J90 SuperCluster

Bruno Loepfe, Computing Centre, ETH, Zurich Switzerland

ABSTRACT: *ETH Zurich operates a cluster of four J90. In a co-operation agreement between CRI and ETH the necessary concepts and software will be evaluated and implemented to provide a Single-System-View of the cluster to the users.*

The co-operation CRI - ETH

History

In early 1994 the Computing Centre of ETH Zurich considered replacing its Y-MP/464D. At that time CRI proposed the configuration shown in figure 1. It is a cluster of four J90, connected to a HIPPI-switch, to which a network-disk is also attached. The proposal looked quite promising: it would raise our peak-performance from 1.3 GFlops to 8 GFlops. It would also have solved our memory-problem by replacing the old 64 MWords with 1.75 GWords.

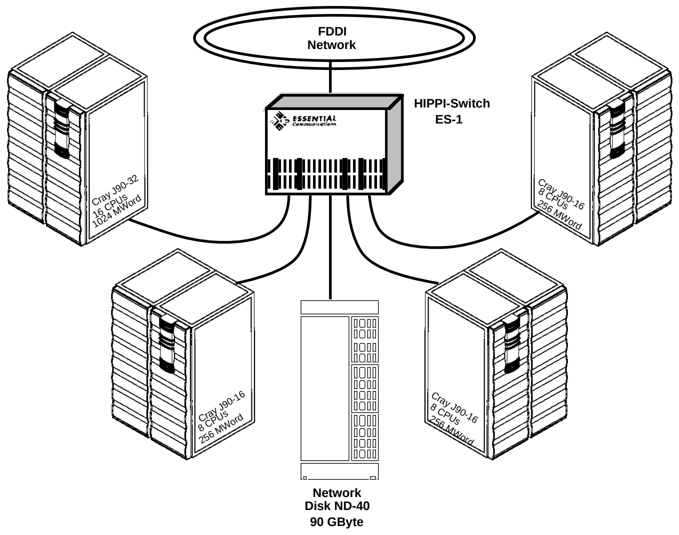
FDDI Network
HIPPI-Switch ES-1
ESSENTIAL Communications
Cray J90-32 16 CPUs 1024 MWord
Cray J90-16 8 CPUs 256 MWord
Cray J90-16 8 CPUs 256 MWord
Cray J90-16 8 CPUs 256 MWord
Network Disk ND-40 90 GByte

Figure 1: Cluster with 4 J90, HIPPI-switch and network-disk

Problems

The proposal had a few drawbacks though. First the peak performance of the single processor of a J90 is 200 MFlops only, as opposed to 333 MFlops for the processor of a Y-MP. In addition, the processor of the J90 has one memory-port less than that of the Y-MP. We found that this reduces the speed of SAXPY-based codes by a factor of 2, instead of the theoretical 1.67. But we expected that the larger memory would provide more opportunities for multitasking and thus compensate for the slower CPU.

Another drawback was the HIPPI-disk, or rather, the shared file system (SFS), the proposed way to access this disk. At first sight SFS offered impressive features. The file system would have been very resilient: any of the machines could crash, but the files would still be available for all the others - accessible from any node in the cluster at a transfer rate of 50 - 60 MByte/s. This seemed ideal for holding the file systems containing user home-directories.

It turned out, however, that the SFS access latency is roughly 100 ms. Thus, a 4 KByte transfer has an effective rate of about 40 KByte/s, a 64 KByte transfer about 600 KByte/s. A survey of our Y-MP files showed 88% with 64 KByte or less. SFS delivers excellent performance for sequential I/O-operations with large blocks (>= 2 MByte) on large files, whereas we have short files, or random access, with small blocks. At this time we decided, that the biggest J90, equipped with as much memory as possible, would better suit our needs.

The co-operation

Next CRI offered us a co-operation, which among other benefits included a joint funding of up to 6 projects. The following 6 projects have been selected:

1. SuperCluster Development and Operations
2. CLUMET: SuperCluster for Meteorology
3. Computational methods for the study of protein structures and intermolecular interactions
4. PVM based SCIDDLE
5. A sparse direct solver for large multigrid systems; Parallel multigrid methods for the continuity equations in semiconductor device simulations
6. Direct Numerical Simulation of Turbulent Flow

After negotiating some details, we accepted. This paper deals with project 1. Except for project 4, the others are self-explanatory.

