Improved Clusterperformance through Parastation Software

Dr. Jochen Krebs

Cluster Competence Center GmbH,

Munich, Germany

Abstract:

This paper gives a short overview on how Parastation Cluster Middleware can help to speed up LINUX Compute Clusters (LCCs) by using an optimized MPI Library. Benchmark examples are given for realistic DYNA testcases. Parastation provides a single MPI programming interface for various industry standard interconnects. The software is licensed on a subscription basis by the Cluster Competence Center GmbH, providing maintenance & consultancy services for LCC environments.

Keywords:

LINUX Computer Clusters, Performance Optimizations, Service & Support

1 Parastation History and the Cluster Competence Center

The Cluster Competence Center GmbH has historically evolved out of a research initiative in the early days of clustering at the University of Karlsruhe under the supervision of Prof. Walter Tichy. The outcome of this effort was a middleware layer for LINUX-based Compute Clusters named *Parastation*. Starting in 1999, a newly founded company - the ParTec AG – marketed this software package commercially. *Parastation* was built to provide a runtime environment for parallel jobs including an optimized MPI library to support a variety of industry-standard interconnects, such as Gigabit Ethernet and Infiniband. Over the following years, ParTec was able to win a remarkable number of reference customers including very big installations such as the 1000 processor ALICEnext cluster at the University of Wuppertal.

While ParTec's business model was built entirely on selling software licenses, it became clear in 2004, that the market demanded a radically new approach for at least two reasons:

- The cluster software available through open source license models had gained sufficient maturity and stability so that many of these packages provided a viable alternative to commercially available solutions. Clearly, a company-driven development effort could hardly keep pace with this evolution.
- With LINUX clustering becoming more and more mainstream, there is a growing demand not for specific "point solutions" but for a complete cluster software stack, addressing all aspects of job queuing and scheduling, parallel file I/O, software provisioning, software development, etc. A "one-fits-all" software solution would be hardly appropriate for the diversity of customer requirements.

Therefore, the Cluster Competence Center takes a much more service-oriented approach to the customer. It's main goal is to deliver integrated and complete software stacks for LINUX-based Compute Clusters by selecting state-of-the-art software components and driving software development efforts in areas where real added value can be provided. It helps it's customers to develop and implement their optimal cluster deployment strategies and optimize their cluster environment. In contrast to conventional system integrators, the CCC has considerable software development knowledge in-house that helps getting a deep understanding of all aspects of cluster operations and to deliver to the expectations of the end-user. While the open source model is supported wherever applicable, the CCC also works with commercial software vendors to integrate their solutions when appropriate. With this software philosophy applied, customers get the choice to pick the best components suited for their needs, without sacrificing interoperability between all the modules in the stack.

Naturally, the *Parastation* product in it's current version *Parastation 4* is also supported and further developed as an important part of the CCC portfolio..

The Cluster Competence Center GmbH is engaged in cooperative partnerships with well-known companies in the field of hard- and software, as well as with service providers in high performance cluster computing. Focus of these business relationships is the ability to deliver complete cluster solutions including maintenance and support.

In particular, Dynamore supports a *Parastation 4* version of LS-DYNA for Ethernet-based Linux clusters. The CCC also has a reseller agreement with Pathscale Inc. for Pathscale's highly optimizing EKO Compiler for Opteron processors and works closely with infiniband vendors and software houses delivering other cluster components such as parallel filesystems.

2 Basic Features of Parastation 4

The *Parastation 4* software consists of two main parts:

- A software environment for starting, managing and controlling parallel applications at runtime
- A single MPI library optimized for various cluster interconnects

2.1 Process Management and Administration

The *Parastation 4* process management is responsible for starting, monitoring and stopping of parallel processes and ensures optimal usage of compute cycles:

- When processes are started, the *Parastation 4* load balancing feature schedules parallel tasks according to node availability and their current workload
- Jobs can be started on any node
- *Parastation 4* uses an optimized, low-overhead starting mechanism to distribute the tasks
- Very large jobs can be started efficiently
- Non available node are detected and ignored
- There is a permanent control of running processes

In case of node failure, the job clean-up facility automatically stops all jobs and the parallel processes associated with that node.

Jobs can be queued on a first-come first-served basis and there is also a pre-emption mechanism that enables correct job suspension and –resume in favour of higher priority jobs.

.

The virtual node concept avoids static node lists and ignores non-available nodes transparently for the users. Nodes can be selected according to various criteria and exclusive rights can be given for specified nodes, processors or users.

