

Greenplum Database: Critical Mass Innovation

Architecture White Paper

August 2010

Greenplum Database:

Critical Mass Innovation

Table of Contents

Meeting the Challenges of a Data-Driven World	2
Race for Data, Race for Insight	2
Greenplum Database: Critical Mass Innovation	3
Greenplum Database Architecture	4
Shared-Nothing Massively Parallel Processing Architecture	4
Data Distribution and Parallel Scanning	5
Multi-Level Self-Healing Fault Tolerance	6
Parallel Query Optimizer	7
gNet Software Interconnect	8
Parallel Dataflow Engine	9
Native MapReduce Processing	10
MPP Scatter/Gather Streaming™ Technology	10
Polymorphic Data Storage™	12
Key Features and Benefits of Greenplum Database	13
About Greenplum	15

Meeting the Challenges of a Data-Driven World

Race for Data, Race for Insight

The continuing explosion in data sources and volumes strains and exceeds the scalability of traditional data management and analytical architectures. Decades old legacy architecture for data management and analytics is inherently unfit for scaling to today's big data volumes. These systems require huge outlays of resources and technical intervention in a losing battle to keep pace with demand for faster time to intelligence and deeper insight.

In today's business climate, every leading organization finds itself in the data business. With each click, call, or transaction by a user, or other business activity, data is generated that can contribute to the aggregate knowledge of the business. From this data can come insights that can help the business better understand its customers, detect problems, improve its operations, reduce risks, or otherwise generate value for the business.

While at one time it might have been acceptable for companies to capture just the obviously "important" data and throw the rest away, today leading companies understand that the value of their data goes far deeper. The decision about which data to keep and which to throw away—that is, which data will prove to be important in the future—is deceptively hard to make. The result, according to industry analyst Richard Winter, is that businesses must store and analyze data at the most detailed level possible if they want to be able to execute a wide variety of common business strategies. Combined with increased retention rates of 5 to 7 years or longer, and it is no surprise that typical data volumes are growing by 1.5 to 2.5 times a year.

Even this figure doesn't tell the whole story, however. Looking forward, many businesses realize their future competitiveness will depend on new business strategies and deeper insights that might require data that isn't even being captured today. Projecting five years out, businesses shouldn't be surprised if they want to store and analyze a rapidly growing data pool that is 100 times or more greater than the size of today's data pool. And beyond size, the depth of analysis and complexity of business questions raised against the data can only be expected to grow.

Greenplum's focus is on being the leading provider of database software for the next generation of data warehousing and large-scale analytic processing. The company offers a new, disruptive economic model for large-scale analytics that allows customers to build warehouses that harness low-cost commodity servers, storage, and networking to economically scale to petabytes of data.

From a performance standpoint, the progression of Moore's law means that more and more processing cores are now being packed into each CPU. Data volumes are growing faster than expected under Moore's law, however, so companies need to plan to grow the capacity and performance of their systems over time by adding new nodes. To meet this need, Greenplum makes it easy to expand and leverage the parallelism of hundreds or thousands of cores across an ever-growing pool of machines. Greenplum's massively parallel, shared-nothing architecture fully utilizes each core, with linear scalability and unmatched processing performance.

¹Richard Winter, "Why Are Data Warehouses Growing So Fast? An Update on the Drivers of Data Warehouse Growth." <http://www.b-eye-network.com/view/7188>

Greenplum Database: Critical Mass Innovation

Greenplum Database is a software solution built to support the next generation of data warehousing and large-scale analytics processing. Supporting SQL and MapReduce parallel processing, the database offers industry-leading performance at a low cost for companies managing terabytes to petabytes of data.

Greenplum Database, a major release of Greenplum's industry-leading massively parallel processing (MPP) database product, represents the culmination of more than seven years of advanced research and development by one of the most highly regarded database engineering teams in the industry.

With this release, Greenplum solidifies its role as the only next-generation database software vendor to achieve critical mass and maturity across all aspects required of an enterprise-class analytical/data-warehousing DBMS platform. Across all seven pillars (Figure 1), Greenplum Database matches or surpasses the capabilities of legacy databases such as Teradata, Oracle, and IBM DB2, while delivering a dramatically more cost-effective and scalable architecture and delivery model.

Critical Mass Innovation						
Complex Query Optimization	Data Loading	Workload Management	Fault-Tolerance	Embedded Languages/Analytics	3rd Party ISV Certification	Administration and Monitoring

FIGURE 1. THE SEVEN PILLARS OF ANALYTICAL DBMS CRITICAL MASS INNOVATION

Greenplum Database stands apart from other products in three key areas:

Extreme Scale

- From hundreds of gigabytes to the largest multi-petabyte data warehouses—scale is no longer a barrier
- Software-only approach is uniquely appliance ready

Elastic Expansion and Self-Healing Fault Tolerance

- Add servers while online for more storage capacity and performance
- Reliability and availability features to accommodate all levels of server, network, and storage failures

Unified Analytics

- Single platform for warehousing, marts, ELT, text mining, and statistical computing
- Enable parallel analysis on any data, at all levels with SQL, MapReduce, R, etc.

The Greenplum Database Architecture

Shared-Nothing Massively Parallel Processing Architecture

Greenplum Database utilizes a shared-nothing, massively parallel processing (MPP) architecture that has been designed for business intelligence and analytical processing. Most of today's general-purpose relational database management systems are designed for Online Transaction Processing (OLTP) applications. Since these systems are marketed as supporting data warehousing and business intelligence (BI) applications, their customers have inevitably inherited this less-than-optimal architecture. The reality is that BI and analytical workloads are fundamentally different from OLTP transaction workloads and therefore require a profoundly different architecture.

OLTP transaction workloads require quick access and updates to a small set of records. This work is typically performed in a localized area on disk, with one or a small number of parallel units. Shared-everything architectures, in which processors share a single large disk and memory, are well suited to OLTP workloads. Shared-disk architectures, such as Oracle RAC, can also be effective for OLTP because each server can take a subset of the queries and process them independently while ensuring consistency through the shared-disk subsystem.

