Weiterentwicklung der Linux Cluster Umgebung bei EDAG

Authors:
J. U. S. Hanlon,
EDAG Engineering & Design AG

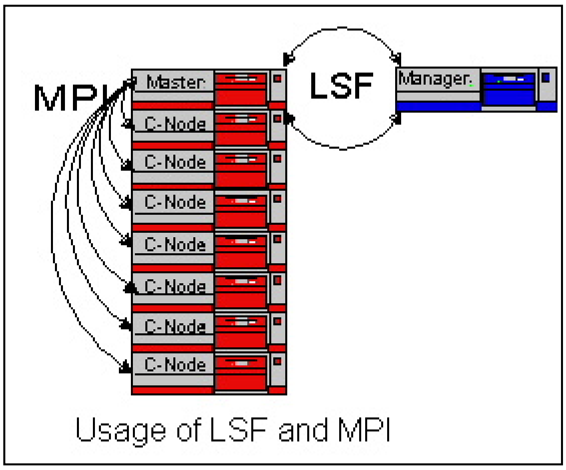
B. Platz,
clucon cluster concepts

Intro-

The Engineering & Design A.G. (EDAG) Linux Cluster System was first started in Jan 2002. It consisted of 16 CPU's running on the industry standard K7 Tyan hardware. The calculation turn-around times caused by the new Linux machines improved job turn around times so dramatically that the Linux cluster approach was seen as the future of High Performance Computing (HPC) for EDAG's CAE hardware culture. It was just a matter of 12 months before EDAG had another 13 clusters each consisting of 16 CPU's. These systems were located in groups at separate locations near the FEM engineering teams. The original management of these systems was a rudimentary scripting effort which through the LSF (queuing software) placed jobs on the clusters. The hardware management was facilitated through open source software. The road map of Cluster centralization was a pioneer effort which sought to bring multiple Linux cluster installations into one entity and govern it with one Job Management system, and to be nationally and internationally accessible for job submission and accounting. This was by no means a trivial effort which could be completed through over the counter software, but was a project that required the usage of self written ideas and concepts which were brought to bear on the maturing cluster environment.

Job Management History-

The original clusters were managed through basic scripts. The LAM and the LSF environments were started through the script which was local on each cluster block. (See Diagram 1)



Lam/MPI/LSF Flow
Diagram 1

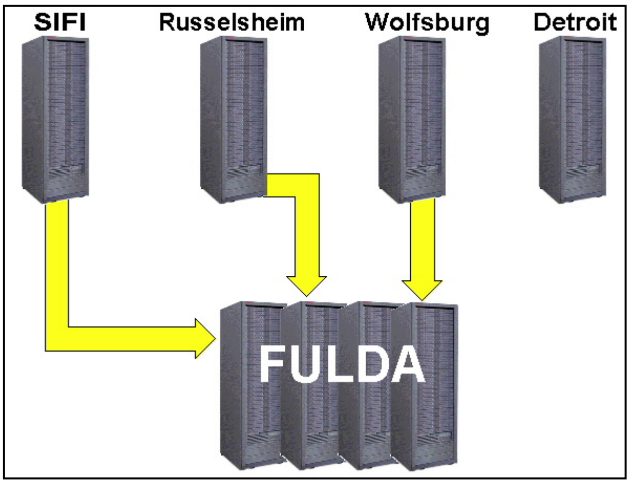
This created the problem that each block was an individual LSF domain causing a blindness across the clusters. There was no complete overview as to which cluster would be available next for a job. The users had to login into each block and inspect the queue to see if the machine would be available soon. This caused a valuable waste of time and resources which was slowed the job turn around. Another problem was the separation of assets and loss

of a global overview for a proper utilization of resources. The job load projected in the original ROI for the hardware was based on a consistent equal utilization of all cluster blocks. The separation of the systems through different locations, caused some cluster installations to be overloaded with long queuing times when others were used less often. This inter-cluster blindness and the decentralization caused loss of utilization were key factors that caused EDAG to pursue a centralization of clustering assets and a web-based job submission solution.

