solution capable of solving data challenges that face businesses today as they try to load, store, manage, and analyze data in order to make faster and more accurate business decisions.

---

## Easy to Add Capacity

---

“Our data is growing faster than expected and we need more capacity within the next 12 months.”

---

Eventually, as your data grows, you need to plan for more capacity. With SMP systems, such as SQL Server 2008 R2, SQL Server 2012, Oracle, or DB2, the only way to add more capacity is to purchase a bigger system that has faster processors, more memory, and more storage. Incremental growth is not a viable option, especially for adding more processors.

Even when you purchase a bigger system, it inevitably becomes difficult to efficiently use one system to meet your growing processing and storage requirements. As a result, businesses are often burdened with the complexity and cost of developing and maintaining their own software to distribute their workload across multiple computing resources.



Figure 3: Scalability of today's SMP systems compared to PDW's MPP design

---

With PDW's MPP design, you do not need to buy a new system in order to add capacity. Instead, PDW grows by adding to the existing system. You don't need to over-buy and waste storage that you don't need and if you under-buy you can quickly add more capacity if your data growth is faster than projected. You also don't need to migrate your data to a new system in order to add capacity. You can scale out without having to redesign your application or re-engineer the distribution mechanism.

PDW is designed to expand processing, memory, and storage by adding scale units consisting of 2 to 3 Compute nodes. By scaling out, you can easily expand capacity to handle a few terabytes to over 6 petabytes in a single appliance. When one rack is full, you can purchase another rack and start filling it with Compute nodes. There is no need to migrate your data to a new system, and no need to restructure your database files to accommodate more nodes. PDW takes care of redistributing your data across more Compute nodes.

The following diagram shows a DELL expansion rack with scalability options.

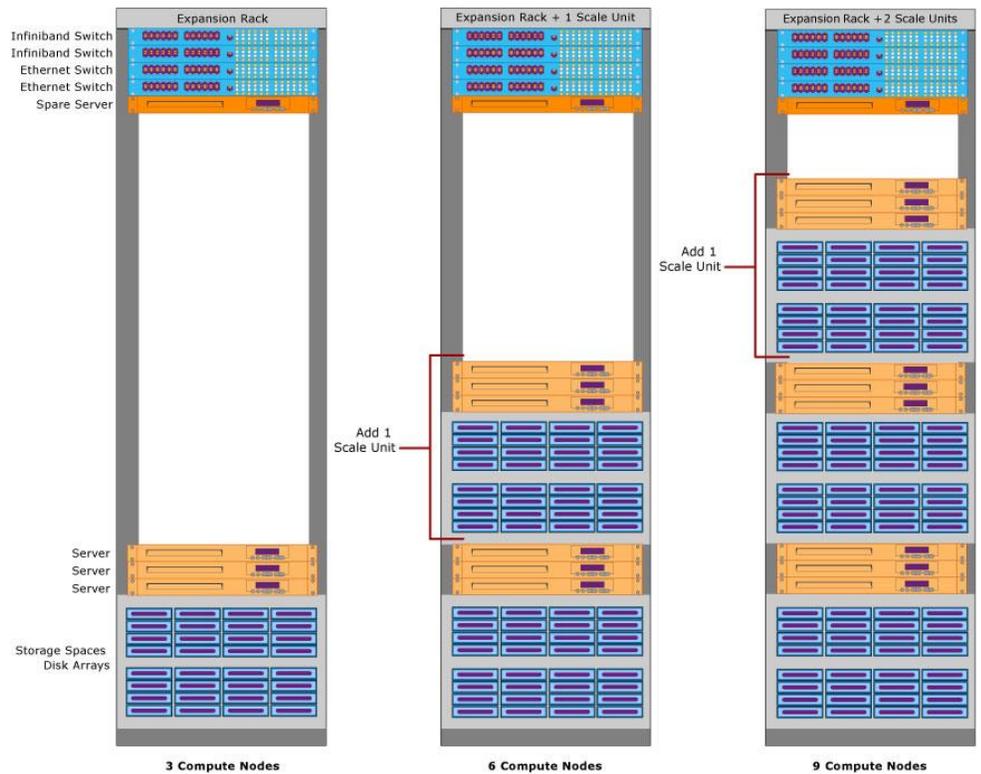


Figure 4: After the base rack is full, PDW grows by adding scale units to expansion racks

## Easy to Load Data Fast

“With our old system, it took 4 hours to load one day of data. With PDW, the same load takes only a few minutes and we are able to load the data without incurring any downtime.”

You might have plenty of storage space available in your data warehouse, but if you can't get the data inserted into SQL Server fast enough, your loading process will be the bottleneck.

PDW loads data up to 7 times faster than SQL Server 2012. Loads are faster with PDW than SMP SQL Server because the data is loaded, in parallel, into multiple instances of SQL Server. For example, if you have 10 Compute nodes and you load 1 Terabyte of data, you will have 10 independent SQL Server 2012 databases that are each compressing and

bulk inserting 100 GB of data at the same time. For most customers, this is 7 times faster than loading 1 TB into one instance of SQL Server 2012.