However, shared-everything and shared-disk architectures are quickly overwhelmed by the full-table scans, multiple complex table joins, sorting, and aggregation operations against vast volumes of data that represent the lion's share of BI and analytical workloads. These architectures aren't designed for the levels of parallel processing required to execute complex BI and analytical queries, and tend to bottleneck as a result of failures of the query planner to leverage parallelism, lack of aggregate I/O bandwidth, and inefficient movement of data between nodes.

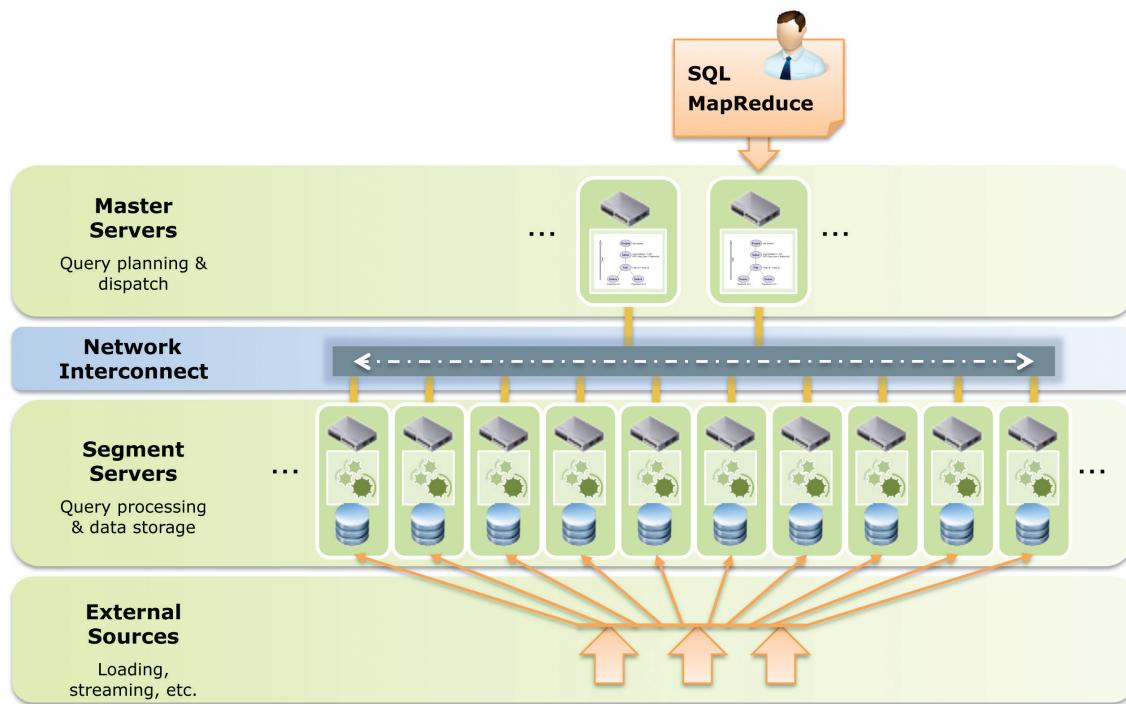


FIGURE 2. GREENPLUM'S MPP SHARED-NOTHING ARCHITECTURE

To transcend these limitations, Greenplum assembled a team of the world's leading database experts and built a shared-nothing massively parallel processing database, designed from the ground up to achieve the highest levels of parallelism and efficiency for complex BI and analytical processing. In this architecture, each unit acts as a self-contained database management system that owns and manages a distinct portion of the overall data. The system automatically distributes data and parallelizes query workloads across all available hardware.

The Greenplum Database shared-nothing architecture separates the physical storage of data into small units on individual segment servers (Figure 2), each with a dedicated, independent, high-bandwidth channel connection to local disks. The segment servers are able to process every query in a fully parallel manner, use all disk connections simultaneously, and efficiently flow data between segments as query plans dictate. Because shared-nothing databases automatically distribute data and make query workloads parallel across all available hardware, they dramatically outperform general-purpose database systems on BI and analytical workloads.

Data Distribution and Parallel Scanning

One of the key features that enables the Greenplum Database to scale linearly and achieve such high performance is its ability to utilize the full local disk I/O bandwidth of each system. Typical configurations can achieve from 1 to more than 2.5 gigabytes per second sustained I/O bandwidth per machine. This rate is entirely scalable, and overall I/O bandwidth can be increased linearly by simply adding nodes without any concerns of saturating a SAN/shared-storage backplane.

In addition, no special tuning is required to ensure that data is distributed across nodes for efficient parallel access. When creating a table, a user can simply specify one or more columns to be used as "hash distribution" keys, or the user can elect to use random distribution. For each row of data inserted, the system computes the hash of these column values to determine which segment in the system the data should be placed on. In the vast majority of cases, data will be equally balanced across all segments of the system (Figure 3).

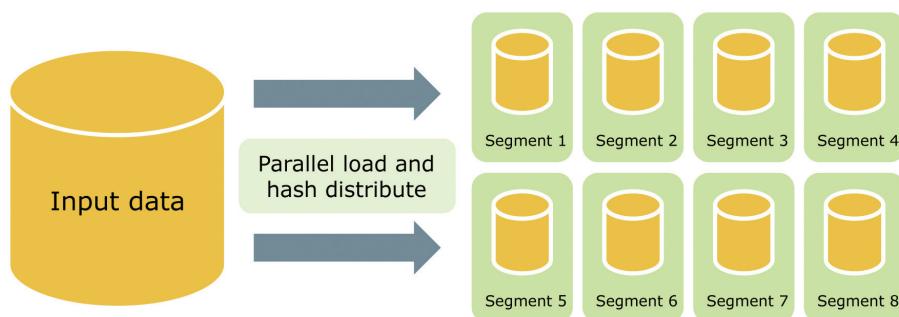


FIGURE 3. AUTOMATIC HASH-BASED DATA DISTRIBUTION

Once the data is in the system, the process of scanning a table is dramatically faster than in other architectures because no single node needs to do all the work. All segments work in parallel and scan their portion of the table, allowing the entire table to be scanned in a fraction of the time of sequential approaches. Users aren't forced to use aggregates and indexes to achieve performance—they can simply scan the table and get answers in unmatched time.