Centralization. –

Many factors lead to the final conclusion that the cluster environment could not be properly maintained with a decentralized deployment concept. One of the major factors is that a Linux cluster generates an enormous amount of heat. The average 32 bit, 16 CPU “block” produces about 1.2 kilowatts of heat. This amount, times the cluster count, is more heat than most small scale servers rooms can handle. This factor is very important to the successful utilization of cluster technology. The separate locations of the EDAG cluster were often marginally capable of handling the Linux clusters heat producing ability. Although the failure rate was low the heat problem was one of the deciding factors in the decision to centralize the EDAG cluster environment. Even though the ability to start jobs across decentralized locations was made possible while we were still using a script based job input systems, the non-graphic “line Command” method was difficult and required specialized training. This slowed the cluster’s integration and gave it a distasteful appeal to the engineers that used them. The need for a more user friendly job input solution was a mandatory product for better cluster acceptance.

During the centralization planning phase, it was decided that the web-based Job management environment should be first installed in Fulda and the other clusters would be migrated into the job management system. The job management software was successfully installed and immediately started to alleviate the inter-cluster blindness and queuing system problematic. During the next few months the remainder of the clusters were removed from their individual locations and integrated into the existing web based central cluster system. (See Diagram 2)



Centralization
Diagram 2

The system concept however was at this time not yet complete. The centralized systems consisted of two individual groups of cluster blocks, one had six blocks, the other had seven blocks. The reasoning for this was that the central manager for the Fulda system was still a single server with only RAID systems for system failure security. It was determined that the risk of all 13 blocks under one single server was too high a risk factor in view of a possible total failure. Each of the cluster groups were individuals and had no influence over the other, but there was a certain failure security if one of the mangers was to go offline. The next step in consolidation was to provide and singular redundant hardware system that would provide a stable and reliable platform for the Job Management software. The dual-server platform was a Fujitsu-Siemens RX300 2HE server using a Fujitsu-Siemens S30 Primergy Raid Hard-Disk Array as a multi-access raid system. The software used was Redhat 2.1 Advanced Server with the kernel update 2.4.9-e.35smp. This Operating System/Hardware solution laid the foundation for the combining of the remaining two cluster groups. In December 2003 the final installation was completed and EDAG had one graphic based Job Management environment for all of its Linux cluster computing capacity.

Necessity for a Web Based Environment.

In short the reasons were clear;

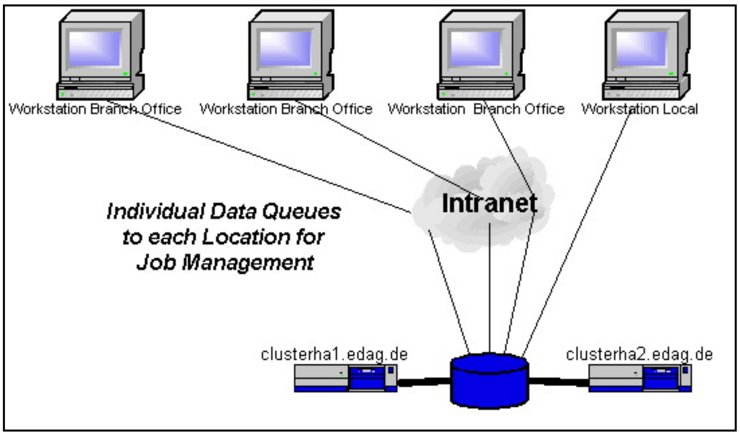
1) individual job submissions required multiple start parameters that needed to be rewritten continually while using Line-Command scripts,
2) inter-branch access, using a common interface that was capable of returning jobs to there original place of submission,
3) data sizes had dramatically increased
4) file transfer management and file transfer queuing,

5) the use of multiple solvers through one Portal,

all of these factors and more were reasons to implement a web-based “Portal” access that provided an FEM/CAE crash simulation job management environment.