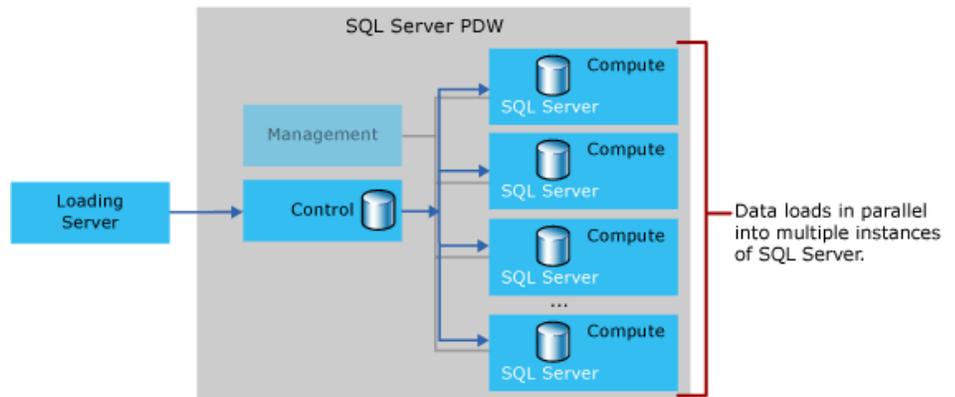


Figure 5: Data loads in parallel into multiple instances of SQL Server

Using PDW, loading is simpler than loading into SMP SQL Server. For SMP SQL Server, the Data Loading Performance Guide ([http://msdn.microsoft.com/en-us/library/dd425070\(v=SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/dd425070(v=SQL.100).aspx)) explains optimizations for loading data. These are suggestions and configurations that you can perform manually. PDW has many of these optimizations already built-in to the PDW architecture and design, making it easier to optimize than SMP SQL Server.

PDW ships with a command-line loading tool called dwloader. Using this tool is the fastest way to load data into PDW and popular among customers who do in-database set-based transformation of data. You can also use SQL Server Integration Services (SSIS) to bulk load data into PDW. The data is loaded from an off-appliance loading staging server into PDW Compute nodes. Informatica also works with PDW.

Loading Tools:

- dwloader Command-line Loading Tool
- SQL Server Integration Services (SSIS)

To load data into the Compute nodes by using dwloader, first use your own ETL process to create the source data you wish to load. The source data must be formatted to match the schema of the destination table. Store the source data into one or more text files or gzip files, and copy files into the same directory on your loading server. Then, run dwloader on your loading server (or ETL server) and dwloader will communicate with the Control node to perform the load.

## Easy to Retain More Data

---

“With PDW, we can store enough data to analyze trends over the last several years. On our old system, we could only store enough data to analyze trends for just 1 year.”

---

As the demand for more data analysis and insight grows, businesses are faced with the challenge of retaining more online data in their data warehouse. Often, the data warehouse is not big enough to store all of the data that analysts need to make more accurate business decisions. You also might have compliance reasons for keeping a particular number of years of data online.

A common way to manage data growth is to use the sliding window approach to data retention in which older data is moved onto tapes or other cheap storage in order to make room for more recent data. For example, you could keep 12 months of data online in your data warehouse. Every month, you might archive the oldest partition to tape, and then reuse the online storage for the current month.

Data stored offline onto tapes is hard to access and therefore impractical to analyze. To analyze data stored on tapes, the data must first be restored to an online system. This takes time and requires available disk space to hold the restored data. Sometimes it's a challenge just to find the right tapes.

PDW's scale-out architecture makes it easy to add capacity so that you can retain more data online for analysis and reporting. Plus, just like SMP systems, you still have the flexibility to partition your data and to archive portions of it off the data warehouse according to your business needs.

# Easy to Consolidate Data

---

“Our data was in so many different places, that we couldn’t keep track of it, or even begin to analyze it. Now, with PDW, we have consolidated many databases onto PDW.”

---

Businesses get data from web clicks, customer transactions, social networks, and a plethora of other data sources. Consolidating disparate data into one data warehouse is often not feasible because of the data size. As a result, management of the data is decentralized and complex.

Wouldn’t it be nice to consolidate your data into one central data warehouse? This makes the data warehouse much more usable because the data is in one place and can be managed centrally. Simplifying data warehouse management lowers the total cost of ownership. You only need to manage one system, and plan for capacity growth for that one system. Your business also benefits from being able to run queries on any of the data which is all stored on one system, or to run queries that select from multiple databases.

Consolidation has several business benefits:

- As hardware reaches end-of-life, instead of a complicated process of refreshing many hardware systems, you can migrate the data to PDW and then manage the data on one system.
- DBA’s can manage all of the data warehouse data in one place which makes their job simpler, and reduces management costs.
- PDW uses Windows Server Update Services to perform software updates on all of the servers. One software update through Windows Server Update Services updates all of the servers in the appliance.
- All of your data sets benefits from the redundancy and high availability of the appliance.
- You can monitor the health and state of the appliance from one centralized place. PDW has an admin console that DBA’s and appliance administrators can use to see the state of queries and the health of the appliance.

