

# Eliminate traffic jams

Dynamic workload management keeps your data moving. *by Dan Graham*

Think about taking a taxi around Paris at 6 a.m. Quiet, cool and uncrowded, the city is magnificent and calm, and traffic is light. But by 8 a.m., the crowds, taxis, trucks and ambulances turn the streets into an all-day traffic jam. Your taxi goes only as fast as the car in front of you. Not until 8 p.m. does the congestion dissipate.

Now imagine Paris without stop signs or traffic lights. Then, eliminate the “drive on the right side of the street” rule. Traffic would gridlock. Embouteillage!

This is similar to the daily workloads that pass through an enterprise data warehouse (EDW). Reports (cars), tactical queries (motorcycles and bikes), executive queries (ambulances), data mining (buses) and data loading (trucks) can simultaneously clog the system. With Teradata Optimizer and parallel server, most customers get great performance, unaware their environment is in constant chaos. Yet, like Paris, the EDW can operate better if the traffic flow is organized.

## Rules of the road

Organizing workloads into categories and prioritizing resource allocation is nothing new. The concept originated in the 1960s to support real-time telemetry for the space program. Later, IBM’s Multiple Virtual Storage (MVS) operating system (OS) applied mixed workload management on the mainframe to keep batch jobs from devastating the performance of its Customer Information Control System (CICS) online transaction processing (OLTP) program.

UNIX and Windows servers sidestepped mixing workloads altogether, proclaiming that hardware was so cheap every workload could run on its own server. But over time, hundreds of servers created excessive cost, so today’s vendors offer virtualization software to consolidate UNIX and Windows servers into one large server. Hence, the workloads were eventually mixed anyway.

However, an EDW mixed workload is more granular than managing multiple

applications. Functioning at the user or query level, an EDW must optimize a mix of ad hoc and tactical queries, online analytical processing reports, data mining, real-time data loads, batch loads, visualization queries, database maintenance—and anything else the users want.

Teradata’s mixed workload management began in 1989 in response to customer needs. (See table 1.) Initially, fair-share scheduling was deployed to ensure that no long-running query would starve for the lack of CPU time. This evolved into priority scheduling, allowing DBAs to assign resource partitions and up to 200 different priorities (fast lanes, slow lanes). Active data warehouses, enabled by new releases of Priority Scheduler (traffic cops), emerged around 2000. This drove demand for more control, so Teradata released Query Manager to apply filters and throttles (stop signs and traffic lights) to out-of-control queries and users.

Then came Teradata Active System Management, a new plateau of optimization and control, to support active data warehousing. Dynamic regulation of workload groups allowed the entire system to be optimized (citywide traffic authority). It also gave DBAs more control to ensure tactical queries (motorcycles and bikes) are consistently fast, regardless of concurrent traffic. The controls include workload classification, workload throttles, exception monitoring and exception handling. Recently, Teradata 12.0 introduced automation of responses to system-wide events, further reducing the stress on DBAs managing the system.

**Table 1: Timeline of Teradata workload management system**

Year	Type	Solution
1989	Request-based	Fair-share scheduling
1997		Priority Scheduler
2000		TDQM-Query Manager
2002	Workload-based	Teradata Active System Management
2005		
2007	Events and workloads	

Teradata has enhanced Priority Scheduler and workload management continuously since 1989. Note that the product names in the “Solutions” column are from the corresponding time periods.

## Assigning traffic lanes and speed limits

The first step in managing fine-grained workloads through Teradata is to organize them by user community into Priority Scheduler allocation groups (AGs). One or more user communities, such as finance or call center, are assigned to an AG. When users submit queries or load jobs, the work submitted is mapped into these AGs, which are assigned a relative weight in the system. (See table 2.) A typical Teradata site will have five to 10 AGs supporting user requests.

Teradata Active System Management helps prevent runaway queries or system overloads generated by incoming work. Using filters and throttles, DBAs can apply rules such as:

- > Allow only five concurrent requests from `MARKETING_Group`
- > Limit answer sets to less than 1 million rows on Mondays from 9 a.m. to noon
- > Prevent full-table scans at month end from 8 a.m. to 3 p.m.
- > Permit `USER_Dave` no more than three concurrent queries
- > Restrict `SALES_Group` to four concurrent queries from 9 a.m. to 4 p.m. daily
- > Limit FastLoad and MultiLoad to four jobs on Mondays from 8 a.m. to 5 p.m.