The cornerstone of the web-based concept was to “Support the User”. The sometimes complicated process of starting a job was a persistent problem that required specialized training of engineers which carried training and “work-in” costs. The need to simplify the job input process and the work flow was the main force that drove the web based job management project. Job input was a matter of logging into the system, every system individually, and then executing long line commands with the needed switches and parameters. This was then controlled and checked by using other software commands through LSF to view job progression and status. Upon job completion the job would have to be re-transferred back to the submitting client, sometimes per Unix copy commands. The portal would alleviate these procedures providing a smooth browser based solution.

Another problem was the ever increasing data load on the intra-net data infrastructure. Even though the inter branch office network lines were considered originally sufficient, the ever continuing growth in job size, combined with the ever growing size of other commercial traffic over the EDAG network infrastructure, was creating bottlenecks in the file transfer portion of the job process. It was clear that the potential to “over-use” the band width was going to be a critical deficit in a centralized proposal. The condensing of data and the usage of varied file queuing strategies and protocols through the portal system was going to be a key factor in the centralization projects success. The problem before was that all data was using the same queue regardless of branch office. The solution was to clarify queues per location and then find the best transfer strategy to get them back to where they came from. This allowed files in Fulda to utilize the Fulda queue, while the files from Sindelfingen had their own queue which took understandably much longer to deliver a file. (See Diagram 3)



File Queuing

Diagram 3

Certain protocols had been given certain priorities over the network system and each queue was assigned the proper protocol to allow for a faster and more efficient delivery tailored to the location. The final step in file transfer improvements was the implementation of job reduction techniques and compression to reduce the size of the file that needed to be transferred. This alone was responsible for a great speed increase in file transfer. Since the file queuing project has been completed the band-width here at EDAG has since been increased and the engineers enjoy a fast and reliable system that is difficult to tell whether the user is local in Fulda or submitting from a different location, the input template is the same, and the download times are very comparable.

Web Based Job Management

The process from script based to web “browser” based job management was long and tested the patience of not only the systems manager but also the user. The software being far from perfect in the beginning was molded and improved upon over the running year. The basis was clear, browser usage for Job input. The browser was not important. The web portal was JAVA based so if the JAVA plug-ins were active the browser could access the portal servers and pull the job management software onto the client. This was a tremendous step forward towards central server-client management. Updates and upgrades to the browser software were now centrally accessed and implemented globally from one location.

As the software matured it was implemented on a parallel installation that could be used for production. The main input method of scripting was still in motion up until the web-based software was published and set as the only input method in Sept 2003. The web-based job management still held “line command” capabilities for those who preferred this method. At the time of the final implementation approximately 95% of the users had already transitioned onto the web browser software.

The multi-faceted abilities of LS-Dyna software and the hundreds of additional tools that have been scripted for it over the years make it one of the most powerful tools for Solver solutions on the market today. These abilities however make it very difficult to package into a Graphic User Interface (GUI). This task was at first too expansive to achieve in one step. The step by step methodical integration of “needed” extensions began with the first software version. To this date we have not only the standard switches in the software but also high end tools such as “Infmak” and a complete integration of accounting into the active SAP systems. The Portal software has opened the door for lower training times and cost, more direct accounting methods that allow for real time ROI accounting and project cost calculation on demand. The ability to utilize the software to pick and chose data to download has added the ability to allow jobs only to run only a portion of the job before data can be retrieved and processed further. Other functions of the portal is the ability to actively view all other jobs in the cluster system that are covered in the users Unix rights. This enables teams to work together on job processing with out having to share local client resources. The Team environment that is built by the portal enables projects to be shared across a companies national and international branch office structure. Through secure tunnel protocols the web-portal runs at the highest level of Intranet security, securing project integrity and assuring data accuracy. The need to assure the service of data integrity is our first priority.