Of course there are times when indices are important, such as when doing single-row lookups or filtering or grouping low-cardinality columns. Greenplum Database provides a range of index types, including b-trees and bitmap indices, that address these needs exactly.

Another very powerful technique that is available with Greenplum Database is multi-level table partitioning. This technique allows users to break very large tables into buckets on each segment based on one or more date, range, or list values. This partitioning is above and beyond the hash partitioning described earlier and allows the system to scan just the subset of buckets that might be relevant to the query. For example, if a table were partitioned by month (Figure 4), then each segment would store the table as multiple buckets with one for each month. A scan for records in April and May would require only those two buckets on each segment to be scanned, automatically reducing the amount of work that needs to be performed on each segment to respond to the query.

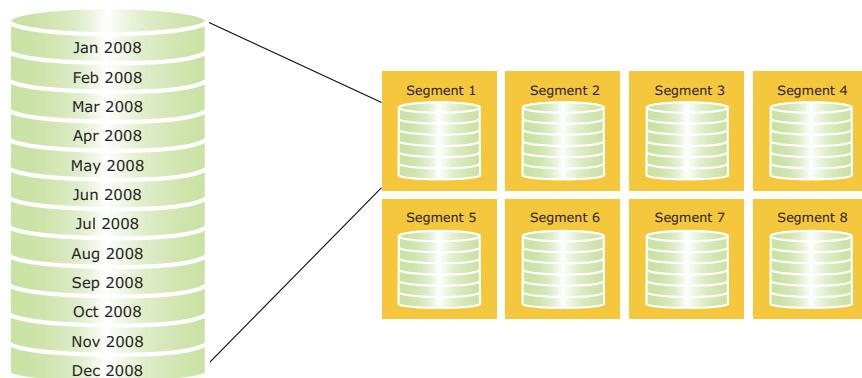


FIGURE 4. MULTILEVEL TABLE PARTITIONING

Multi-Level Self-Healing Fault Tolerance

Greenplum Database is architected to have no single point of failure. Internally the system utilizes log shipping and segment-level replication to achieve redundancy, and provides automated failover and fully online “self-healing” resynchronization.

The system provides multiple levels of redundancy and integrity checking. At the lowest level, Greenplum Database utilizes RAID-0+1 or RAID-5 storage to detect and mask disk failures. At the system level, it continuously replicates all segment and master data to other nodes within the system to ensure that the loss of a machine will not impact the overall database availability. The database also utilizes redundant network interfaces on all systems, and specifies redundant switches in all reference configurations.

With Greenplum Database, Greenplum has expanded its fault-tolerance capabilities to provide intelligent fault detection and fast online differential recovery, lowering TCO and enabling mission-critical and cloud-scale systems with the highest levels of availability (Figure 5).

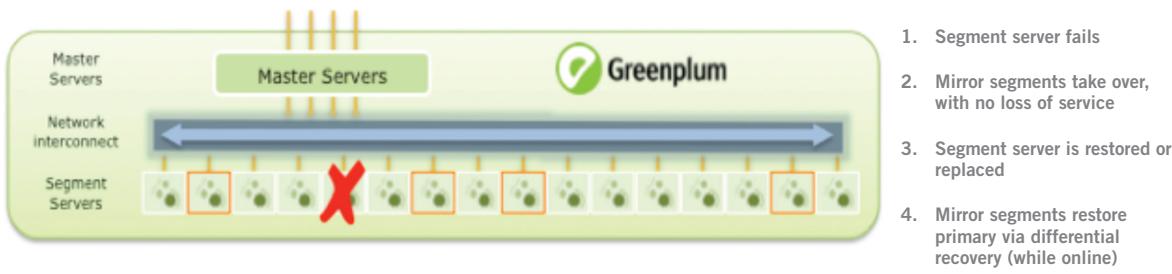


FIGURE 5. GREENPLUM DATABASE MULTI-LEVEL SELF-HEALING FAULT TOLERANCE PROCESS

The result is a system that meets the reliability requirements of some of the most mission-critical operations in the world.

Parallel Query Optimizer

The Greenplum Database parallel query optimizer (Figure 6) is responsible for converting SQL or MapReduce into a physical execution plan. It does this by using a cost-based optimization algorithm to evaluate a vast number of potential plans and select the one that it believes will lead to the most efficient query execution.

Unlike a traditional query optimizer, Greenplum's optimizer takes a global view of execution across the cluster, and factors in the cost of moving data between nodes in any candidate plan. The benefit of this “global” query planning approach is that it can use global knowledge and statistical estimates to build an optimal plan once and ensure that all nodes execute it in a fully coordinated fashion. This approach leads to far more predictable results than the alternative approach of “SQL-pushing” snippets that must be replanned at each node.