For cluster communication, the optimal network is selected. Within a node, an optimized shared memory communication mode is used. If redundant networks are present in the cluster (such as a dedicated high-speed cluster interconnect and an Ethernet connection), *Parastation 4* automatically selects an alternative connection at application startup in case of failure of a network.

For elementary user requirements, it can also serve as a simple queuing system. If more sophisticated features are required, *Parastation 4* can also be integrated with batch systems such as PBSpro, LSF, Sun Grid Engine, etc.

2.2 The Communication Library

The MPI Library is fully compatible with the industry-standard MPICH 1.2.5 API. Since a single MPI library is used for all cluster interconnects, the specifics of the interconnect are hidden from the user as well as from the application software provider. This reduces the risk for deploying emerging technologies in a production environment such as Infiniband.

Parastation 4 enhances the performance of distributed applications through optimization of parallel communication. To achieve optimal results, a relinking of the application is necessary if static libraries are used. There is no recompiling of the application required.

The *Parastation 4* protocol was developed specifically for cluster communication to achieve optimal performance with minimal overhead. A safe delivery of data is ensured. Distributed applications benefit directly from the resulting lower latencies and higher throughput rates as shown in the examples below.

In those cases, where the application is available only in binary form (as may be the case for many ISV applications) the so-called TCP-bypass functionality of *Parastation 4* enables TCP-based communication to be redirected automatically to the optimized *Parastation 4* protocol. This mechanism works for all application and services which communicate over TCP and is only used for intra-cluster message exchange.

Besides Ethernet, *Parastation 4* also supports Infiniband as well as Myrinet as a cluster interconnect.

3 The P4 sock protocol

By far the most LINUX clusters communicate over a Gigabit Ethernet interconnect. In spite of clear advantages of GigE in terms of costs, vendor independence and robustness, there are well known drawbacks in terms of communication performance. This is both true for bandwidth- and for latency bound parallel applications. The grey bars in fig. 2 & 3 show peer-to-peer measurements of MPI latency and bidirectional bandwidth (Pallas MPI Benchmark PMB 2.2)for the well-known “channel-p4” interface; the setup in this case consists of two Dual XEON Systems at 2.6 GHz and 2 GB of memory, SuperMicro motherboards P4DPE-G2 (E7500 chipset) with Intel E1000 (82540 chipset) on board and Broadcom NetXtrem BCM5701.

The “best-case” latency in this test is 27 microsecs, the “best-case” bidirectional bandwidth is only 140 MB/sec, which is far below the theoretical value of 250 MB/sec. In more realistic configurations, when a switch is present, the hardware latency of the switch adds to the latency of the TCP/IP software stack.

Other high-speed interconnects, such as Infiniband, Myrinet or Elan from Quadrics, avoid the overhead of the TCP/IP software stack by bypassing the operating system kernel and exchanging data directly between the user address spaces. While this method has a clear advantage in speed, it is difficult to implement on top of GigE hardware and it also has certain drawbacks with respect to hardware independent implementations and potential security problems.

In order to enhance the communication performance of GigE in LCC environments by simultaneously preserving all the other benefits of this interconnect technology, *Parastation 4* uses a so-called “hybrid approach” for the cluster communication. This approach is shown schematically in fig. 1 below.

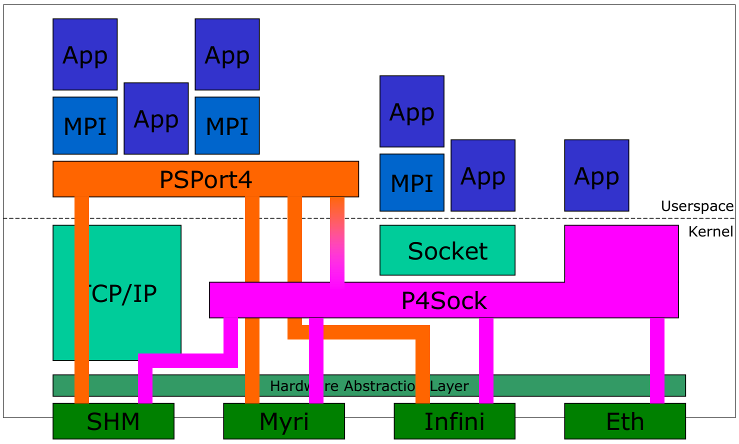


Fig. 1: The Parastation 4 communication architecture

Let’s first have a look on the upper left hand side of the graph. Parallel application can access the *Parastation 4* communication infrastructure through a standard MPICH API. All underlying specifics of the supported communication interconnects are hidden from the application programmer and cluster user through the so-called “PSport” library, that requires relinking (but not recompiling) of the application (an alternative way of accessing the communication layer that completely avoids application modification will be discussed below). The PSport library supports the Infiniband and Myrinet interconnects in the normal user-to-user mode, without any intervention of the operating system. It also provides a specially optimized module for the intranode communication within the cluster nodes. Measurements in the case of Myrinet show a performance advantage for shared memory communication over the standard GM-2 protocol of a factor of 15 (15 microsecs latency in the case of GM-2 vs. 1 microsec latency for *Parastation 4*).