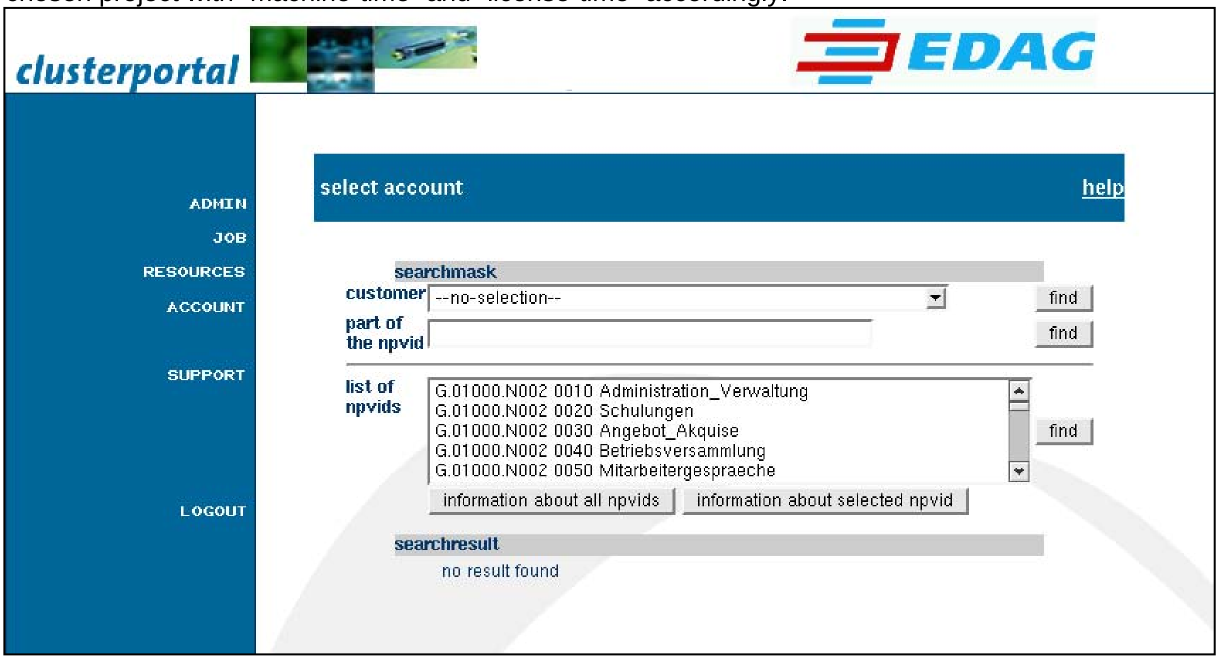
Web Based Portal Utilities

Administrative

The ability for users to be edited in the portal through an authorization structure separate from the companies central authorization is important. This internal authorization server enables users to be allowed certain “rights” with-in the Portal structure. The user must not however be re-entered into this internal system, all users who are contained in the companies central authorization pool are updated into allowable user pool in the portal every 24 hours through “cron” jobs. The user would then through the Portals Administrative GUI be granted access in the Portal with one of three different users levels:

- normal user level
- mini-admin level (authorized to kill and promote jobs other than their own)
- admin level (authorized all /root powers)

This provides freedom from the authorization central system, but still maintains a thread to it. If the user is not in the central system then they will not be able to be entered into the portal. This is an important factor when it comes to accounting. The accounting portion of the
portal is directly connected to the SAP system in EDAG. The user ID triggers the accounting process. The user in the job submission field (See diagram 4, screen shot accounting) then chooses the correct account to which they will charge the job. This is a direct link to the SAP which then charges the chosen project with “machine-time” and “license-time” accordingly.