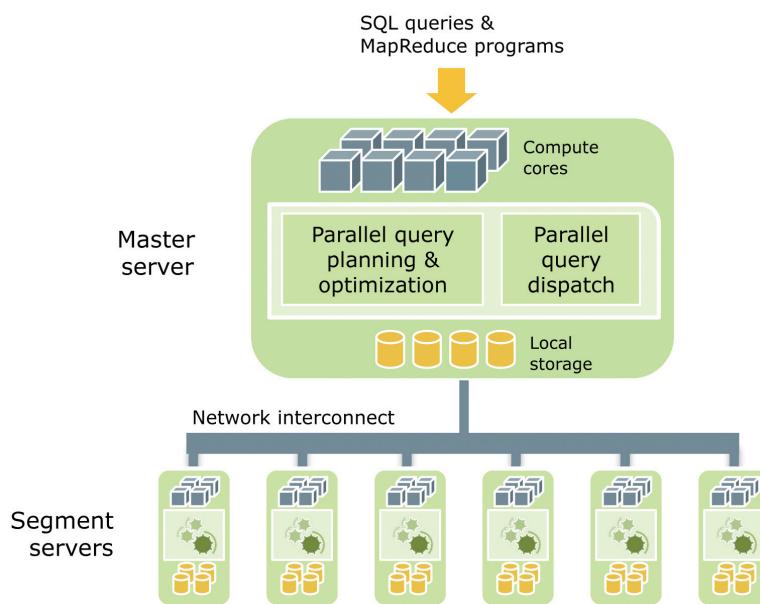


FIGURE 6. MASTER SERVER PERFORMS GLOBAL PLANNING AND DISPATCH

The resulting query plans contain traditional physical operations—scans, joins, sorts, aggregations, and so on—as well as parallel “motion” operations that describe when and how data should be transferred between nodes during query execution.

Greenplum Database has three kinds of “motion” operations that may be found in a query plan:

- *Broadcast Motion (N:N)*—Every segment sends the target data to all other segments.
- *Redistribute Motion (N:N)*—Every segment rehashes the target data (by join column) and redistributes each row to the appropriate segment.
- *Gather Motion (N:1)*—Every segment sends the target data to a single node (usually the master).

The following example shows an SQL statement and the resulting physical execution plan containing “motion” operations (Figure 7):

```
select
    c_custkey, c_name,
    sum(l_extendedprice * (1 - l_discount)) as revenue,
    c_acctbal, n_name, c_address, c_phone, c_comment
from
    customer, orders, lineitem, nation
where
    c_custkey = o_custkey
    and l_orderkey = o_orderkey
    and o_orderdate >= date '1994-08-01'
    and o_orderdate < date '1994-08-01'
        + interval '3 month'
    and l_returnflag = 'R'
    and c_nationkey = n_nationkey
group by
    c_custkey, c_name, c_acctbal,
    c_phone, n_name, c_address, c_comment
order by
    revenue desc
```

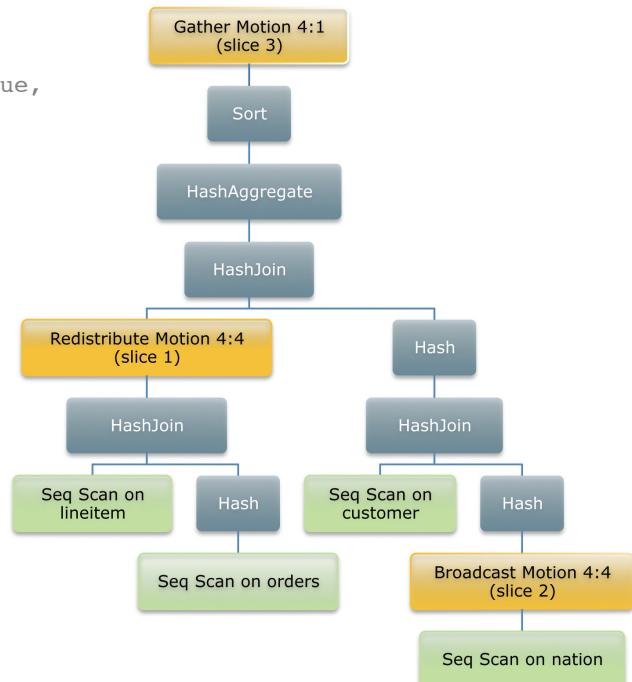


FIGURE 7. EXAMPLE PARALLEL QUERY PLAN

gNet Software Interconnect

In shared-nothing database systems, data often needs to be moved whenever there is a join or an aggregation process for which the data requires repartitioning across the segments. As a result, the interconnect serves as one of the most critical components within Greenplum Database. Greenplum’s gNet interconnect (Figure 8m) optimizes the flow of data to allow continuous pipelining of processing without blocking on all nodes of the system. The gNet interconnect is tuned and optimized to scale to tens of thousands of processors and leverages commodity Gigabit Ethernet and 10GigE switch technology.

At its core, the gNet software interconnect is a supercomputing-based “soft switch” that is responsible for efficiently pumping streams of data between motion nodes during query-plan execution. It delivers messages, moves data, collects

results, and coordinates work among the segments in the system. It is the infrastructure underpinning the execution of motion nodes that occur within parallel query plans on the Greenplum system.

Multiple relational operations are processed by pipelining within the execution of each node in the query plan. For example, while a table scan is taking place, rows selected can be pipelined into a join process. Pipelining is the ability to begin a task before its predecessor task has completed, and this ability is key to increasing basic query parallelism. Greenplum Database utilizes pipelining whenever possible to ensure the highest-possible performance.

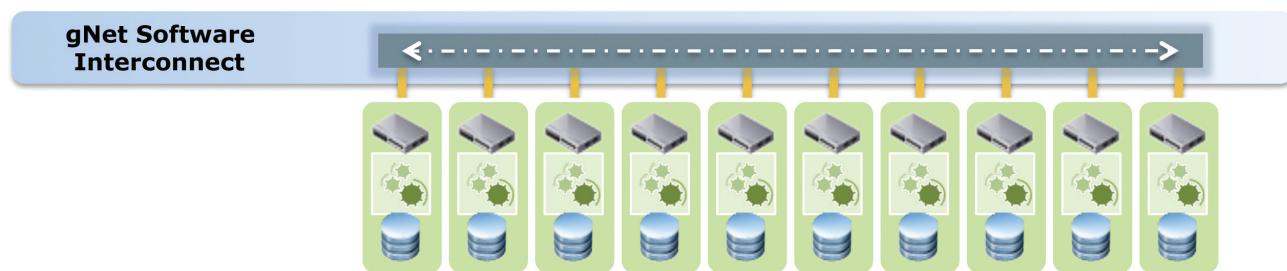


FIGURE 8. GNET INTERCONNECT MANAGES THE FLOW OF DATA BETWEEN NODES

Parallel Dataflow Engine

At the heart of Greenplum Database is the Parallel Dataflow Engine. This is where the real work of processing and analyzing data is done. The Parallel Dataflow Engine is an optimized parallel processing infrastructure that is designed to process data as it flows from disk, from external files or applications, or from other segments over the gNet interconnect (Figure 9). The engine is inherently parallel—it spans all segments of a Greenplum cluster and can scale effectively to thousands of commodity processing cores.

