Using Platform LSF To Harness Non-Dedicated Computational Resources for LS-DYNA Crash Simulations at DaimlerChrysler

Dale Dunlap and Joseph Cieslak P.E.
Platform Computing

John Picklo
DaimlerChrysler Corporation

Abstract

Computer aided engineering (CAE) tools are an important part of the product development lifecycle. These tools are used to perform complex simulation and analyses during the design phase. CAE requires significant computational resources in order to meet the response time requirements of the design engineers that use these applications. The speed and volume of CAE work provides a competitive advantage to the manufacturer by helping to bring products to market faster, reducing the need to build costly prototypes, and by increasing the quality of the end product.

This paper documents a comprehensive approach to extending the HPC cluster grid to DaimlerChrysler's engineering workstations. This approach will allow DCX to transparently make use of idle workstation CPU cycles and thereby significantly increase overall computing power at a fraction of the cost of a comparable dedicated HPC solution. This solution will also provide the foundation for further grid deployment to the DCX infrastructure so that additional benefits can be achieved in the future.

By collectively harnessing the latent power of existing resources, Platform and DCX feel it maximizes the value of assets already owned while it gains compute power to accelerate and refine research and analysis, without any impact on the daily usage of these workstation users. The typical utilization of a desktop in the enterprise is about 5-7%, and large enterprises have tens of thousands of desktops. This will provide a cost effective path for increasing computing power while avoiding additional procurements of HPC cpu's during usage spikes.

Background

DaimlerChrysler (DCX) is a leading user of CAE tools. DCX has successfully been using Platform Computing's LSF product for several years to "grid enable" its High Performance Computing (HPC) clusters in the NAFTA region that run CAE toolsets such as LS-DYNA. It has been documented that LSF optimizes CPU utilization on HPC clusters to over 90% efficiency by minimizing job execution latencies. Even though LSF is being used effectively on the DCX HPC clusters, there was an opportunity to extend the HPC cluster grid by making use of idle CPU cycles in approximately 1,500 UNIX workstations deployed to DCX engineers.

DaimlerChrysler has been using LSF for several years. Currently, two production clusters are working separately. One of the production clusters, used primarily for LS-DYNA crash simulations, has more than 1,000 CPUs running 24x7. This cluster serves hundreds of engineers and is very important to DCX's business. DCX also has over 1,500 UNIX desktops, mostly used for the CAD application CATIA. Because CATIA is a very CPU and memory intensive application principally used by designers during the day, it was thought that these workstations could be used as a potential computing resource for LS-DYNA during night and weekend periods of user inactivity. DCX wanted however to make use of the HPC cluster initially, and "steal" idle computing cycle of those desktops to add to the capacity of the HPC cluster. To make sure computational commonality was maintained, the LS-DYNA jobs on the workstations had to be run in a 12 cpu per job configuration which is the same processor requirements used in its HPC environment.

Why Platform Computing

Platform Computing has over fourteen years of experience in helping companies to more effectively gain visibility and utilization of their IT investment. We have a stable history of delivering Grid solutions to more than 1,600 customers with around the clock global support. Platform has the most comprehensive Grid product suite in the industry, backed by a robust methodology to deliver results quickly and efficiently. No other company can match Platform's Grid lifecycle solutions and implementation experience. Our track record for delivering stable products reduces the risks related to deploying a Grid solution. Our products have been proven to offer an effective cost of ownership due to a reduced number of failures, lower administrative costs, and our superior ability to scale to manage large workloads. Platform is committed to emerging Grid standards, and is a major participant in the Global Grid Forum. Our goal is to support the OGSA and OGSI standards. This reduces the interoperability risks for working with other standards compliant Grid implementations within DCX.

DaimlerChrysler Requirements

Platform and DaimlerChrysler identified the following requirements to enable UNIX desktop asset optimization:

1. Exploit desktops during configurable time window without affecting users.

   Jobs will be running for several hours, consuming a lot of resources. This could slow down the machine, so users don't want jobs running on their desktops while they are working. Desktops should run jobs ONLY when they are absolutely idle. The solution should allow the administrator customize time window to control "cycle-stealing" activity and minimize the impact on desktop users.

2. Only desktops satisfying "node selection criteria" will become candidates.

Desktops satisfying pre-defined criteria will be considered to join a cluster. This node selection needs to be automatic, and these criteria must be configurable by the administrator. The known criteria are:

- Minimum 18 G of local disk available.
- "Catia" process should not be running
- No user logged into the console, or at least half an hour keyboard/mouse idle time.