- Perform more comprehensive analysis where analysts can ask questions involving multiple data sources that are all centrally located. This makes it easy to cross-analyze data from different data sources.

## Why Queries Run Fast

---

“When we first started to use PDW, we couldn’t believe how fast our queries were finishing. In fact, we kept running them “again” just to convince ourselves the queries actually ran.”

---

PDW’s MPP design is the cornerstone for breakthrough query performance. Queries run fast because they run in parallel on distributed data. PDW’s cost-based query optimizer figures out how to run each parallel query fast, and Data Movement Service (DMS) efficiently transfers data among the Compute nodes.

To support parallel query processing, PDW distributes fact table rows across the Compute nodes and stores the table as many smaller physical tables.

Within each Compute node the distributed data is stored into 8 physical tables that are each stored on independent disk-pairs. Each independent storage location is called a distribution. PDW runs queries in parallel on each distribution. Since every Compute node has 8 distributions, the degree of parallelism for a query is determined by the number of Compute nodes. For example, if your appliance has 8 Compute nodes your queries will run in parallel on 64 distributions across the appliance. The following diagram shows data being distributed to four Compute nodes and 32 distributions.

## Queries Run on Distributed Data

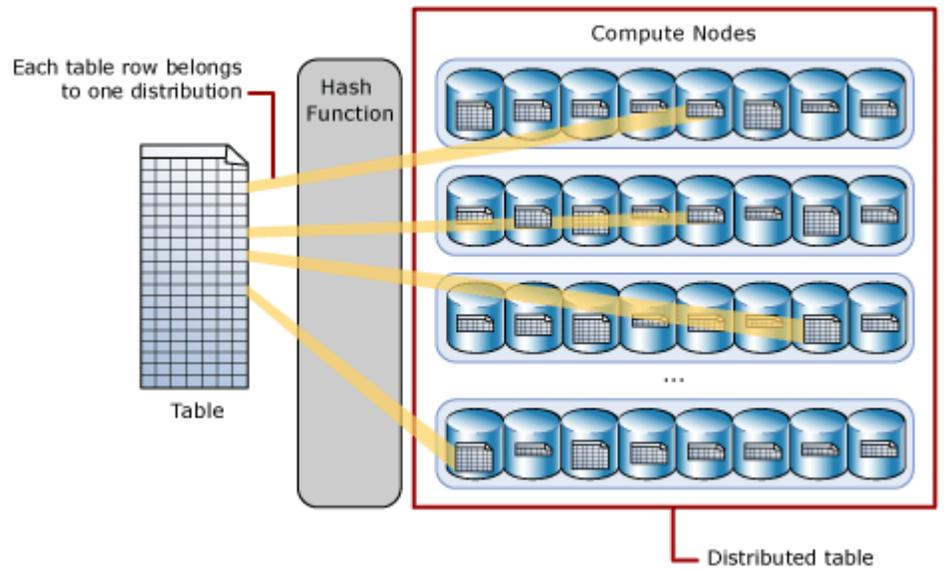


Figure 6: Fact table rows are stored across all of the Compute node distributions

When PDW distributes a fact table, it uses one of the columns as the key for determining the distribution to which the row belongs. A hash function assigns each row to a distribution according to the key value in the distribution column. Every row in a table belongs to one and only one distribution. If you don't choose the best distribution column when create the table, it is easy to re-create the table to use a different distribution column.

PDW doesn't require that all tables get distributed. Small dimension tables are usually replicated to each Compute node. Replicating small tables speeds query processing since the data is always available on each Compute node and there is no need to waste time transferring the data among the nodes in order to satisfy a query.

PDW's cost-based query optimizer is the "**secret sauce**" that makes parallel queries run fast and return accurate results. A result of Microsoft's extensive research and development efforts, the query optimizer uses proprietary algorithms to successfully choose a high performing parallel query plan.

The parallel query plan, also called the "dsq" plan, contains all of the operations necessary to run the query in parallel. As a result, PDW handles all the complexity that comes with parallel processing and processes the query seamlessly in parallel behind the scenes. The results are streamed back to the client as though only one instance of SQL Server ran the query.

Queries Are  
Highly  
Parallelized

Here's a look into how PDW query processing works "under the covers".

