Dynamic Balancing Complex Workload in Workstation Networks - Challenge, Concepts and Experience

Wolfgang Becker
Institute of Parallel and Distributed High-Performance Systems (IPVR)
University of Stuttgart, Germany
email: wbecker@informatik.uni-stuttgart.de

Abstract

Workstation clusters are being recognized as the main promising computing resource of the near future. A large size workstation cluster, consisting of locally connected workstations, has the power comparable to a supercomputer, at a fraction of the cost. Further, a wide area coupling of workstation clusters is not only suitable for exchange of mail and news or establishment of distributed information systems, but can also be exploited as a large metacomputer. The wide area distribution aspects will be covered in a separate paper by the E=MC^2 project [8]. This paper shows the potential power by characterizing the system and the needs of current applications, and outlines the general idea to efficiently utilize networks of workstation. The second part of the paper introduces the approach of the HiCon project to solve the operating system and programming environment problems that currently restrict proper exploitation of workstation clusters, and demonstrates the feasibility by real measurement results. It concludes with general results for the research community in this area.

1 The Challenge

Workstations offer high computing performance at lower cost than mainframes, but nevertheless their operating systems support multiple users, multitasking and networking and promise application portability. They can be used as clients and as servers as well. Client - server computing is further encouraged by multithreading and symmetric multiprocessing. However, already within one cluster, workstations usually differ significantly in CPU speed, main memory, secondary storage capacity and in architecture. Workstation clusters are shared nothing parallel systems, connected by LANs with low bandwidth and high latency - compared to the local processing power; Different and distant clusters are coupled by WANs that have even lower bandwidth and higher latencies by orders of magnitude.

As these systems begin to be accepted by research centers and industry, their usage patterns will change towards less various, but more resource intensive, mission critical, large applications. In the scientific area, numerical simulations and image processing will be main challenges, while in the commercial area large distributed databases and information services have to be supported. These application types will have to be decomposed and distributed across the workstation clusters and have to use the resources concurrently.

Currently, workstation clusters are utilized by at most 10% on average; High speed networks will soon be available, enabling better sharing of distributed resources. There is no single operating system image and no primary support for parallel executions or load balancing. The goal of load balancing within workstation clusters is to maximally exploit the huge aggregated processing capacity by automatic task assign-

Proceedings High Performance Computing and Networking (HPCN) Europe
Lecture Notes on Computer Science (LNCS), Springer Verlag, 1995

ment or shifting of workload. Matching the different real world requirements by automatic dynamic, application independent load balancing is a major research topic:

- Loosely coupled parallel systems and computer networks are rarely fully loaded, while it appears frequently that some of the nodes are overloaded while others are idle. Simple automatic load distribution mechanisms achieve a more equalized resource utilization by migrating tasks from overloaded or assigning tasks to underloaded nodes. A node's load usually is just the queue size of runnable processes.
- In real computing centers, heterogeneous grown up systems can be found, consisting of faster and slower processors. Here, load balancing has to take into account that faster nodes yield the same response times at more workload; It may even be better to sometimes leave less powerful nodes idle.
- Real applications' tasks are heterogeneous. Even within one large parallel application the task profiles are different, depending on runtime parameters. Hence, nodes have to be considered as more or less loaded, depending on the tasks' resource demands. Nodes are occupied for shorter or longer time, so further tasks will have to wait there or will get an according share of the resources only.
- Expensive systems or clusters of autonomous nodes are usually not only used by several independent sequential tasks, but also heterogeneous mixes of parallel applications are executing concurrently. Tasks within complex applications are correlated and interdependent; tasks on critical paths and tasks entailing large parallelism must be prioritized, for they determine the overall execution time between synchronization points, and resources can be maximally utilized then.
- Tasks access global data which can be located remotely; tasks within one application cooperate by data communication. Hence, buffers of persistent data, intermediate results, and other shared objects have to be sent across the network and task response time depends significantly on the location of the data, i.e. whether data are locally available, whether communication can be performed locally or not. Load balancing should avoid unnecessary network load and task execution delays due to data communication.
- Other boundary conditions and effects significantly affect the performance of parallel systems. For example, node performance depends on the load: many parallel processes cause context switch overhead and usually extensive paging due to main memory congestion. Overloaded networks or congested load balancing components cause additional overhead and delays. Hence, a suitable degree of system resource exploitation and appropriate load balancing efforts must be adjusted.