In cluster environments where the MPI communication occurs over GigE (which is the situation we are focussing on in this paper), the “hybrid” path is taken (shown in orange and magenta in the graph) that branches to the “P4sock” layer in the operating system kernel. This layer marks an alternative path to the underlying GigE hardware, completely bypassing the TCP/IP stack. It should be noted, however, that the reliability and failure-tolerant features of standard TCP/IP features are fully preserved in the P4sock implementation.

The P4sock modification of the standard LINUX Kernel provides a lean and reliable method of internode communication over GigE networks. It is entirely transparent to the parallel applications and requires no modifications on source code level. In situations where multiple networks are present (like an additional system management network or additional high-speed interconnects like in the case of the ALICEnext cluster in Wuppertal), and when a network failure occurs, the application may use alternative communication paths (even over different interconnects). This can be triggered by simply restarting the job (higher layers of *Parastation 4* can do this automatically). No further modifications of the cluster runtime environment are necessary.

What happens in situations where a parallel application is only available as an executable (like for packages from Independent Software Vendors) and can not be re-linked with the PSport user library? This case is shown schematically on the right hand side of Fig. 1: *Parastation 4* implements a “TCP/IP bypass” mechanism in the LINUX operating system Kernel. What this means is that socket calls from user applications can be trapped within the LINUX kernel and are redirected to the P4sock layer, thus again avoiding the software overhead of the TCP/IP stack. An interesting implication of this architecture is that the actual communication path in this case does not necessarily have to go through the GigE interconnect, but could also take the alternative paths to shared memory, Infiniband or Myrinet.

4 Performance Results

It is very instructive to discuss the performance results achieved in the *Parastation 4* communication environment compared to standard TCP/IP measurements. In the following graphs, in each case the exactly same hardware is used; modifications are entirely in software.

Fig. 2 shows the latency results for various packet sizes as measured with the standard Pallas PBM benchmark. The best-case value comes down from 27 to 11 microsecs and the performance advantage stays remarkably constant over the measured interval. The same is true for the bandwidth measurements (Fig. 3). *Parastation 4* achieves values up to 214 MB/sec, close to the theoretical peak performance of GigE (bidirectional). The limitations of the TCP/IP stack for cluster communication become apparent.