SCIDDLE is a remote procedure call (RPC) package, which allows easy programming of distributed applications on a network of loosely coupled processing nodes. It generates the code for the communication automatically, and splits up the user-program into a client and a server. The authors tested it first in 1991 with a chemistry package named DISCO on Crays in Zurich, Lausanne and Minnesota and reached 1.2 GFlops. In 1993 they tested it with a C90 in Pittsburgh and another one on the Cray Corporate Network, and reached 16 GFlops.

What is a SuperCluster ?

Possible answers

Many people are already familiar with some kind of clusters. Workstations are usually installed in clusters. Personal computers can be networked. So what is the difference between a cluster and a SuperCluster ? An argument might be that in a SuperCluster, the participating machines are super computers. Another one might be the presence of the HIPPI-switch, which provides a fast connection for the participating machines.

Single-System-View, Single-System-Image

In our opinion, the distinction between SuperCluster and cluster lies principally in the software layers which provide a Single-System-View (SSV), rather than in hardware characteristics.

In its invitation to the *Cluster Forum* in October 1995, CRI gives the following definition of SSV: "*Single-System-View involves providing a coherent view of the cluster only for those functions, where it most matters*". SSV is a relaxation of the Single-System-Image, where a coherent view of the cluster for **all** functions is required. CRI and ETH agree that the phrase "*where it most matters*" should mean that a coherent view is provided for jobs/sessions only, as opposed to SSI, where coherence is provided for every single process. So SSI is the logical extension of SSV, a long-term vision.

The semantics of SSV (and even more so of SSI) is, that a cluster presents itself to the user as a single system, in fact pretends to be a single system. This means that the user does not have to know how many machines are participating in the cluster. He does not have to know their host names nor their IP-addresses. In fact, he doesn't even have to care on which machine his job/session really resides. When working on a cluster with SSV, one should have the look and feel of working on a single system. The fact that, in reality, one is working on a cluster of possibly different machines should be transparent to users.

Consequences

The implementation of SSV faces a couple of real challenges:

- transparent file access,
- load balancing,
- clusterwide authentication,
- clusterwide accounting/administration/operating.

When working under UNIX or any of its derivates, the most basic task is accessing files. SSV requires that regardless of where a job/session is actually executed, it always sees the same files or file system layout. In addition, a user can always use the same set of commands to access files within a cluster. There is no need for *rcp* or *ftp* in a cluster, *cp* will always work.

Load balancing is not really mandatory in a cluster, but it is ranked extremely high in the list of wishes. In order to obtain maximum throughput from a cluster, one would like to distribute incoming jobs/sessions to different machines. It is, however, not always possible to distribute the load evenly. A particular node in a cluster may provide a unique resource, so that all jobs requiring this resource are forced to run on this node.

SSV also requires that authentication on a particular host for a particular services implies authentication on the whole cluster for this service. For example: if you are logged in to the cluster, your session will be assigned to one of the machines in the cluster. From there, submitting a job to NQS/NQE shouldn't require a password because the load balancer might assign this job to another host. In the same vein, a password change on one host should propagate to all others in the cluster. Otherwise these changes would have to be done manually.

Clusterwide accounting/administration/operating is another important point. For users, a clusterwide accounting is mandatory. Users don't want their statistics (bills) for the usage of resources separately. A clusterwide view of the resources has to be provided, and based on local accounting data collected on each machine.

The system-administrator needs tools to manage the cluster as a whole. It would be a major pain if the creation/deletion/maintenance of accounts would have to be done on each machine separately.

An operator needs global information about the state of the cluster. The more machines participate in a cluster, the harder it is for an operator to identify a particular system's console, for example one that just started to emit alerting beeps.

Certainly, these four points do not cover SSV entirely. But we think, that they are the most important ones. These are the areas, where we concentrate our efforts.

Problems/solutions

Transparent file access

As already mentioned: UNIX heavily relies on files. So accessing files in a cluster appears to be the crucial problem. In our opinion, three aspects are important:

- transparency,
- resiliency, and
- performance.

The low ranking of performance is indeed intentional. In a cluster, you need transparency and resiliency first before providing performance.