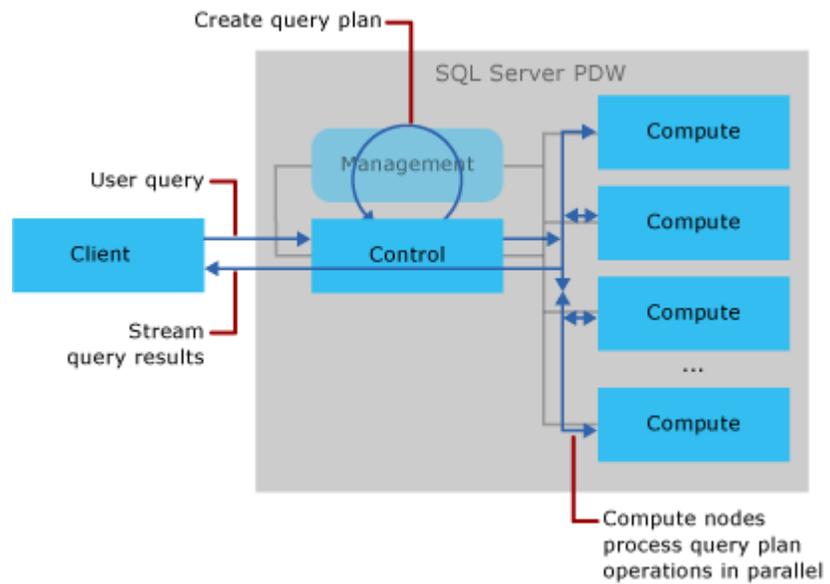


Figure 7: PDW chooses a query plan that runs queries fast and accurately in parallel.

First, a query client submits a Transact-SQL user query to the Control node, which coordinates the parallel query process. All query clients connect with SQL Server Native Client, or .NET, so you can use SQL Server Data Tools, sqlcmd, or your own application to submit the query.

### Query Clients

- Customer LOB applications that connect with SQL Server Native Client or .Net Framework
- SQL Server Data Tools
- SQL Server's sqlcmd Command-Line SQL Client

After receiving a query, PDW's cost-based parallel query optimizer uses statistics to choose a query plan, from many options, for running the user query in parallel across the Compute nodes. The Control node sends the parallel query plan, called the dsql plan, to the Compute nodes, and then the Compute nodes run the parallel query plan on their portion of the data.

The Compute nodes each use SQL Server 2012 to run their portion of the query. When the Compute nodes finish, the results are quickly streamed back to the client through the Control node. All of this occurs quickly without landing data on the Control node, and the data does not bottleneck at the Control node.

# Data Movement Service Transfers Data Fast

PDW data relies on colocated data, which means the right data must be on the right Compute node at the right time before running a query. When two tables use the same distribution column they can be joined without moving data. Data movement is necessary, though, when a distributed table is joined to another distributed table and the two tables are not distributed on the same column.

PDW uses Data Movement Service (DMS) to move data, as necessary, for parallel query processing. DMS only moves the minimum amount of data necessary to satisfy the query. Since data movement takes time, the query optimizer considers the cost of moving the data when it chooses a query plan.

# In-Memory Clustered Columnstore Indexes Improve Query Performance

PDW uses in-memory clustered columnstore indexes to improve query performance and to store data more efficiently. These indexes are updateable, and are applied to the data after it is distributed.

A clustered columnstore index is a technology for storing, retrieving and managing data by using a columnar data format, called a columnstore. The data is compressed, stored, and managed as a collection of partial columns, called column segments.

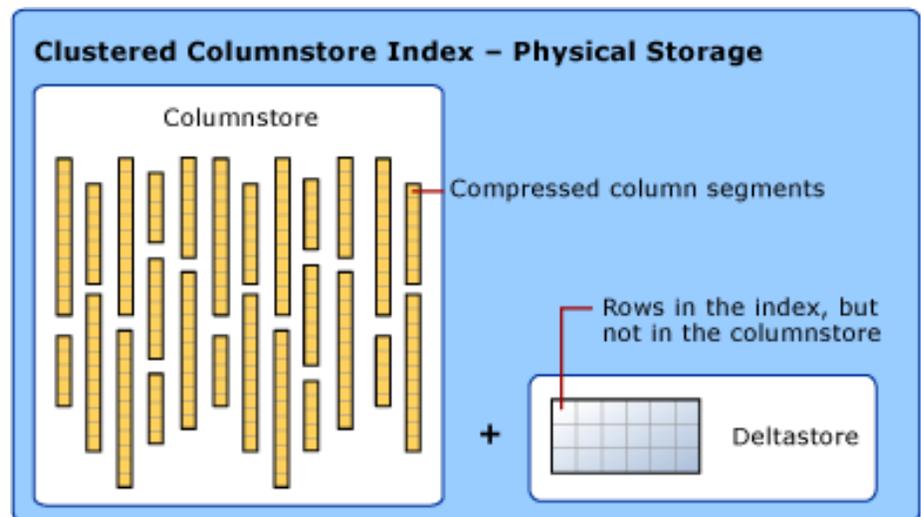


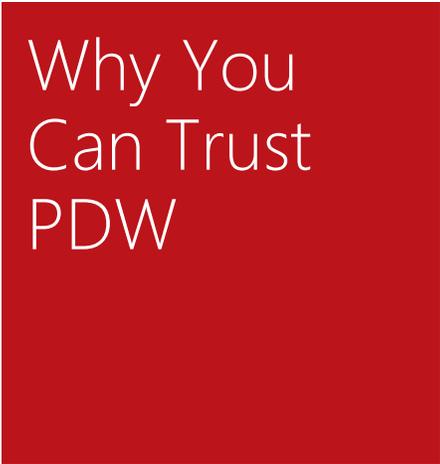
Figure 8: Clustered columnstore indexes improve query speed and data compression