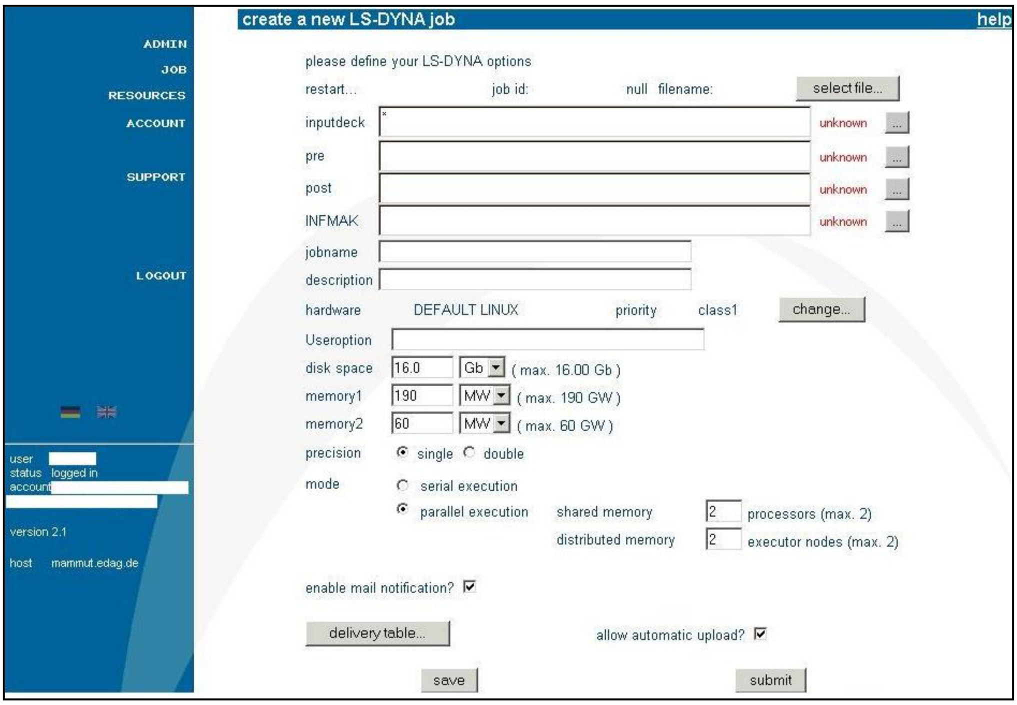


Screen Shot Accounting

Diagram 4

Job Input Interface-

The job input interface provides the user the ability to enter job start commands utilizing all of the switches that he would usually have available in the line command method. Using the interface however provides an overview of the complete job input process on one screen. The available settings, switches, file finding, memory allocations, and pre and post routines are all available centrally. The great advantage of this is the user can then save these settings on a job and use them as a template to run other jobs under the same conditions. (see diagram 5) This ability to easily enter jobs saves training time and offers the environment over the corporate network to all engineering depts.



Job Input Interface

Diagram 5

Possibilities

Although the administration and installation-concepts of the cluster are faster and easier than before, there is still some room for improvement:

1. replacing the fixed 16-CPUs-block structure by installing an alternavtive job management system like PBS on all cluster nodes
2. reducing administrative work by migrating to disk-less or almost disk-less compute nodes
3. better support of the engineers during the simulation process

Resolving the 16 CPUs blocks

At the moment all LS-DYNA jobs are using 16 CPUs. Because each cluster block contains 16 CPUs and only the first node, the head node, has LSF installed, other job running configurations with more or less CPUs are not possible. This is ok for the current demands, but in the future larger models, or new applications, need a new and more flexible concept with the possibility of requesting a user-defined amount of CPUs. So it is necessary to install the job management system on all cluster nodes. Installing LSF on over 220 CPUs equals high license costs, also the large amount of features provided by LSF are not needed at EDAG, it is logical to migrate to a different job management system like PBS or GridEngine.
From the users viewpoint the migration is transparent because by using the portal they don’t interact with the job management system directly. This is a great advantage of the portal, that contains interfaces to all important job management systems, while the access for the users is always the same.

