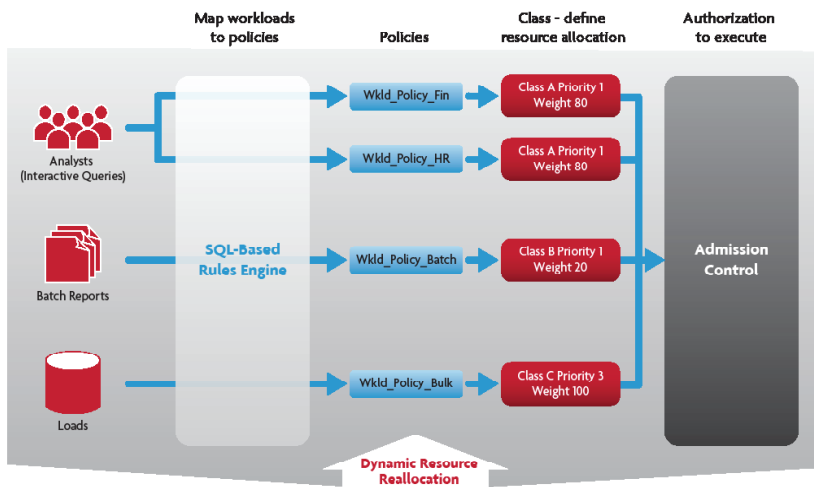


Aster Data nCluster 4.5: Dynamic Workload Management

The data warehouse has become a complex environment as companies rely increasingly on data-driven applications and ultra-fast analytics across all sectors of the organization. Along with traditional back-office decision support and business intelligence, data warehouses are now expected to serve a broad range of applications and users in their daily business operations. These “big data” deployments need to be well-managed to optimize performance and maximize hardware investment—in particular to avoid replicating systems just to manage different workloads.

Today’s data warehouse may be processing hundreds of interactive queries submitted by call center users who need to complete within a few seconds. At the same time, the system may be performing much more complex ad hoc queries used by business analysts examining terabytes of data for deeply buried patterns and trends. Meanwhile other essential workloads, such as batch or trickle feed loads and data backups, are running in the background. All these diverse workloads are contending for resources (CPU, disk, memory, and network), and the system has to make the right choices instantly to meet increasingly stringent service level requirements and user expectations.

Aster Data’s Dynamic Workload Management provides capabilities that ensure predictable, high performance in the high-concurrency, mixed workload environments of data warehousing and big data applications. Administrators can define workloads with fine-grained precision and set priority and resource allocation. The system uses these rules to optimize resource utilization, both for admission control for new queries, as well as dynamically while a query is in flight. Users across the organization get the insights and response times they need; you get an extremely cost-effective system built for today’s mixed-workload, big-data environments.



Granular rules-based prioritization, Workloads can be managed pre- and post- admission, Dynamic resource allocation and re-allocation

Figure 1: Aster Data’s Dynamic Workload Management provides high user and application concurrency and predictable service levels.

Quick Overview

Aster Data’s Dynamic Workload Management ensures highly predictable performance and guaranteed service levels for the complex mixed workloads of an enterprise data warehouse and analytically-intensive applications. Intuitive, fine-grained policy controls allow administrators to define and manage diverse workloads to meet the organization’s business priorities. Aster Data provides the first-ever dynamic workload management capability for a massively parallel processing (MPP) system that runs on commodity hardware. The ability to dynamically reallocate CPU and storage resources based on in-progress transactions ensures time-critical queries can be processed immediately.

Highlights

- Both pre- and post-admission control with dynamic resource re-allocation for optimum performance with mixed workloads
- Scalability to support multiple workloads running in parallel across commodity servers
- Rich set of attributes allows workload management rules to be as general or as specific as you need
- Controls both CPU and I/O resource consumption across the whole system for effective SLA enforcement
- Flexibility to quickly specify priority and resource usage for special projects
- Easy to understand and use, with familiar SQL tables and expressions that administrators use every day

“Aster Data’s mixed-workload management enables large numbers of users to run simultaneous interactive queries with little or no latency—a key enabler in delivering real-time operational analytics.”