3. The "best" desktop candidates to be used.

Initially, DCX wants some of the available desktops to be used for building a "new cluster" from 1,500 potential candidates pool every night. The selection needs to be automatic, for it is hard for the administrator to do manually. The "best" selection will be configurable, possibly based on CPU factor and/or memory of desktops. The size of "new cluster" was required to be configurable as well, either based on configured number or based on workload.

4. Jobs pending in HPC cluster can use desktops during the time window.

Jobs can be dispatched to desktops during the time window, based on their run-time requirements.
- Users can monitor job status running on desktops
- Users can get job result back after job is done
- Automatic checkpointing of unfinished jobs when time window closes, and requeueing them back to HPC cluster at top of the queue.
- Cleanup jobs on desktops

5. Job characteristics and run-time requirements for LS-DYNA:

- 12-cpu MPI parallel job.
- Average runtime more than 10 hours.
- Job must be run on the same type of machines.
- The job must be application-level checkpointable. (output and checkpoint files saved on shared file system)
- Job should migrate gracefully if one of the machines goes down.
- Jobs will be checkpointed and go back to pending in the morning.

6. The production cluster (HPC) is quite complicated and mission-critical. The solution must have no interruption to their production workload.

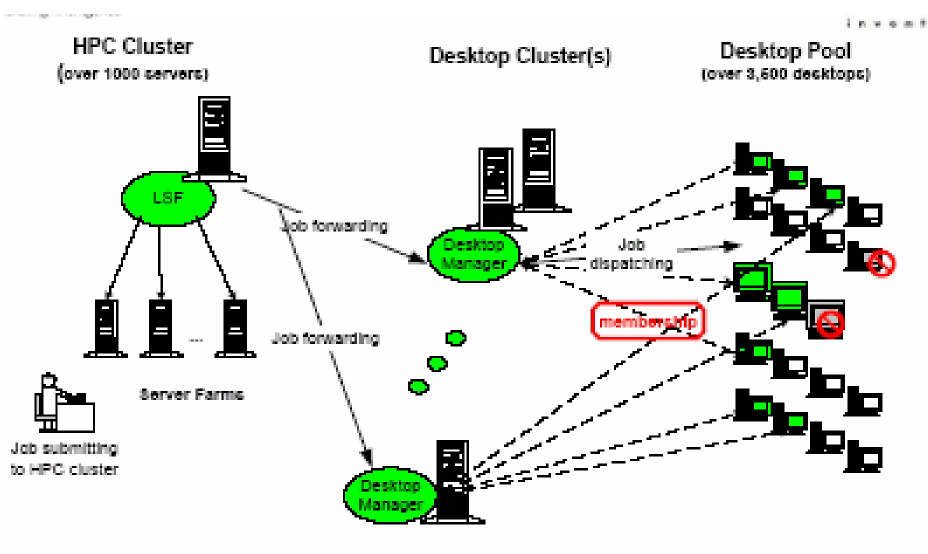


Figure 1.1 – UNIX Desktop ActiveCluster system organization

Extra desktop resources are organized in one or several "desktop" clusters that are managed by each "desktop manager". The desktop cluster accepts and executes jobs from HPC cluster

ONLY. This multiple desktop clusters and managers design allows for handling a huge desktop pool in the future. The next picture shows the architecture and flow of the solution.