Several vendors and tools suppliers offer the capability to organize EDW tasks into queues, staging them into the system and applying filters. Organizing vehicles trying to get into Paris is useful but not good enough. We must control traffic inside the city that's taking up space (CPU) and producing congestion (blocking others). Likewise, we don't want to give top priority to a billion row table-join. To manage query elapsed times, we need throttles and dynamic priority controls inside the relational database management system (RDBMS).

## Determining the right of way

Teradata's Priority Scheduler operates independently on each node in the configuration. It puts AMP and parsing engine

(PE) requests for CPU time into OS dispatch queues based on the AG weights and other factors. The OS then assigns an available CPU to the task with the highest priority (right of way) first. When the highest-priority queue is empty, tasks in the next highest-priority queue are given CPU time.

You might think a priority weight of 50% means your group's tasks get 50% of the resources. Not so. Priority Scheduler uses a sophisticated method of calculating a relative priority based on active AGs in the node. An AG is considered active as long as tasks are

assigned to that AG, or if CPU usage occurred in that AG in the time period immediately prior (61 seconds is the default). Using DBA-assigned weights, anti-starvation logic and recent history of the AG, Priority Scheduler computes a relative priority to place tasks into an OS priority queue.

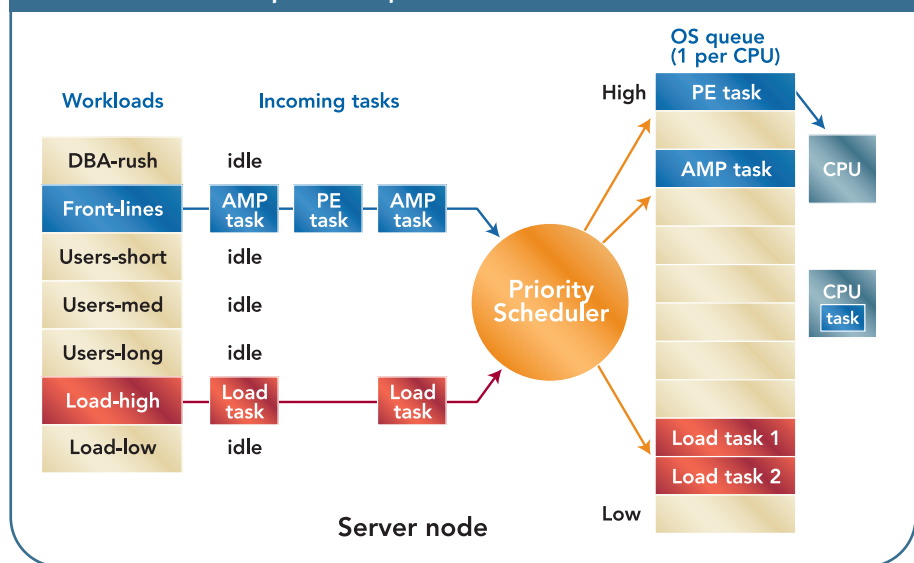
This is profound. In the example in figure 1, some AGs, like `DBA-rush`, are inactive and would not be included in priority calculations. This means the load tasks would get more frequent CPU access than if all AGs were active. Furthermore, if

Table 2: Example of allocation group weightings

Workload	Daytime weight %	Nighttime weight %	Description
DBA-rush	10	10	Specific user names
Front-lines	50	35	Call center applications
Users-short	12	8	Quick reports
Users-med	8	7	Medium ad hoc reports
Users-long	2	7	Long-running reports
Load-high	8	19	GoldenGate, TPump
Load-low	4	8	Month-end jobs
Internal-work	6	6	Internal database tasks

Teradata Active System Management enables users to set priorities for tasks based on day and type of work.

Figure 1 CPU dispatch queues



Teradata Priority Scheduler organizes tasks into priority queues for each CPU.

no users are active in the system, the load-high task will be offered up to 100% of the CPU resources because it's the only task in the OS queues.

In static priority management systems, the load-high tasks are always limited to 8% of the CPU resources no matter what else is executing. This means the load job runs slower and the node is poorly utilized. But with Priority Scheduler, the load job can use 100% of the CPU available until a higher-priority task arrives.

### May I cut in front of you?

Priority Scheduler dynamically interacts with the OS in other ways as well. As a task is dispatched on the system, a CPU quantum clock starts counting down. When the clock reaches zero, the current task is interrupted so others can use the CPU. This context switch prevents any one task from hogging the CPU, ensuring all tasks in the same AG get a fair share of CPU time.