Dan Vesset, Vice President, Business Analytics
IDC



Dynamic Workload Management combines rich control with administrative simplicity for managing complex environments with many applications, users, and multiple constituencies, all relying on the data warehouse and accessing it constantly. This is a key feature of Aster Data's nCluster 4.0 massively parallel data-analytics server that runs on commodity hardware. The result is phenomenal performance gains for critical applications, and enterprise-wide business intelligence that provides a competitive advantage.

Workload Definition: Powerful Rules Framework

Aster Data's Dynamic Workload Management provides control, flexibility, and simplicity for managing a mixed-workload environment, and makes sure service level goals are met. Using familiar SQL tables and expressions, administrators can define workload priorities and resource allocation in many different ways: by user, role, time, location, application, and other key attributes, used separately or combined.

These workloads can be defined with the level of precision appropriate for your environment. For example, you could define workloads for different branches or departments, and then create more fine-grained workloads within that category, such as by user role or application, each with its own priority and resource weight. Or you could simply create a few very specific, high-priority workloads, with the remainder as a single, lower-priority default workload.

When a query statement is executed, it maps to a defined workload (and if the statement doesn't match a specific workload, it executes as a default workload). Each workload in turn maps to a service class that specifies the priority and resource weight used by the system to execute that workload. Thus, workloads are managed using two SQL tables: the Workload table and the Service Class table.

The system collects information about each activity in the system—from every SQL statement executed to complete physical backup operations—and exposes it to the user through a rich set of configuration variables that can be used in the workload definitions to enable a rich set of use case possibilities. For example, workloads may be:

- **User-Based** – Executives, developers, business analysts, power-users, executives, and more—different roles can have priority over others. For example, you could define a single workload to prioritize the work of the whole Sales department or specify multiple rules to isolate specific user names.
- **Time-Based** – For example, to implement different policies for business hours and overnight. Loads could be higher priority at night when executed in batches and have a lower priority during the day to minimize the impact on operational workloads.
- **Application-Based** – For example, to specify an interactive application as high priority, or set a maximum weight for a resource-intensive application.
- **Object-Based** – For example, define workloads based on database and table names, such as using a high-priority workload for queries issued against the summary table used by an important reporting application.
- **Task-Based** - Different priorities could be used for read-only workloads and specific operations such as DDL and DML statements.
- **Location-Based** – In a scenario where different branches access a single system from different subnets, statements issued by clients at a specific branch could be given higher priority using a single rule based on the IP addresses used.
- **Reprioritized Dynamically** – Even a single SQL statement can change priorities over time. For example, your rules can ensure high resource allocation for a newly added query of a given type but throttle down resources for that query if it runs so long that it is suspected of being a runaway query. Both CPU share and disk I/O share are adjusted to enforce priorities.

Variables can also be readily combined to create precise and powerful workload rules. As an example, although John Doe's batch data mining tasks are usually processed as low priority, he has an urgent project to finish tonight and asks you, the system administrator, to make an exception. You can add a temporary rule that specifies a high-priority workload for "every SELECT statement issued by John when connecting from home between 8PM and 12AM." Before and after this time window, statements from John execute as his regular low-priority workload.

Rich Activity Set for Workload Rules	
dbName (varchar)	E.g.: "select statements issued by Daniel between 9AM and 12PM when connecting from home to the raw database"
userName (varchar)	
roles (varchar[])	
clientIpAddr (inet)	
connTime (timestamp)	
currentTime (timestamp)	
stmtElapsedTime (interval), stmtStartTime (timestamp)	
stmtType (varchar)	
tableNames (varchar[])	
txElapsedTime (interval), txStartTime (timestamp)	
activity (varchar)	

Table 1: Rich activity set for workload rules.

Resource Optimization on the Fly

Dynamic Workload Management is used for admission control before execution begins and while queries are in flight. New queries execute according to the priorities and weights defined for that workload. (If system usage is light, all queries execute with high resource utilization.) The system then optimizes resource utilization dynamically as queries are running, in order to ensure service level goals continue to be met as demands on the system change.