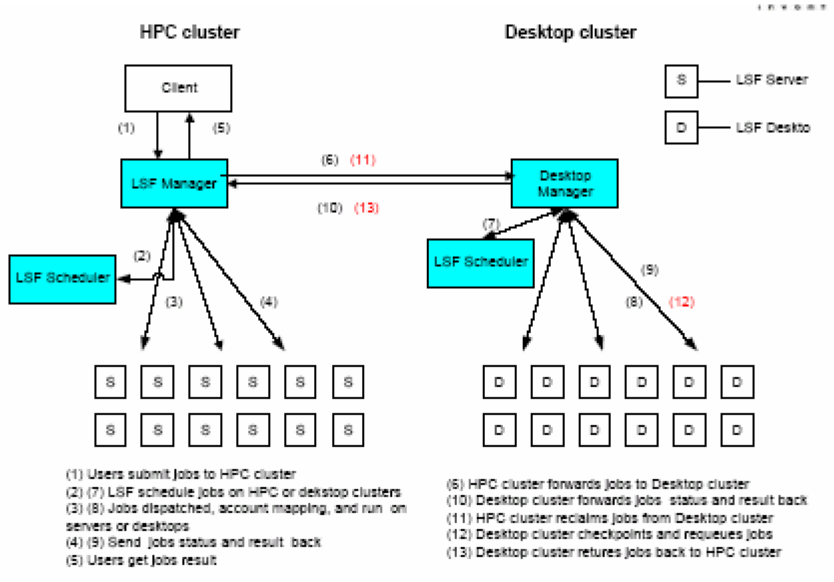
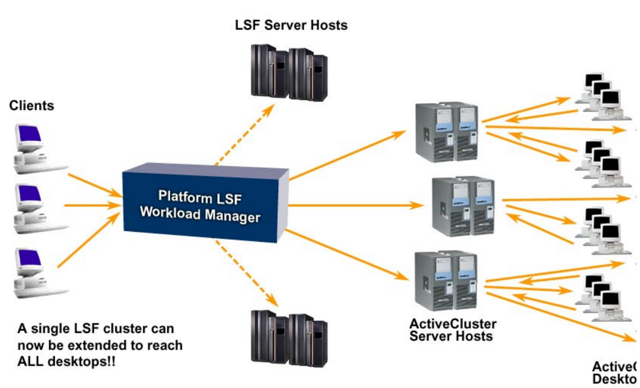


Figure 1.2 – UNIX Desktop ActiveCluster system architecture

Solution: Platform LSF Desktop

Platform LSF Desktop (formerly known as Platform LSF ActiveCluster) is a software product for managing and accelerating batch workload processing for compute-and data-intensive applications by harnessing the untapped processing power of desktop-based computers (including workstations, desktops, and laptops) across the enterprise. The computational power of most desktops is wasted approximately 90 percent of the time, so by recycling this wasted resource, you can increase your organization's processing power without further hardware investment. As an enterprise computing solution, it enables organizations to utilize valuable computing resources available in existing Windows-based resources to process compute-intensive tasks efficiently, complete workloads faster, and increase user productivity. Platform LSF intelligently schedules and guarantees completion of batch workload across a distributed or virtualized, IT environment.



Benefits to the Enterprise

- Platform LSF - transforms enterprise-wide networks of Windows-based resources into a virtual supercomputer that deliver massive processing scalability and breakthrough levels of computational capability.
- Organizations can achieve higher quality results by exhaustively simulating or evaluating all parameters in computational models. Greater levels of productivity allow for faster iteration of analyses and greatly reduce time to market.

Platform LSF extends an organization's processing reach to the unused cycles available in existing Windows-based resources. This significantly boosts processing power, without investing in additional hardware.

Features	Benefits
Advanced Network Scheduling	Schedules and allocates workload based on memory and idle CPU capacity, disk space, temp space, software availability, and user-defined resource restrictions
Prioritizing and Optimizing Workloads	Executes jobs as continuous background processes, taking advantage of any unused computing cycles and applying maximum computing power to high priority tasks
Reliable Job Execution	Has no single point of failure as it automatically reschedules jobs to combat any problems during execution
Network Friendly Distributed Computing	Is designed and easily configured to match enterprise network architectures, providing automatic compression, caching and staging technology to ensure minimal network traffic
Secure Information	Leverages native Windows security features to ensure the confidentiality of both desktop and distributed data and applications

How the Solution Works

During office time, Stage 1, the behavior of HPC cluster is exactly same as the current environment. Users submit jobs to HPC cluster, and scheduler will dispatch jobs to computing servers in HPC cluster. No job will be running on any desktops to trouble the desktop users. The desktop cluster is in “standby mode”, and only the desktop manager host (or some desktop manager candidates for system fail-over) is in the cluster

Stage 1) Office time (out of the desktop membership time window)