OSs used by Teradata servers also force a task off the CPU if a higher-priority task is ready to run. This pre-emptive scheduling allows front-line tactical queries to cut in

front of, for example, load-high tasks to run a call center request. The pre-emption method and quantum-clock quanta vary by OS.

While most modern OSs use quantum clocks and pre-emption, the systems cannot associate these with business objectives. Also, most OSs are coarse-grained, meaning they give all tasks within the RDBMS the same priority. With Teradata, the DBA can assign users and workloads with weights that map to CPU priorities for every PE or AMP worker task. These three capabilities (quantum clock, priority scheduling and pre-emption) interact to get the best use from a Teradata server.

### Reducing traffic congestion

In the performance analysis graphic (see figure 2), a Teradata client monitored system response time at consistent intervals. The blue line shows total CPU burn rate while the red line shows the sum of query elapsed times. On Feb. 5 and March 19, the amount of work moving through the system was nearly identical (blue markers). But on March 18 the client turned on Teradata Active System Management, which includes

Priority Scheduler. Congestion in the system was reduced and total elapsed time dropped 48%. While the system seemed faster to end users, what happened was the query traffic was prioritized and optimized. The server was just being used better. Of course, performance results will vary.

All of this sophistication is invisible to the users. In fact, there is little for the DBA to do besides set up AGs and throttles. This ensures high-priority tasks have numerous ways to be first in line for CPU resources. The executive query (ambulance) or tactical query (motorcycle or bike) can reliably zip through traffic. Furthermore, Teradata Active System Management enables maximum throughput from every node. TPump can insert rows on nodes 4 and 6 while tactical queries fetch rows using single AMP requests on nodes 5 and 6. Simultaneously, massive complex joins are running on all the nodes.

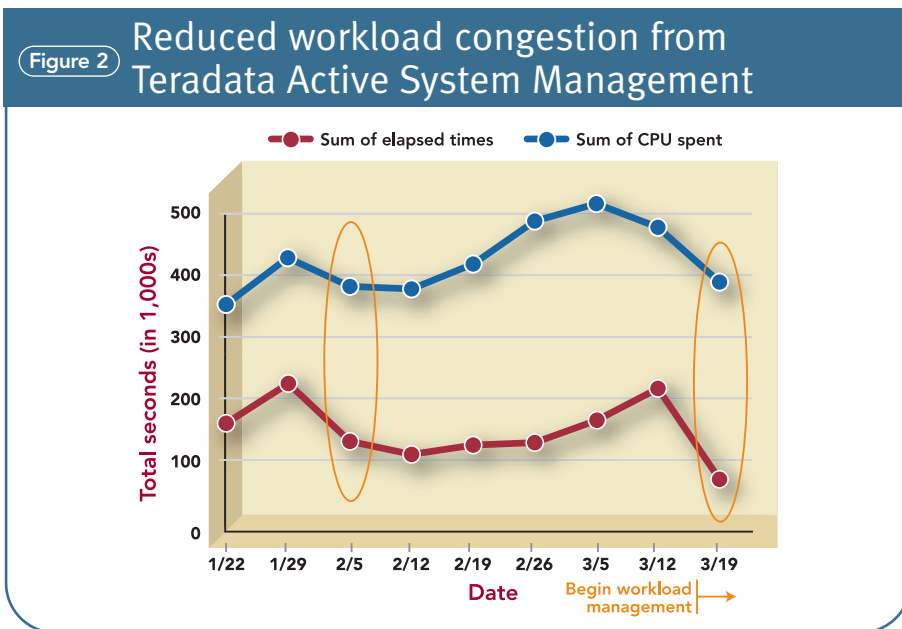
Mixed workload management provides several benefits to users:

- > Eliminates or delays unnecessary upgrades by better node utilization
- > Provides more consistent elapsed times
- > Helps meet service level agreements
- > Enables active data warehousing by ensuring tactical queries can coexist with reporting and data loading
- > Facilitates operational data store consolidation into an active data warehouse

By proactively and consistently using Priority Scheduler, Teradata customers can get better throughput and improved query elapsed times—even on systems not heavily burdened. Although many customers have been successful on their own, Teradata Professional Services can help set up the system.

It's too bad Teradata Professional Services and Priority Scheduler can't do their magic for Paris traffic! **T**

*Dan Graham has more than 35 years in IT and leads Teradata's Active Data Warehouse Technical Marketing.*



Teradata Active System Management reduces server congestion, producing better elapsed times for many queries.