The engine was designed based on supercomputing principles, with the idea that large volumes of data have “weight” (i.e., aren’t easily moved around) and so processing should be pushed as close as possible to the data. In the Greenplum architecture this coupling is extremely efficient, with massive I/O bandwidth directly to and from the engine on each segment. The result is that a wide variety of complex processing can be pushed down as close as possible to the data for maximum processing efficiency and incredible expressiveness.

Greenplum’s Parallel Dataflow Engine is highly optimized at executing both SQL and MapReduce, and does so in a massively parallel manner. It has the ability to directly execute all necessary SQL building blocks, including performance-critical operations such as hash-join, multistage hash-aggregation, SQL 2003 windowing (which is part of the SQL 2003 OLAP extensions that are implemented by Greenplum), and arbitrary MapReduce programs.

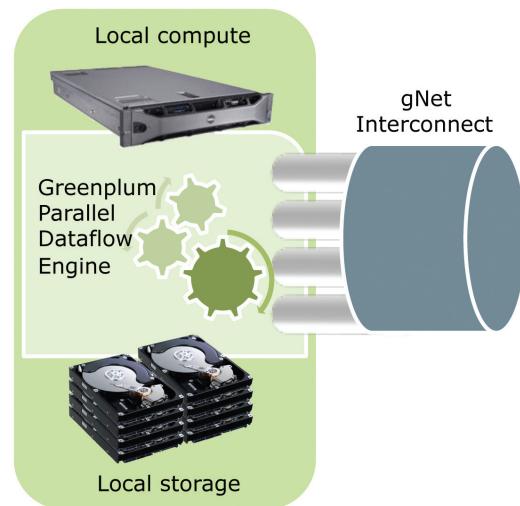


FIGURE 9. THE PARALLEL DATAFLOW ENGINE OPERATES IN PARALLEL ACROSS TENS OR HUNDREDS OF SERVERS

Native MapReduce Processing

MapReduce has been proven as a technique for high-scale data analysis by Internet leaders such as Google and Yahoo. Greenplum gives enterprises the best of both worlds—MapReduce for programmers and SQL for DBAs—and will execute both MapReduce and SQL directly within Greenplum's Parallel Dataflow Engine (Figure 10), which is at the heart of the database.

Greenplum MapReduce enables programmers to run analytics against petabyte-scale datasets stored in and outside of Greenplum Database. Greenplum MapReduce brings the benefits of a growing standard programming model to the reliability and familiarity of the relational database. The new capability expands Greenplum Database to support MapReduce programs.

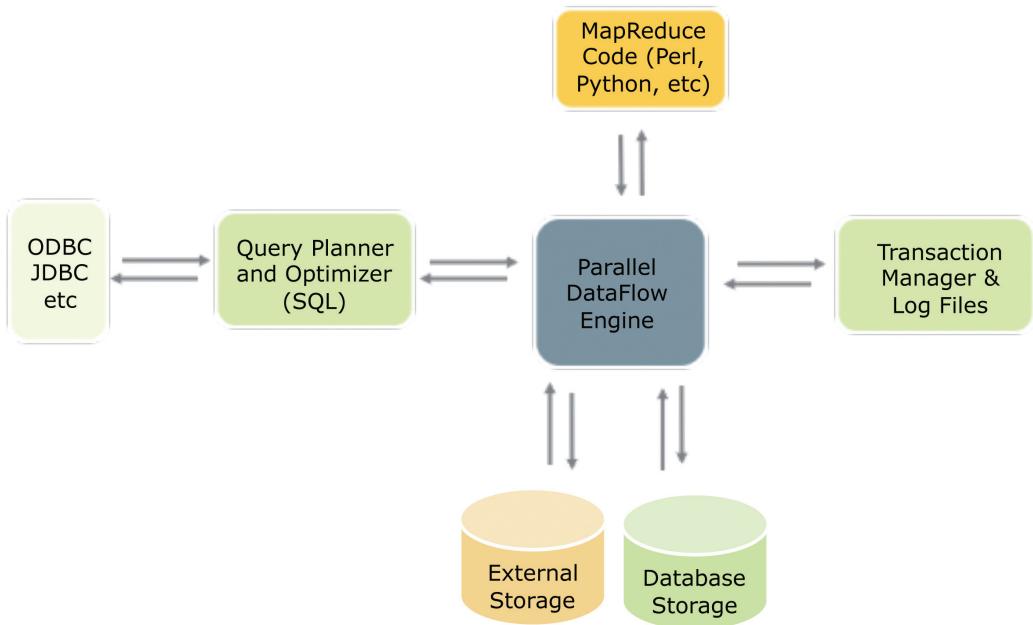


FIGURE 10. BOTH SQL AND MAPREDUCE ARE PROCESSED ON THE SAME PARALLEL INFRASTRUCTURE

MPP Scatter/Gather Streaming™ Technology

Greenplum's new MPP Scatter/Gather Streaming™ (SG Streaming™) technology eliminates the bottlenecks associated with other approaches to data loading, enabling lightning-fast flow of data into Greenplum Database for large-scale analytics and data warehousing. Greenplum customers are achieving production loading speeds of more than 4 terabytes per hour with negligible impact on concurrent database operations.