PDW uses the column-based data layout to significantly improve compression rates and query execution time. For example:

- Columns often have similar data, which results in high compression rates. In turn, higher compression rates further improve query performance by using a smaller in-memory and I/O footprint.
- High compression rates improve query performance by using a smaller in-memory footprint. In turn, query performance can improve because SQL Server PDW can perform more query and data operations in-memory.
- Most queries select only a few columns from a table, which reduces total I/O to and from the physical media. The I/O is reduced because columnstore tables are stored and retrieved by column segments, and not by B-tree pages.
- Advanced query execution technology processes chunks of columns called batches in a streamlined manner, which reduces CPU usage.

To prevent fragmented column segments, some of the clustered columnstore index data is stored temporarily in a rowstore table, called a deltastore, until it is compressed and moved into the columnstore. The clustered columnstore index operates on both the columnstore and the deltastore, to return the correct query results.

PDW's design is built in Windows Server 2012 Storage Spaces, Failover Clustering, and Hyper-V to keep your data safe and highly available.



Why You  
Can Trust  
PDW

# PDW's Hardware Redundancy Keeps Your Data Safe

The hardware is designed to make each Compute node run efficiently on its own hardware without interfering with the performance of other Compute nodes or storage. Each Compute node runs in a Virtual machine on one server that is direct-attached to two Windows Storage Spaces disk arrays.

The hardware is designed for redundancy too, so that you can trust PDW to not lose your data.

Capacity is added to PDW by adding scale units. In the scale unit, one Compute node runs on each server, and writes to 16 pairs of dedicated mirrored disks. The number of Compute nodes per scale unit depends on the hardware vendor. The following diagram shows how 2 Compute nodes are grouped together to form a scale unit.

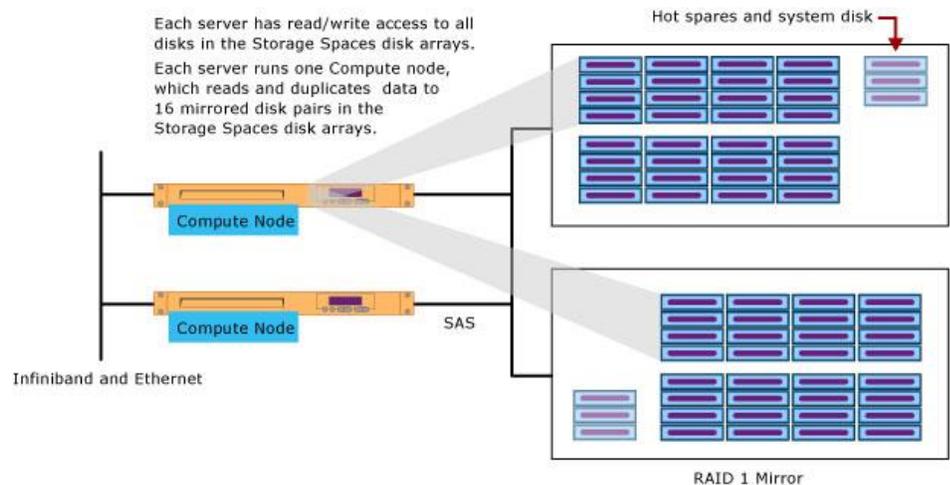


Figure 9: A scale unit with 2 Compute nodes

By using Windows Storage Spaces, each Compute node has read and write access to all the disks in the Windows Storage Spaces disk arrays for the scale unit. If the server fails, the Compute node fails over to a spare server in the rack. It continues to run and uses the InfiniBand network to access its disks through another server in the scale unit.

# High Availability

The hardware and software are architected together for high availability. Using features in Windows Server 2012, PDW is able to run each node within a virtual environment. With this ability, the nodes can technically run on any server. When a server goes down, PDW uses Windows Failover Clustering to fail the node, which runs in a Virtual Machine, over to the spare server.

Using Windows Storage spaces, each Compute node can read and write to all of the disks in the scale unit, if it needs to. If a Compute node fails to the spare server, the Compute node can continue running and accessing its data and disks through the InfiniBand connection to another server in its scale unit.