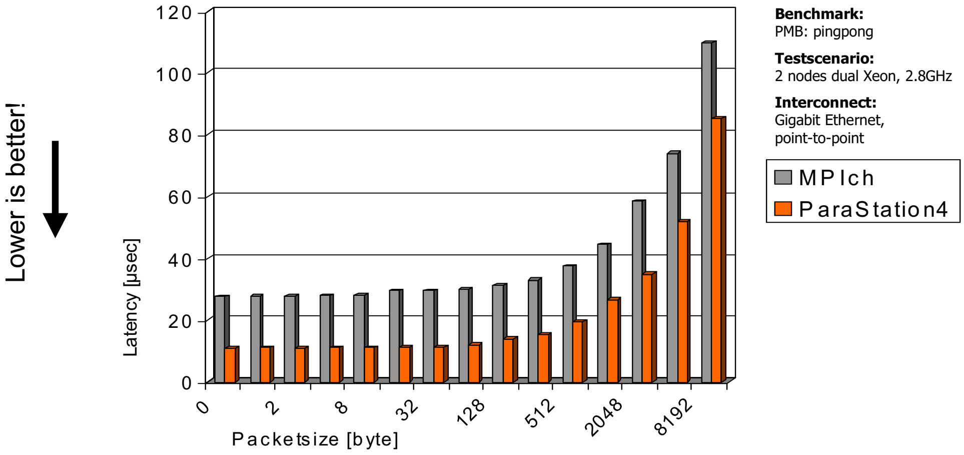


Fig. 2: Latency improvements through Parastation 4

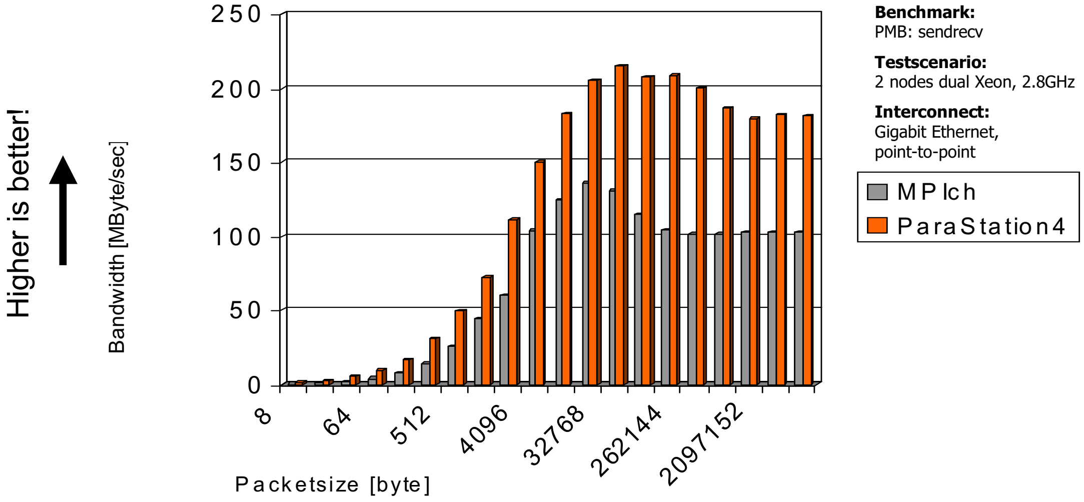


Fig. 3: Bandwidth improvements through Parastation 4

Fig. 4 shows results of a Linpack test on 12 CPUs (Dual Xeon) in a GigE cluster. There is a 30% performance improvement over the standard LAM implementations (A number of other tests has shown that there is only a marginal performance difference between standard MPICH and LAM implementations)

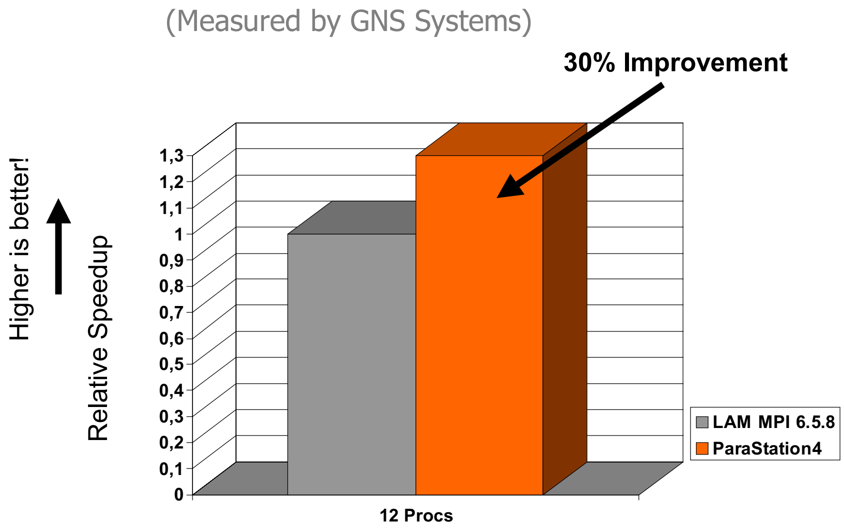
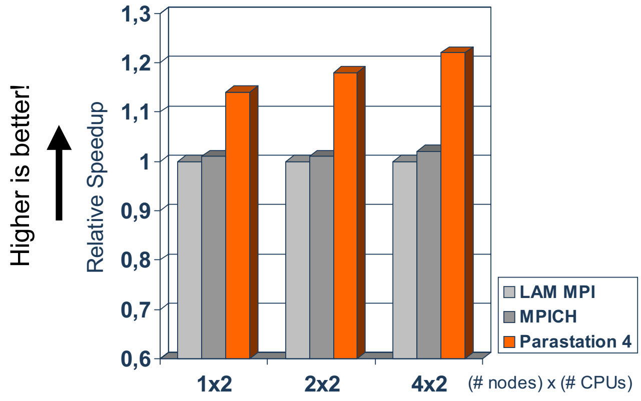


Fig. 4: Linpack Benchmark Improvements through Parastation 4

Fig. 5 & 6 represent a real-life benchmark test with LS-Dyna. In Fig. 5, the application was re-linked against the PSport library, Fig. 6 uses the TCP/IP bypass mechanism described above. In the case, the executable was not modified at all. The results show that there is a measurable performance benefit when the application is re-linked, but there is still a 10% improvement when the bypass is taken.