For example, at 8:00 a.m. an analyst starts a long-running ad hoc query joining large tables. This is normally a lower-priority workload, but since system activity is light at that time, resource usage is below the threshold, and execution begins at a high resource level. At 9:00 a.m., a team of 100 customer care representatives report to work and start conducting high-priority interactive queries, which normally would be delayed by the ongoing ad hoc query. The Dynamic Resource Manager automatically reallocates resources from the ad hoc query to the interactive queries to prevent bottlenecks and ensure fast response. The lower priority ad hoc query continues running while consuming fewer resources, without impacting the interactive queries.

High Concurrency with Predictable Response Time

To illustrate how Dynamic Workload Management enables highly predictable running times, a representative test scenario was developed consisting of high-priority interactive queries together with low priority ad hoc queries and data loads. The two charts below highlight the impact of Dynamic Workload Manager in this typical mixed-workload environment. Without mixed-workload management, the ad hoc queries and bulk loads impact the performance of higher-priority interactive queries, causing the system to slow. In contrast, with mixed-workload management, the system prioritizes interactive queries over the two other workloads, and average running time is not significantly affected.

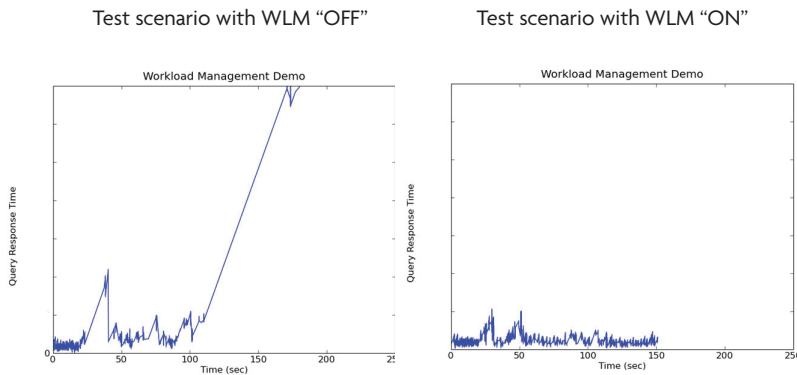


Figure 2: Dynamic Workload Management Maintains Predictable Response Times even as Mixed Workloads Increase

Workload Management for “Big Data” Challenges

Organizations are more reliant on data than ever as competitive differentiation. More data-intensive applications, as well as more users, across more departments, running more applications, now take advantage of the data warehouse for data processing and analytic capabilities of all kinds. These diverse workloads, all contending for resources, place extraordinary demands on an underlying information platform. Highly efficient workload management is even more important as data volumes grow, windows of opportunity shrink, and applications become more complex.

Aster Data’s Dynamic Workload Management provides innovative capabilities that ensure predictable high performance for the full range of applications running concurrently in an enterprise. Rather than costly replications, now you can run all of your workloads on a single system where big data resides. Administrators get the simplicity, flexibility, and finely-grained precision they need to set priorities and allocate resources to match the requirements of the most demanding mixed-workload environments. It’s one more way nCluster 4.0 provides a uniquely powerful, massively parallel platform, allowing you to run your analytically-intensive applications with unprecedented performance and cost-effectiveness.

About Aster Data

Aster Data is a proven leader in big data management and big data analysis for data-driven applications. Aster Data’s nCluster is the first MPP data warehouse architecture that allows applications to be fully embedded within the database engine to enable ultra-fast, deep analysis of massive data sets. Aster Data’s unique “applications-within™” approach allows application logic to exist and execute with the data itself. Termed a “Data-Analytics Server”, Aster Data’s solution effectively utilizes Aster Data’s patent-pending SQL-MapReduce together with parallelized data processing and applications to address the big data challenge. Companies using Aster Data include Coremetrics, MySpace, comScore, Akamai, Full Tilt Poker, and ShareThis. Aster Data is headquartered in San Carlos, California and is backed by Sequoia Capital, JAFCO Ventures, IVP, and Cambrian Ventures, as well as industry visionaries including David Cheriton, Ron Conway, and Rajeev Motwani. For more information please visit www.asterdata.com, or call 1.888.Aster.Data.