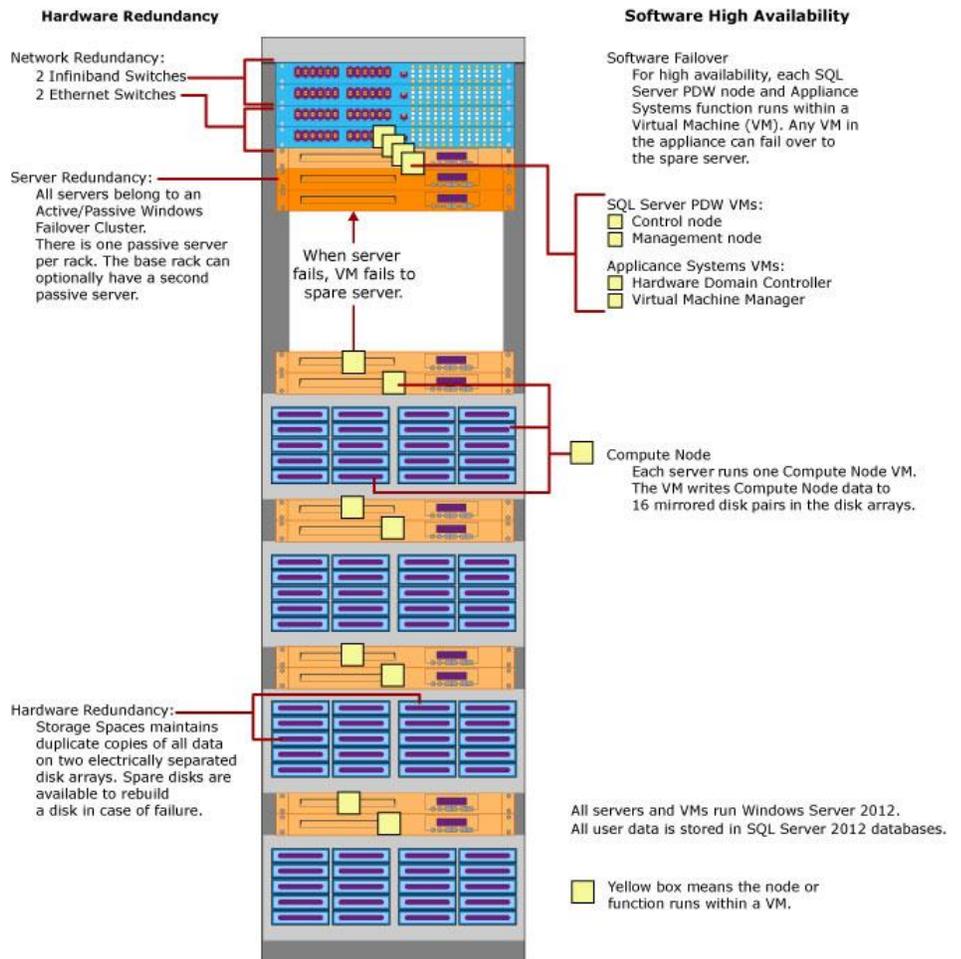


Figure 10: Redundant and high availability design keeps your data safe and online.

PDW is Easy to Use and Manage

“We can’t afford to lose our current employees and hire new ones to support a new data warehouse technology. Please make it simpler to maintain a data warehouse!”

**PDW has a minimal learning curve.** There's no need to hire new talent in order to move from SQL Server to SQL Server PDW. DBAs who already know Transact-SQL can easily transfer their knowledge to PDW. Client queries are written with Transact-SQL statements. Some statements are added or extended to accommodate the MPP architecture. Some statements are not needed because they don't apply to MPP, or PDW is already performing the operations behind the scenes. There's no need to hire new talent in order to move from SQL Server to SQL Server PDW.

**PDW is easy to configure.** When the appliance arrives at your data center, the hardware and software are already installed and configured. PDW ships with a checklist of tasks for configuring your system, and has a built-in Configuration Manager for perform many of the configurations.

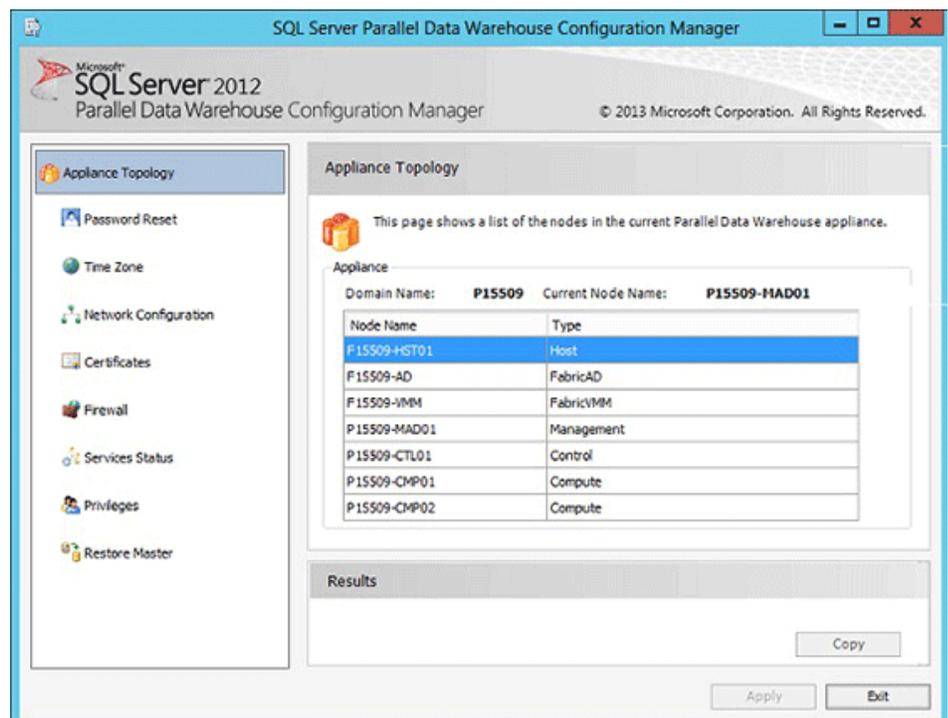


Figure 11: Administrators configure the appliance with the Configuration Manager Tool

**PDW has built-in monitoring tools.** There is a web-based Admin Console that users and administrators can use to monitor the appliance. The Admin Console has information about queries, loads, and backups. It also has information about the appliance health, and storage resources.