Scatter/Gather Streaming:

- Manages the flow of data into all nodes of the database
- Does not require additional software or systems
- Takes advantage of the same Parallel Dataflow Engine nodes in Greenplum Database

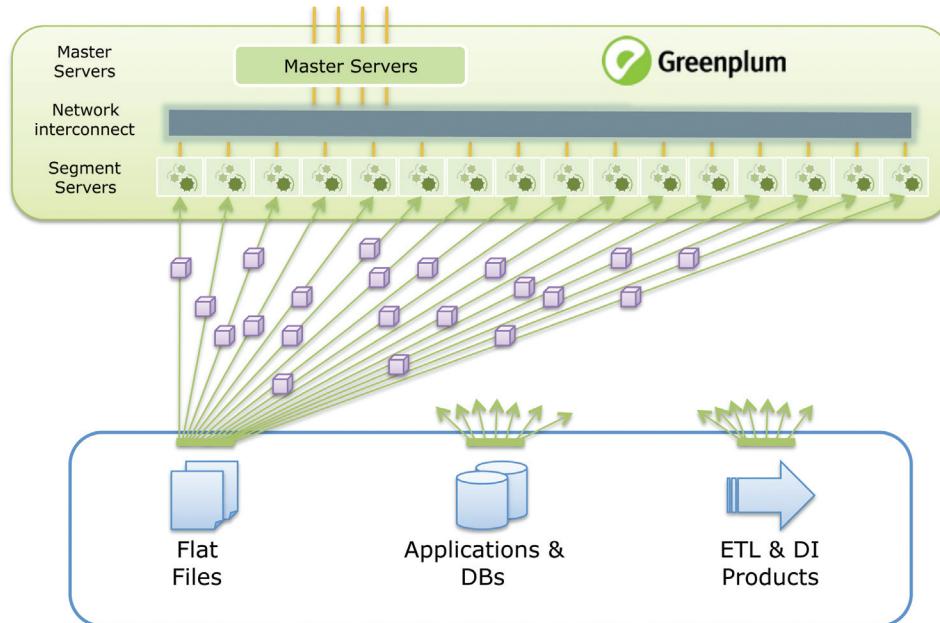


FIGURE 11. "SCATTER" PHASE

Greenplum utilizes a “parallel-everywhere” approach to loading, in which data flows from one or more source systems to every node of the database without any sequential choke points. This approach differs from traditional “bulk loading” technologies—used by most mainstream database and MPP appliance vendors—which push data from a single source, often over a single channel or a small number of parallel channels, and result in fundamental bottlenecks and ever-increasing load times. Greenplum’s approach also avoids the need for a “loader” tier of servers, as required by some other MPP database vendors, which can add significant complexity and cost while effectively bottlenecking the bandwidth and parallelism of communication into the database.

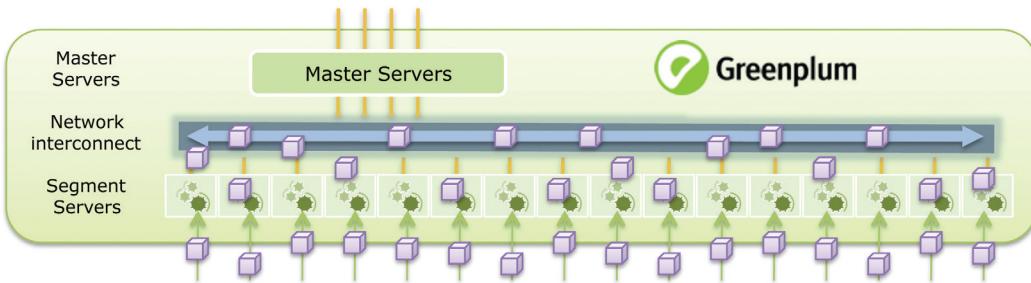


FIGURE 12. "GATHER" PHASE

Greenplum’s SG Streaming™ technology ensures parallelism by “scattering” data from all source systems across hundreds or thousands of parallel streams that simultaneously flow to all Greenplum Database nodes (Figure 11). Performance scales with the number of Greenplum Database nodes, and the technology supports both large batch and continuous near-real-time loading patterns with negligible impact on concurrent database operations. Data can be transformed and processed in-flight, utilizing all nodes of the database in parallel, for extremely high-performance ELT (extract-load-transform) and ETLT (extract-transform-load-transform) loading pipelines. Final “gathering” and storage of data to disk takes place on all nodes simultaneously, with data automatically partitioned across nodes and optionally compressed (Figure 12). This technology is exposed to the DBA via a flexible and programmable “external table” interface and a traditional command-line loading interface.

Polymorphic Data Storage™



FIGURE 13. FLEXIBLE STORAGE MODELS WITHIN A SINGLE TABLE

Traditionally relational data has been stored in rows—i.e., as a sequence of tuples in which all the columns of each tuple are stored together on disk. This method has a long heritage back to early OLTP systems that introduced the “slotted page” layout still in common use today. However, analytical databases tend to have different access patterns than OLTP systems. Instead of seeing many single-row reads and writes, analytical databases must process larger, more complex queries that touch much larger volumes of data—i.e., read-mostly with big scanning reads and infrequent batch appends of data.

Vendors have taken a number of different approaches to meeting these needs. Some have optimized their disk layouts to eliminate the OLTP fat and do smarter disk scans. Others have turned their storage sideways (literally) with column-stores—i.e., the decades-old idea of “vertical decomposition” demonstrated to good success by Sybase IQ, and now reimplemented by a raft of newer vendors. Each of these approaches has proven to have sweet spots where they shine, and others where they do a less effective job.