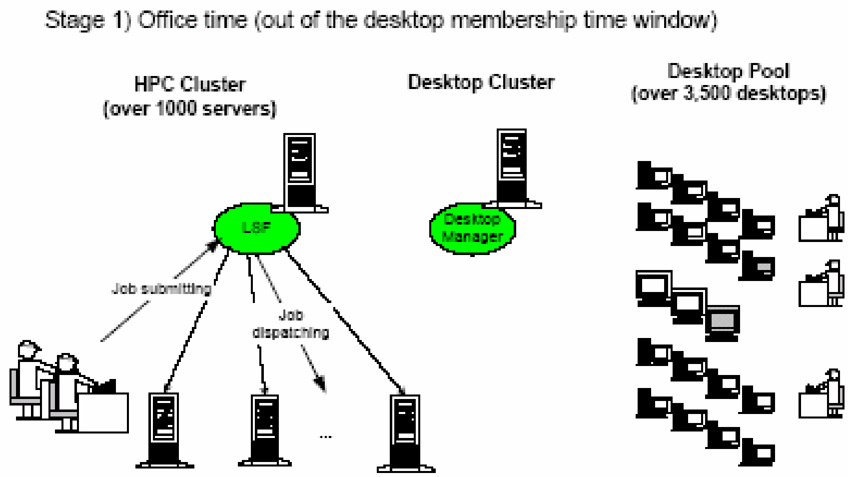


Figure 2 – System overview during *office time*

When entering desktop membership time window, Stage 2, the desktop cluster will expand automatically. Eligible desktops join the cluster. The cluster is ready for accepting jobs from HPC cluster. The scheduler of HPC cluster will forward pending jobs to the desktop cluster for extra computing resources. The scheduler of the desktop cluster then dispatches them to desktops following job requirements and scheduling policies defined.

Stage 2) After-hours time (Within the desktop membership time window)

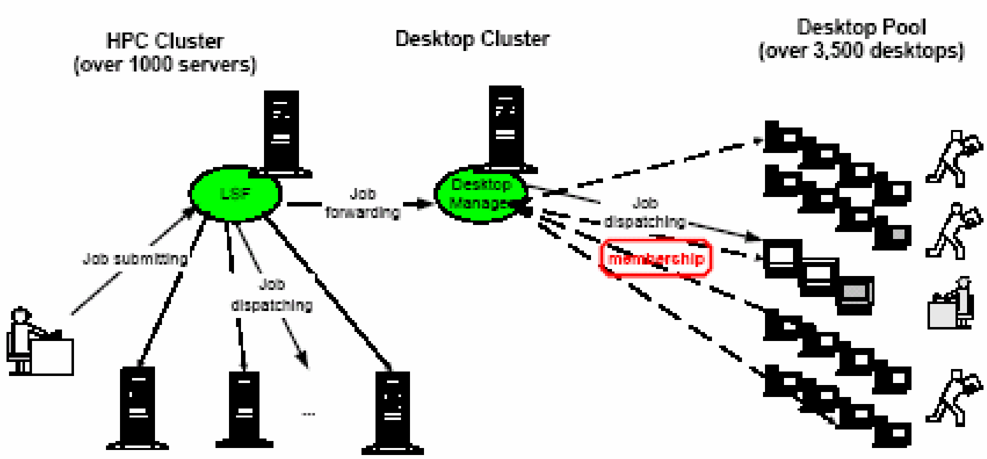


Figure 3 – System overview during *after-hours time*

At the end of the desktop membership time window, HPC cluster reclaims forwarded jobs. Started jobs will be checkpointed. Jobs will go back to pending state in the HPC cluster at top of the queue. The HPC cluster goes back to its initial status. It may restart checkpointed jobs whenever is the required resources are available in the cluster. Desktops clean up after jobs and quit from the desktop cluster. They are ready for desktop users

Stage 3) Next office time – Go back to stage 1

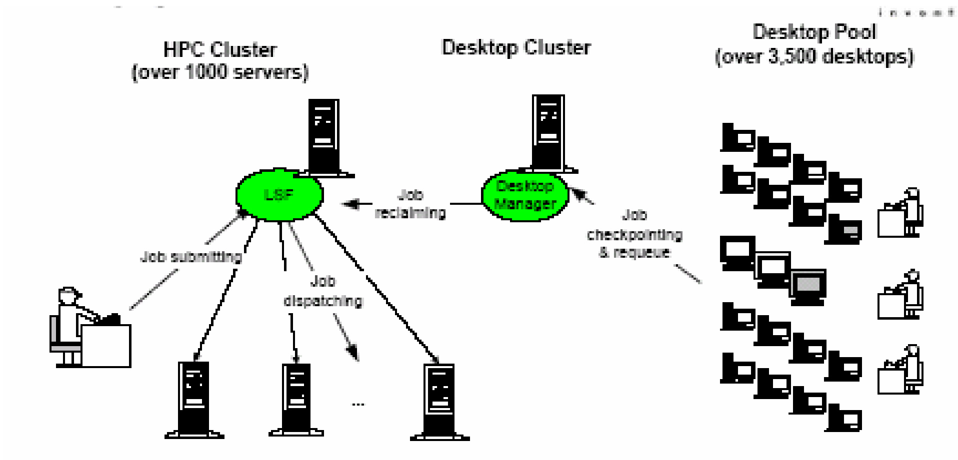


Figure 4 – System overview entering to next office time

.