Information surfaced by the Admin Console can also be viewed through scripts that query dynamic management views. In addition, PDW has Management Packs so that you can use System Center Operations Manager (SCOM) to monitor PDW.



Figure 1.2: PDW has a web-based Admin Console for monitoring the appliance.

## Platform for Data Mining and Analysis

“It’s not enough to store data in Hadoop. Businesses today need to figure out how they can analyze Hadoop data fast and seamlessly in order to make more informed business decisions.”

In today’s world, data is generated from many sources, some of which are not suitable for relational database management systems. For example, businesses could have clickstream data that is generated at a rapid rate, social sentiment data with different schemas, sensor data, or telemetry data feeds. The buzz about Big Data is not only about figuring out how to store, manage, and analyze data from non-relational sources, but also about mashing various non-relational data with an organization’s relational data to gain business insight.

# PolyBase for Hadoop Integration

Today, many businesses are exploring possibilities with Hadoop to store their non-relational data. Hadoop is open source software that manages non-relational data across many nodes. It is easy to add data to Hadoop, but not as quick to extract and analyze the data you need. The idea is that, if the data is there, it may take a while to retrieve it, but at least the data is stored somewhere in the system. Hadoop is best for batch-oriented processing.

PDW's PolyBase is a breakthrough technology for analyzing Hadoop data, and it opens up a new world of data analysis possibilities. Queries that run too slow in Hadoop can now run quickly in PDW, data mining queries can combine Hadoop and PDW data, Hadoop data can be stored as relational data in PDW, and query results can be stored back to Hadoop. By using the power of PDW to run queries on Hadoop data, it is now possible to do more in-depth data mining, reporting, and analysis without acquiring the skills to run MapReduce queries in Hadoop.

PolyBase gives you the flexibility to bring the Hadoop data you need, when you need it, into PDW for fast analysis. You can seamlessly select from both Hadoop data and PDW data in the same query, and join data from both data sources. To satisfy a query, PolyBase transfers data quickly and directly between PDW's Compute nodes and Hadoop's DataNodes.

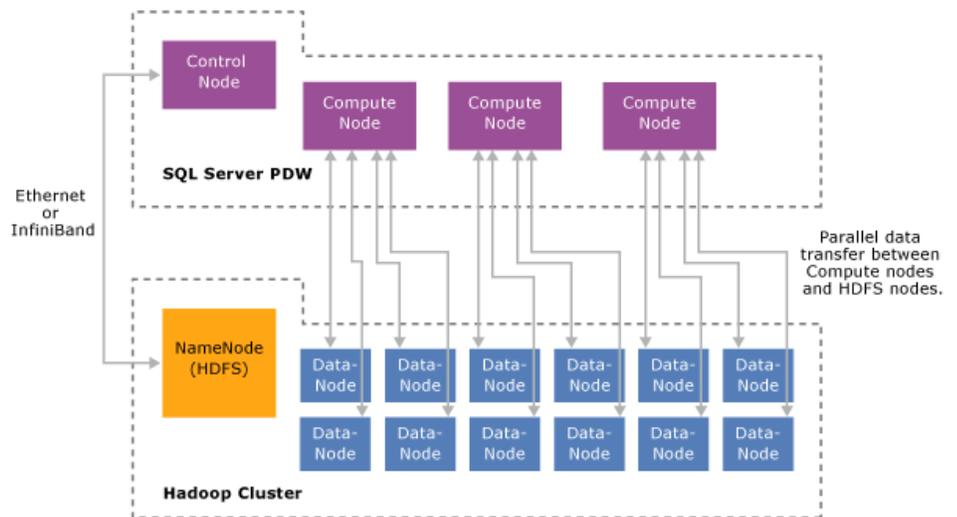


Figure 13: Data transfers in parallel between PDW Compute nodes and HDFS DataNodes.

PDW uses external tables to point to data stored on Hadoop. Once an external table is created, the table can be used in a SELECT statement in the same manner as a PDW table.