Rather than advocating for one approach or the other, we’ve built in flexibility so that customers can choose the right strategy for the job at hand. We call this capability Polymorphic Data Storage™. For each table (or partition of a table), the DBA can select the storage, execution, and compression settings that suit the way that table will be accessed (Figure 13). With Polymorphic Data Storage, the database transparently abstracts the details of any table or partition, allowing a wide variety of underlying models:

- *Read/Write Optimized*—Traditional “slotted page” row-oriented table (based on the PostgreSQL native table type), optimized for fine-grained CRUD operations.
- *Row-Oriented/Read-Mostly Optimized*—Optimized for read-mostly scans and bulk append loads. DDL allows optional compression ranging from fast/light to deep/archival.
- *Column-Oriented/Read-Mostly Optimized*—Provides a true column-store just by specifying “WITH (orientation=column)” on a table. Data is vertically partitioned, and each column is stored in a series of large, densely packed blocks that can be efficiently compressed from fast/light to deep/archival (and tend to see notably higher compression ratios than row-oriented tables). Performance is excellent for those workloads suited to column-store. Greenplum’s implementation scans only those columns required by the query, doesn’t have the overhead of per-tuple IDs, and does efficient early materialization using an optimized “columnar append” operator.

An additional axis of control is the ability to place specific tables or partitions on particular storage types or tiers. With Greenplum’s multi-storage/SSD support (tablespaces), administrators can flexibly control storage placement. For example, the most frequently used data can be stored on SSD media for faster response times, and other tables or partitions can be stored on standard disks for cost-effective storage.

Greenplum's Polymorphic Data Storage performs well when combined with Greenplum's multi-level table partitioning. With Polymorphic Data Storage, customers can tune the storage types and compression settings of different partitions within the same table. For example, a single partitioned table could have older data stored as "column-oriented with deep/archival compression," more recent data as "column-oriented with fast/light compression", and the most recent data as "read/write optimized" to support fast updates and deletes.

Advanced Workload Management

Using Greenplum Database Advanced Workload Management, administrators have all the control they need to manage complex, high-concurrency, mixed-workload environments and ensure that all query types get their appropriate share of resources.

These controls begin with connection management—the ability to multiplex thousands of connected users while minimizing connection resources and load on the system. Next are user-based resource queues, which perform "admission control" and manage the flow of queries into the system based on administrator policies. Finally, Greenplum's Dynamic Query Prioritization technology gives administrators full control of runtime query prioritization and ensures that all queries get their appropriate share of system resources (Figure 14).

Together these controls provide administrators with the simple, flexible, and powerful controls they need to meet workload SLAs in complex, mixed-workload environments.

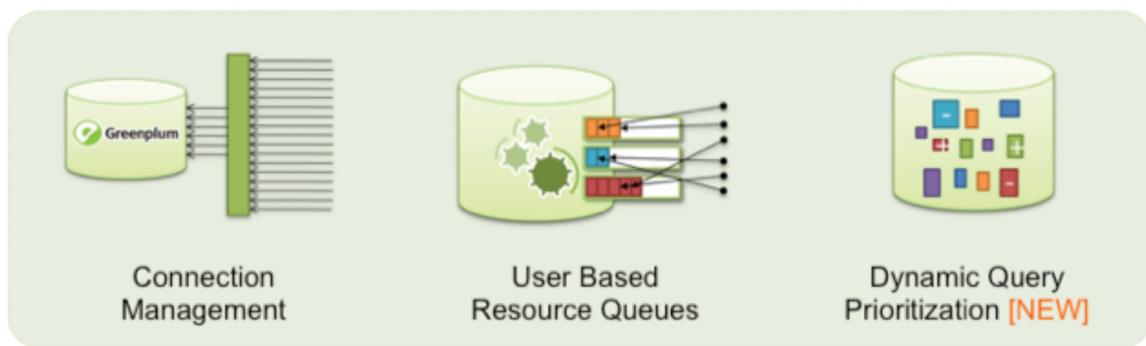
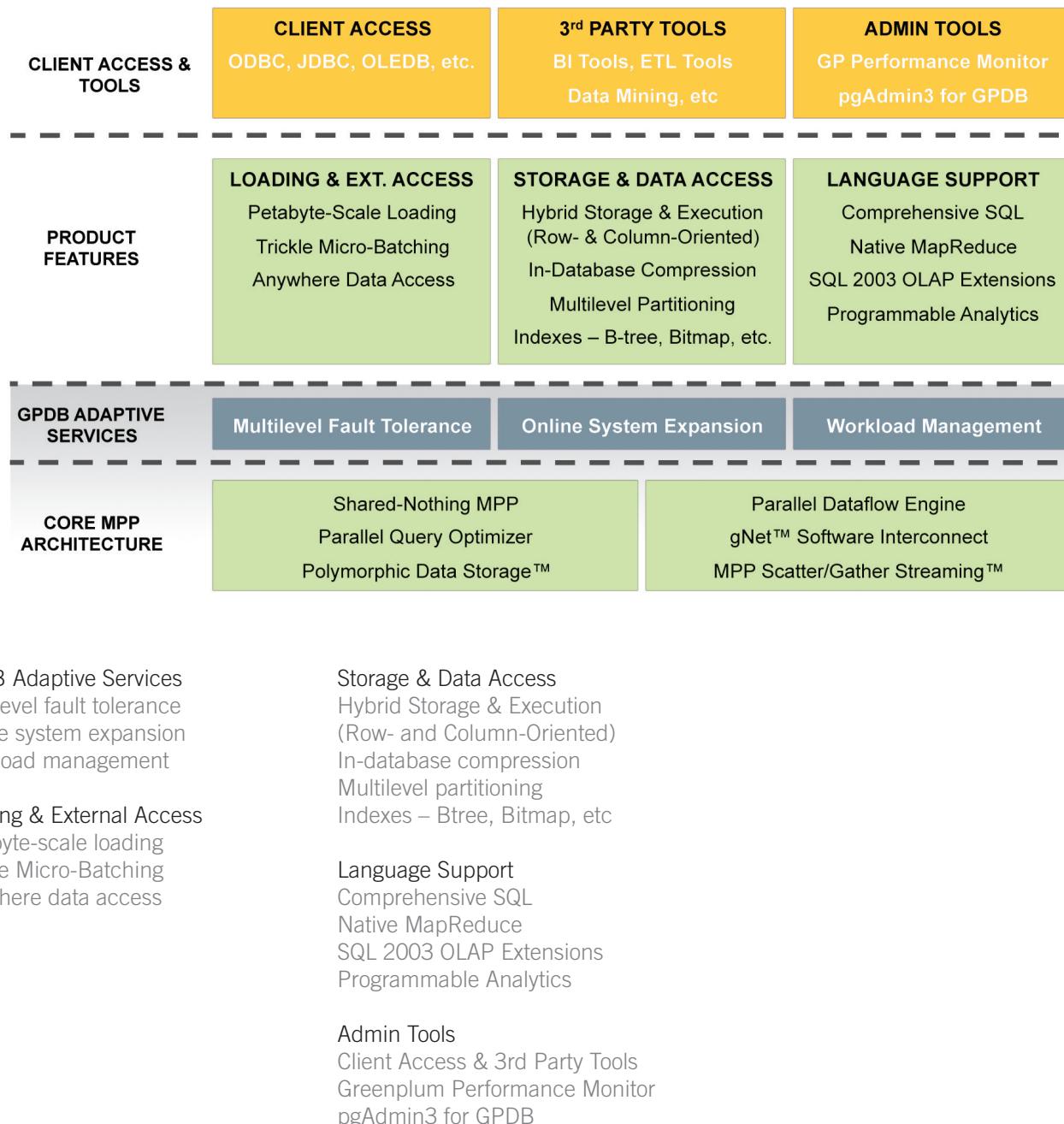