(Measured by FujitsuSiemens)



Benchmark:
mpp970 Chrysler Neon 30 msec

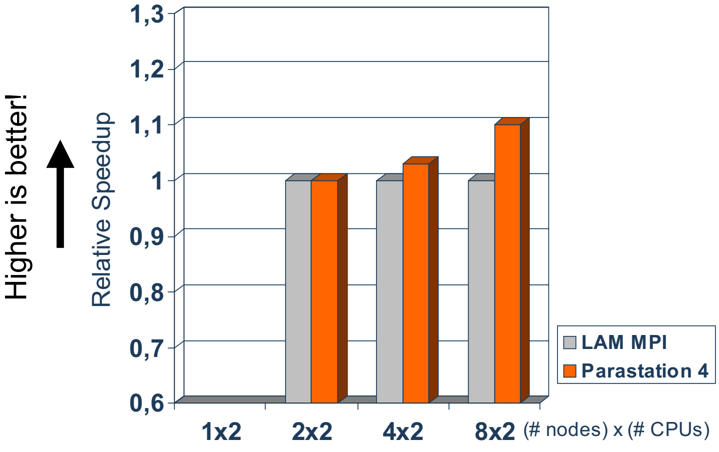
Testscenario:
8 nodes, hpcLine with Dual Xeon 2.8 GHz CPUs

Interconnect:
Gigabit Ethernet

Modell:
LS-DYNA mpp970 Chrysler Neon 30 msec.

Note: application linked with ParaStation MPI

Fig. 5: LS-DYNA Performance Improvements through Parastation 4



Benchmark:
mpp970 Chrysler Neon 30 msec

Testscenario:
8 nodes, hpcLine with Dual Xeon 3.2 (+3.06) GHz CPUs

Interconnect:
Gigabit Ethernet

Modell:
LS-DYNA mpp970 Chrysler Neon 30 msec.

Note: application unmodified! (LAM-MPI)

Fig. 6: -DYNA Performance Improvements through Parastation 4 (cont.)

Fig. 7 represents a realistic fluid dynamics test using the popular STAR-CD code. Speedup is up to 35% percent; since STAR-CD allows for dynamic linking, usage of PSport library is not a problem.

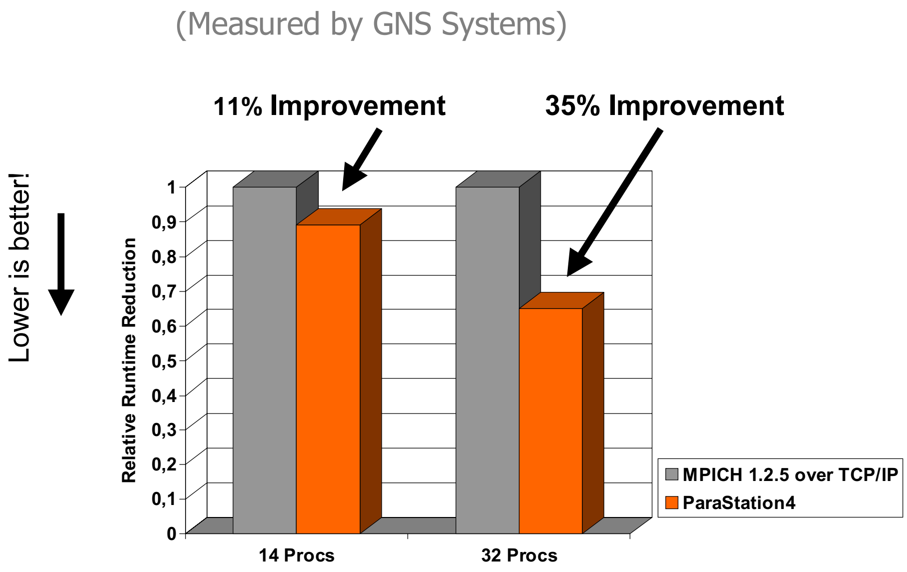


Fig. 7: STAR-CD Performance Improvements through Parastation 4

5 Summary

Using *Parastation 4* as a cluster middleware layer in a Linux compute Cluster Environment has a number of significant advantages for users:

- Better performance of parallel applications like LS-DYNA through optimized communication in the cluster
- Transparent handling of high-speed cluster interconnects through a single MPI library
- More efficient cluster management through better control over the application runtime environment
- Increased fault resilience in case of network and node failures
- Higher productivity with Linux Clusters through professional service and support