The following example creates an external table called DimCustomer\_Hadoop\_Export and transfers data from the DimCustomer table in PDW to the DimCustomerExport.txt file residing in the Hadoop Cluster.

```
CREATE EXTERNAL TABLE DimCustomer_Hadoop_Export
WITH
(
    LOCATION =
'hddfs://10.192.63.147:8020/DemoDb/DimCustomerExport.t
xt,
    FORMAT_OPTIONS ( FIELD_TERMINATOR = '|' )
)
AS SELECT * FROM DimCustomer;
```

To create an external table without transferring data, the syntax is similar to creating a PDW table. This example creates an external table called ClickStream that points to data stored in a text file called employee.txt on Hadoop.

```
CREATE EXTERNAL TABLE ClickStream (
    url varchar(50),
    event_date date,
    user_IP varchar(50)
)
WITH (
    LOCATION =
'hddfs://10.192.63.147:8020/DemoDb/employee.txt',
    FORMAT_OPTIONS (
        FIELD_TERMINATOR = '|'),
        DATE_FORMAT = 'MM/dd/yyyy'
    )
)
[;]
```

To view this external table in PDW or to perform operations on the data, simply SELECT from it as from any other table.

```
SELECT * FROM ClickStream;
```

For an example of combining Hadoop data and PDW data to solve a problem, see the video [Insight Through Integration: SQL Server 2012 Parallel Data Warehouse Demo - PolyBase Video](http://go.microsoft.com/fwlink/?LinkId=320541) (<http://go.microsoft.com/fwlink/?LinkId=320541>).

PDW's deep integration with Business Intelligence (BI) tools makes PDW a comprehensive platform for building end-to-end data mining and analysis solutions. PDW integrates with both the Microsoft BI Stack (Reporting Services, Analysis Services, PowerPivot for Excel, PowerView). It also integrates with a growing list of leading non-Microsoft BI platforms, such as Business Objects, Cognos, SAP Data Integrator, Tableau, MicroStrategy, QlikView, Oracle Business Intelligence, and TIBCO Spotfire.

# Business Intelligence Integration

BI tools connect to PDW by using the standard SQL Server connection tools, SQL Server Native Client or the .NET Data Provider for SQL Server. The fact that all these BI tools can access a PDW database as though it is a standard SQL Server 2012 database allows you to quickly update or modify BI solutions to point to PDW.

New in SQL Server 2012 PDW, you can use PDW as a high performance relational data source for building multidimensional OR tabular models with SQL Server Analysis Services (SSAS). For example, you can:

- Use DirectQuery to perform real-time queries against a SQL Server PDW data source from a tabular model.
- Reduce ROLAP query times by using clustered columnstore indexes on SQL Server PDW tables.
- Use new features of SSAS. For example, use `EnableRolapDistinctCountOnDataSource` to run distinct count operations on PDW tables.
- Use PDW as a data warehouse for aggregating Hadoop data for use in multidimensional or tabular models.

## Summary

Microsoft SQL Server 2012 Parallel Data Warehouse is a compelling and innovative solution for next generation data challenges. It enables you to make faster and more accurate business decisions by providing an innovative platform for managing data warehousing workloads that scales from a few terabytes to over 6 petabytes of data in a single appliance. Queries commonly run 50 times faster than traditional relational database systems. Significantly higher data compression ratios allow more data to be stored cost efficiently. PDW has deep integration with Microsoft's BI tools and other leading non-Microsoft tools, making it simple to use the BI tools you are familiar with to perform analysis. PolyBase technology enables you to query Hadoop data and merge it with your relational data by using Transact-SQL and without learning a host of new skills. The management costs are minimal and the learning curve for SQL Server DBAs is easy, making it possible to use your current DBAs to manage PDW.

PDW's breakthrough query performance and scalability result from PDW's purpose-built design for data warehousing workloads. PDW's hardware and software are designed for fast parallel processing, distributed data storage, hardware redundancy, scalable capacity, and high availability. All of these factors give PDW the ability to handle growing storage and analysis needs for your data. As a result, upgrading from SQL Server to SQL Server PDW is an important step forward towards empowering your business to manage growing data needs and towards performing complex analysis that enables you to make faster and more accurate business decisions.

**For additional information:**

- SQL Server 2012 Parallel Data Warehouse Web site: <http://www.microsoft.com/PDW/>
- Microsoft data warehousing web site: <http://www.microsoft.com/en-us/sqlserver/solutions-technologies/data-warehousing.aspx>

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

- Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
- Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback.](#)

**About the Authors**

Barbara Kess leads data warehousing content at Microsoft and is the primary author responsible for SQL Server PDW content.

Dan Kogan leads Database Appliance marketing at Microsoft and is responsible for the SQL Server PDW product.

**Acknowledgements**

A special thank you to the SQL Server PDW engineering, consulting, and sales teams who have dedicated themselves to developing and shipping a great product. They have relentlessly reviewed the PDW product documentation, which much of this paper is based on.

Thanks to Murshed Zaman and John Hoang who are on the SQL Server Customer Advisory Team and use their everyday experiences helping customers to develop best practices for SQL Server PDW. Their deep experience with PDW is embedded in the details of this paper.

Henk van der Valk is a Technical Solution Professional who works with prospective customers who are interested in SQL Server PDW. His input helped us to better explain what our customers want to know.

Thanks to Rick Byham, content author at Microsoft, for reviewing the paper and giving insightful feedback.