Diskless Nodes

At the moment installing a new full running cluster node takes about 20-30 minutes. In addition, changes (e.g. new software packages) have to be rolled out to all cluster nodes individually, also the hard disks on each node provides a certain risk of failure. With the availability of the two HA-nodes clusterha1 and clusterha2 and the multi-access SCSI-RAID system, the image of the cluster nodes can be centralized and installed on this system. Then each cluster can boot from this image by using DHCP and tftp, therefore a full installation on the nodes’s hard disk is not needed. Nevertheless for I/O-intensive applications like MSC.Nastran a simple scratch-disk should be installed on each node.

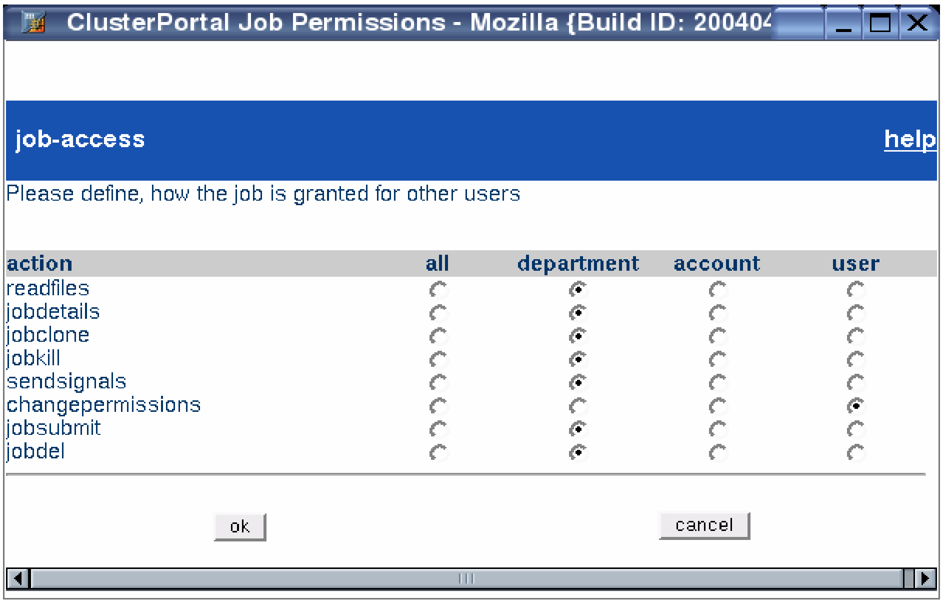
Improving the CAE-Workflow for the Engineer

The submission of jobs with the portal is very smooth. The automatic data transfer to and from the compute nodes back to the workstation reduces the engineer’s work dramatically. Nevertheless the work with the cluster and the portal has shown, that some processes could be bettered or are not yet addressed. The three most important things are:

- better support of project collaboration within the portal
- controlling or accessing intermit job results during a job-run
- information about the estimated walltime of the jobs for better scheduling and better efficiency of the cluster

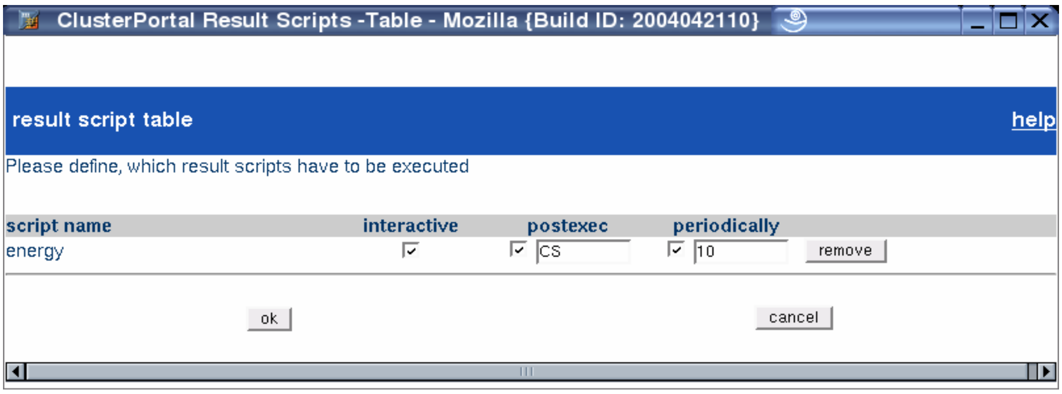
Better Support of Project Collaboration within the Portal-