The possibilities for accessing files on Cray machines are NC1, NFS, DFS and SFS. If one needs transparency, NC1 is not suitable. If one needs resiliency, SFS is the choice. But as explained earlier, for our file sizes and I/O-operations, the performance of SFS is not sufficient. What we really need is the performance of NC1 combined with the resiliency and sharing capabilities of SFS.

Fortunately, we found a solution: SFS exported via DFS. This way, you use the DFS-cache to hide the high latency of SFS. DFS, however, requires a DCE-cell. With its complexity, DCE needs considerable efforts in building up know-how, in administration and in maintenance. Like NFS, DFS exports file systems. But when an exporting machine crashes, SFS is still available for other machines in the SuperCluster, so a fall-back server for DFS on another machine can take over. This way one can get almost the same resiliency as for SFS. In addition, we can obtain 75% or more of the performance of NC1. This has been measured on a dedicated system; it is not clear yet how the combination SFS/DFS will perform under heavy load. We hope that we will need some fine-tuning only for DFS-parameters like the size of the local cache.

An open problem is the co-operation between the distributed Data Migration Facility (DMF) and the available file system-types and combinations. Distributed DMF is currently not able to handle all combinations. The problem is that DMF performs I/O-operations on the file systems directly and therefore requires the "real" path names, rather than path names which a user sees in file systems exported with DFS. Very likely, this problem will be solved by replacing the commands *dmget* and *dmput* with local versions, which will transform a path seen by the user into a path needed by DMF. Then they delegate the get- or put-operation to the server, which actually exports the file system.

Load balancing

The problem of load balancing is solved. The current NQE 2.0 does load balancing of batch jobs. NQE 3.0 and later NQE 4.0 will be able to distribute interactive sessions as well.

Clusterwide authentication

This is an open problem. When logging into the cluster, one gets an authentication for the local machine and the DCE-cell. But authentication is for the local machine only, without authentication for the other machines in the cluster. OSF DCE 1.1, expected around the end of March 1996, will deliver this functionality.

In this category, we also have the problem of forwarding DCE-tickets. It is actually not possible to forward the ticket of an interactive session to a job submitted to NQE from this session. One has to type the password again, when submitting a job, so that it can get its own ticket. As an intermediate solution we decided that a user will have two choices: either the job runs on the local host only, (then no password is needed), or the user tries load balancing, (where a password is needed to get a DCE-ticket for the job). Towards the end of 1996 NQE 4.0 should be available, which should cure this problem.

Both problems are not critical. But, they will complicate the life of users, thus reducing the acceptance of the SuperCluster. It is perhaps rather a psychological than a technical problem, but it definitely should not be underestimated.

Clusterwide accounting/administration/operating

At the moment, the clusterwide accounting is in its conceptual phase. There are some ideas how it should work, and how it could be implemented.

Clusterwide administration does not yet exist. At the moment, we "declared", that the main production machine in the cluster holds the "reference-UDB". All changes to the UDB are done on the reference-UDB. Subsequently, the reference-UDB is copied to all other machines manually. CRI has announced Central Administration CA 1.0 for mid 96. The DCE-registry will not be included in version 1.0.

CA 1.0 will contain features for operators as well. They will receive an activity display, alert information, tape information, disk information and network disk status information.

Actual state of the project

The four machines are logically divided into two parts: the main production machine (a J932, with 16 CPUs and 8 GByte memory), and three co-operation machines (J916, with 8 CPUs and 2 GByte memory each). The main production machine is open for all users, and replaces the old Y-MP. The co-operation machines are only open for users of one of the six co-operation projects. Their benefit is the funding, faster turn-around-time for jobs, and more available disk-space. Their drawback is that they are the "field-testers" for the new parts of SSV as soon as they arrive.

Due to the different arrival times of the necessary software-parts, we expect to have a restricted SuperCluster operational around mid 1996. This restricted SuperCluster will involve all four machines, so the logical division mentioned above can be removed. The full version of SSV is expected to be operational around 1Q97. The project ends at the end of 1997.

Conclusion

In the framework of a co-operation agreement between CRI and ETH we are implementing a Single-System-View for a cluster of processing nodes which should give users the look-and-feel of a working on a single computer system. The advantage for the user is, that he doesn't need to care which node in the cluster provides a particular resource. The cluster can even be expanded to provide new resources, without causing significant changes for users.