FIGURE 14. GREENPLUM DATABASE PROVIDES FLEXIBLE CONTROLS FOR ADMINISTRATORS

Key Features and Benefits of Greenplum Database



Core MPP Architecture

The Greenplum Database architecture provides automatic parallelization of data and queries—all data is automatically partitioned across all nodes of the system, and queries are planned and executed using all nodes working together in a highly coordinated fashion.

Multilevel Fault Tolerance

Greenplum Database utilizes multiple levels of fault tolerance and redundancy that allow it to automatically continue operation in the face of hardware or software failures.

Online System Expansion

Enables you to add servers to increase storage capacity, processing performance, and loading performance. The database can remain online and fully available while the expansion process takes place in the background. Performance and capacity increases linearly as servers are added.

Workload Management

Provides administrative control over system resources and their allocation to queries. Provides user-based resource queues that automatically manage the inflow of work to the databases, and provides dynamic query prioritization that allows control of runtime query prioritization and ensures that all queries get their appropriate share of the system resources.

Petabyte-Scale Loading

High-performance loading and unloading utilizing MPP Scatter/Gather Streaming technology. Loading (parallel data ingest) and unloading (parallel data output) speeds scale linearly with each additional node.

Trickle Micro-Batching

When loading a continuous stream, trickle micro-batching allows data to be loaded at frequent intervals (e.g., every five minutes) while maintaining extremely high data ingest rates.

Anywhere Data Access

Allows queries to be executed from the database against external data sources, returning data in parallel, regardless of their location, format, or storage medium.

Hybrid Storage and Execution (Row and Column Oriented)

For each table (or partition of a table), the DBA can select the storage, execution, and compression settings that suit the way that table will be accessed. This feature includes the choice of row- or column-oriented storage and processing for any table or partition. Leverages Greenplum's Polymorphic Data Storage technology.

In-Database Compression

Utilizes industry-leading compression technology to increase performance and dramatically reduce the space required to store data. Customers can expect to see a 3x to 10x disk space reduction with a corresponding increase in effective I/O performance.

Multi-level Partitioning

Allows flexible partitioning of tables based on date, range, or value. Partitioning is specified using DDL and allows an arbitrary number of levels. The query optimizer will automatically prune unneeded partitions from the query plan.

Indexes—B-tree, Bitmap, etc.

Greenplum supports a range of index types, including B-tree and bitmap.

Comprehensive SQL

Offers comprehensive SQL-92 and SQL-99 support with SQL 2003 OLAP extensions. All queries are parallelized and executed across the entire system.

Native MapReduce

MapReduce has been proven as a technique for high-scale data analysis by Internet leaders such as Google and Yahoo. Greenplum Database natively runs MapReduce programs within its parallel engine. Languages supported include Java, C, Perl, Python, and R.

SQL 2003 OLAP Extensions

Provides a fully parallelized implementation of SQL recently added OLAP extensions. Includes full standard support, including window functions, rollup, cube, and a wide range of other expressive functionality.

Programmable Analytics

Offers a new level of parallel analysis capabilities for mathematicians and statisticians, with support for R, linear algebra, and machine learning primitives. Also provides extensibility for functions written in Java, C, Perl, or Python.

Client Access and Third-Party Tools

Offers a new level of parallel analysis capabilities for mathematicians and statisticians, with support for R, linear algebra, and machine learning primitives.

Health monitoring and alerting

Provides email and SNMP notification in the case of any event needing IT attention.

Greenplum Performance Monitor

Enables you to view the performance of your Greenplum Database system, including system metrics and query details. The dashboard view allows you to monitor system utilization during query runs, and you can drill down into a query's detail and plan to understand its performance.

pgAdmin 3 for GPDB

pgAdmin 3 is the most popular and feature-rich Open Source administration and development platform for PostgreSQL. Greenplum Database 4 ships with an enhanced version of pgAdmin 3 that has been extended to work with Greenplum Database and provides full support for Greenplum-specific capabilities.

ABOUT GREENPLUM AND THE DATA COMPUTING PRODUCTS DIVISION OF EMC EMC's new Data Computing Products Division is driving the future of data warehousing and analytics with breakthrough products including Greenplum Database, Greenplum Database Single-Node Edition, and Greenplum Chorus—the industry's first Enterprise Data Cloud platform. The division's products embody the power of open systems, cloud computing, virtualization, and social collaboration—enabling global organizations to gain greater insight and value from their data than ever before possible.