In to order to allow the best colloboration within a project it might be important to allow all team members portions of control over the jobs and the job results. Therefore it is important that the user can allow control over his or her jobs. This provides explicit access rights, controlled by the user and job owner. These rights can be controlled centrally through templates, and then individually enlarged or reduced as the users sees fit. The authorization is almost the same as in UNIX. You can see a GUI mask in the picture below. It shows the definition of access-rules of a job.

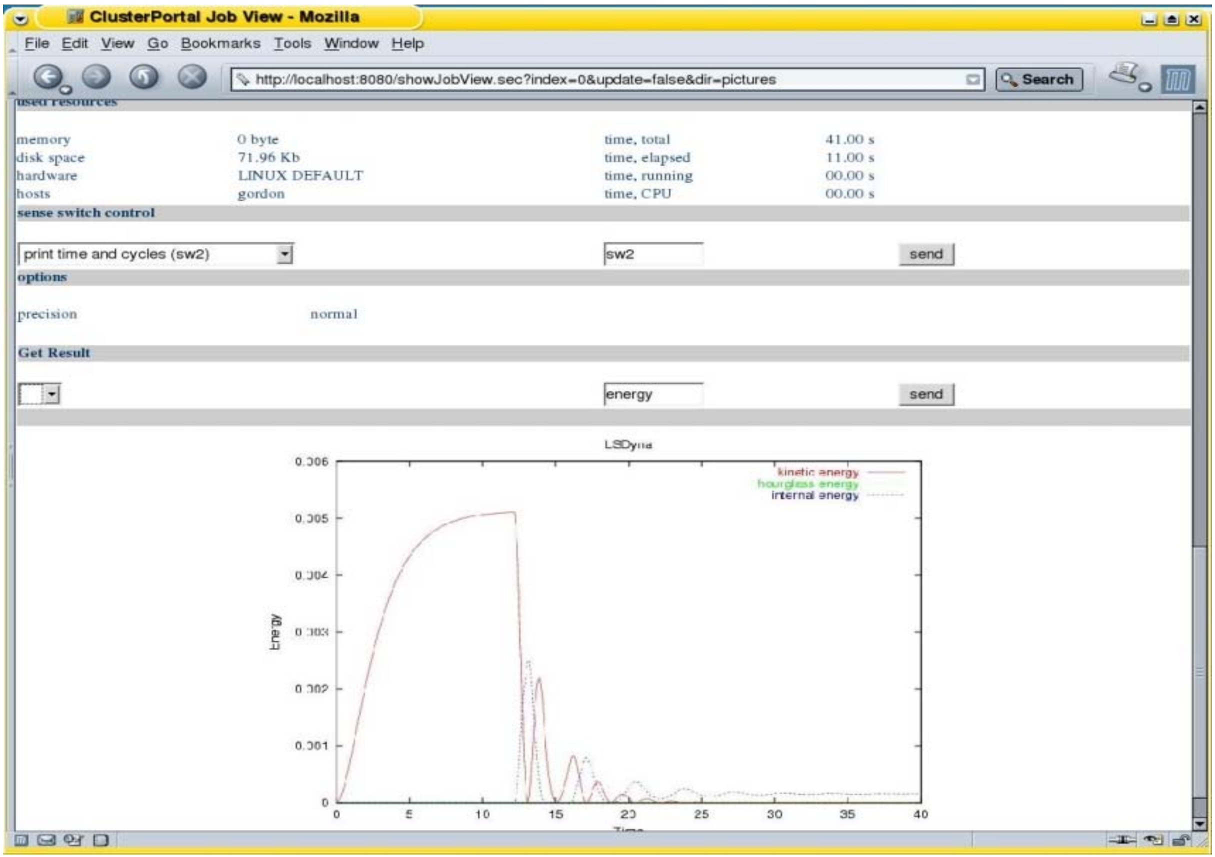


Controlling or Accessing Intermediate Job Results during a Job-Run -

The portal provides full access to all the outputfiles created by the application during the run. If the engineer wants to evaluate intermediate data to decide whether the job is ok and will end with correct result. He can download any needed files from the portal over to his workstation, and start his post-processing tools. Usually the user needs only simple information from the job, (for example the characterisitc of the kinetic or hourglass energy) so it would be useful if he could define arbitrary scripts, which would be triggered during the job run. The following example shows a sub-panel of a prototyp of the portal, where the user can define scripts, that helps him to improve his CAE-workflow:

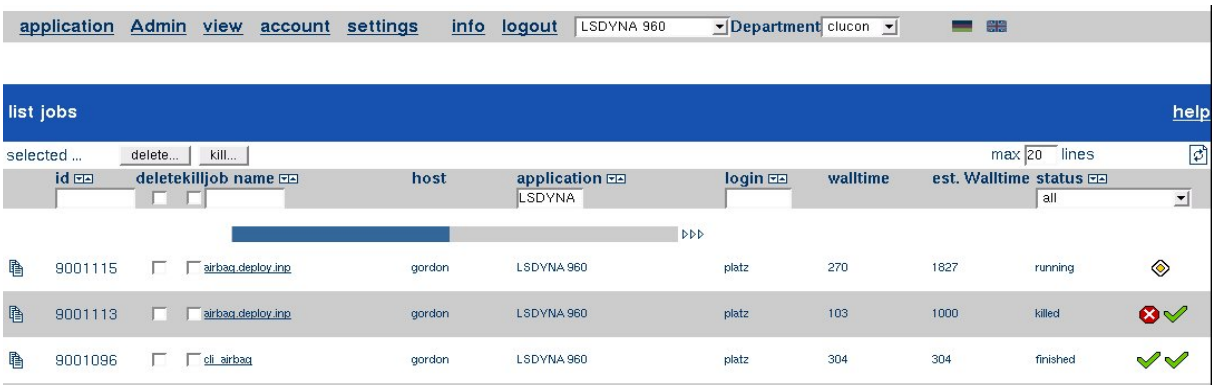


In this example the user has defined a script called “energy”, that will trigger during the run (interactive is marked) by clicking on a special button in the jobpanel during the run. In addition the same script is called peridically every 10 seconds while the job is running. The script is stopped on the compute node after LS-DYNA is finished (postexec = CS). So the user can control the energy only during the job-run:



Information about the Estimated Walltime of the Jobs for Better Scheduling and Better Efficiency of the Cluster

In order to help the engineer bring in his project on time, it would be useful if the engineer could get information about the estimated walltime of his jobs so he can estimate when his jobs (or the jobs of a other employee) will be finished. By looking at the calculated cycles in the glstat-file of DYNA he can receive this information, but this method is very uncomfortable and fault-prone. An automated calculated walltime estimate of all jobs containing this information, the job scheduler could then determine the estimated start-time of new jobs. This leads to a better efficiency of the whole cluster.



This example shows that the job 9001115 has a estimated walltime of 1827 seconds, his current walltime is 270 seconds.

Conclusion-

The setting up and running of a Linux cluster for HPC calculation environments is not anything new anymore. Even small companies have entered into the 21st Century and are running small 16 CPU or less systems for multiple HPC requirements. The future of HPC lies not in the magic of the mpp process, but in the management of the users, jobs and applications. As hardware speeds increase and latency times are filed down, the last frontier that's needs to be defined and standardized is the user, job and application management of these massive systems. We here at EDAG have been active and experimental in achieving a professional management system that fits our needs for now and the years to come.