Specifics on LS-DYNA jobs submission

The following diagram outlines the life cycle of a job on the UNIX desktop

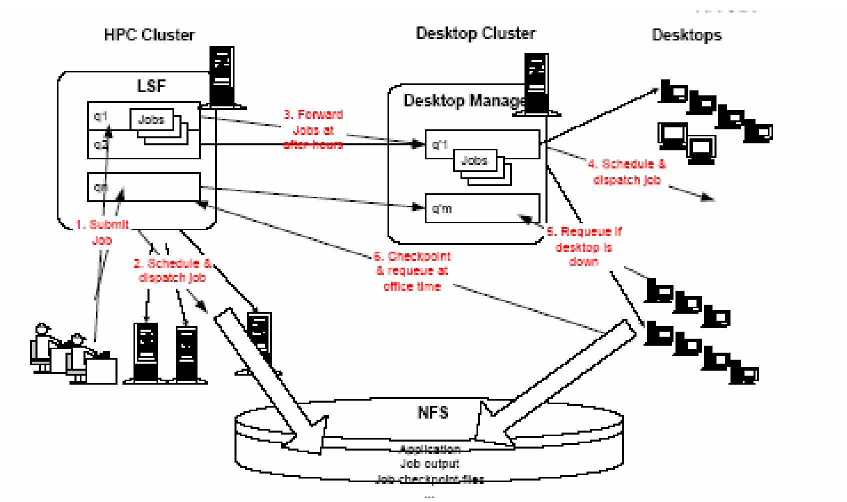


Figure 5 – Execution of jobs on desktops

LSF users submit jobs into HPC cluster. HPC cluster forwards some of the pending jobs to the desktop clusters during the desktop time window. Receiving queue(s) at desktop cluster(s) will queue jobs in the order of forwarding sequence. Desktop cluster(s) try to dispatch jobs following job execution requirements and scheduling policies defined at receiving queue(s). If one of the job execution hosts goes down, the job will be migrated. The job goes back to pending state. The scheduler will dispatch the job to other available hosts, and the job will start from the last checkpoint, or from the beginning if no checkpoint exists. The job outputs are saved on a shared file system that both HPC and desktop cluster can access, or transferred back through existing LSF file transfer solution on non-shared file system

.

If job(s) cannot finish during the time window, jobs will be checkpointed. The checkpoint data are saved in the shared file system. The jobs will be requeued to HPC cluster at the top of the queue. The desktops and the desktop cluster cleanup after any jobs that are forwarded.

Implementation Challenges

There were several challenges to implementing this solution at DaimlerChrysler, including:

- LS-DYNA checkpointing: while the use of a “d3kill” file is an acceptable method of checkpointing LS-DYNA jobs in most cases, we needed a much faster way to checkpoint jobs. For this reason, we used LS-DYNA “running restart” files. This required DaimlerChrysler personnel to make significant changes to the way they submit jobs.

- Running LS-DYNA in a heterogeneous environment: in the past, LS-DYNA has been shown to calculate different results on different computer architectures. Since the desktop grid is likely to be different from the hosts in the HPC environment, it was a pre-requisite for this project that LS-DYNA be configured to yield identical results on all machine types. DaimlerChrysler and LSTC invested significant resources in satisfying this prerequisite.

- Network storage of checkpoint files: in order to be able to restart jobs on another set of hosts after being requeued, network storage of checkpoint files was required. To simplify the necessary job control scripts, a single NFS server was used for this purpose. However, because of the volume of traffic (potentially several hundred hosts writing several GB of data at the same time), there were problems in configuring this server to perform reliably.

- LSF scheduling issues: we make use of an LSF “pre_exec” script to set up a host for running a job, and a “post_exec” script to clean up the host after the job completes. By the time the post_exec is run, LSF considers the job to be finished, and therefore the hosts are available for running another job. Therefore, it is possible for a pre_exec to begin on a host before the post_exec from a previous job has finished. We needed to put significant error checking into the scripts to handle this situation.

Conclusions

The LSF Desktop solution has been shown to be a viable solution for creating a “grid” of engineering workstations, for running LS-DYNA jobs. This solution has been used successfully in a production environment for close to a year. DaimlerChrysler is able to run several 12-way parallel LS-DYNA jobs each night, using “scavenged” desktop cycles. Catia users have not noticed any performance degradation on their desktops, and LS-DYNA users have seen significant improvement in their job turnaround time.