Existing approaches usually cover fractions of these aspects, while the HiCon concept is designed to manage all these real world requirements. Complex dynamic adaptive assignment algorithms, considering data affinity, were developed for database transaction routing [11], however they are not generally applicable. Decentralized scalable approaches [7], [9] tend to non-coherent decisions and often have too simple load/execution models. Workstation load sharing environments [6], [10] also employ simple decision models and focus on transparently stealing CPU cycles from nodes that are currently not used interactively.

2 Concepts

The HiCon concept [3], [4] was developed to provide efficient automatic load distribution in the domain described above: advanced dynamic and adaptive task scheduling and placement of large parallel and heterogeneous concurrent applications based on the client - server model. The computing resource consists of heterogeneous, arbitrarily connected clusters of workstations. Servers are configured on processors and receive tasks from clients which drive applications; they operate on global shared, volatile or persistent data as well as on common data within applications. Data are moved and copied among the nodes on demand by a runtime environment. Fig. 1 gives a survey of the components and their interaction within a HiCon cluster.

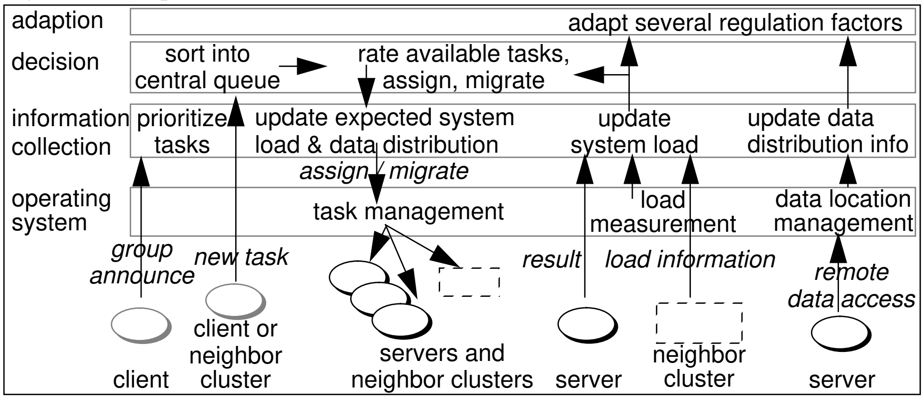

Fig. 1 HiCon load balancing, system an application architecture per cluster.

Load balancing operates as a rather sophisticated central agent per cluster, while the agents of neighbored clusters equalize their load by a simple distributed policy. This yields optimal decisions within clusters but retains scalability [2]. Within each cluster tasks are queued centrally and assigned arbitrarily to server-local queues, between clusters tasks are exchanged from/to central queues. Task queueing enables load control, which is necessary because the nodes are sensitive to high load factors due to context switches and overflow of active memory.

HiCon load balancing is application independent. The goal is overall throughput maximization and task response time minimization. Applications can support load balancing by dynamic estimations of task size and data reference patterns. Even critical paths within small task groups can be recognized [1]. Load balancing considers not only processing demands and processor load factors, but also data affinity and data communication costs for task placement. Finally, the HiCon model employs several adaption techniques for dynamic regulation of inaccurate or missing pre-estimations and of heuristic parameters in the decision model, and also adjusts its relationship between overhead and profit [5].

The HiCon decision algorithm basically reacts on system state change events by rating the available tasks in the central queue and assigning them to their favorite processor, as long as the processor does not become overloaded in the near future. The best processor for a task is usually the one promising the shortest response time: HiCon load balancing estimates the sum of the expected compute time under current load, the expected data communication time according to data reference estimations and current data distribution, and the wait time if the servers on that processor are busy. In heavy load situations the balancing criterion is shifted towards throughput optimization, i.e. increased response times of single tasks are tolerated in order to reduce communication efforts and processor idle times. The informations used for these placement decisions are system load measurements and extrapolations as well as task profile assumptions provided by clients at call time.

3 Experiences

For evaluation of the concepts a prototype environment has been implemented, and a wide spectrum of applications has been investigated: heterogeneous mixes of parallelized complex applications like image recognition, finite element analysis and relational database processing can be executed on arbitrary workstation networks. Following four measurements shall briefly show the main features and verify the flexibility and applicability of the concepts:

3.1 Appropriate Distribution of Parallel Applications and Multiuser Concurrence

The first measurement observes three concurrent parallel finite element analysis computations. Fig. 2 shows the typical execution profile of this application type and the trial configuration. A static data partitioning of the tasks, where each processor performs the calculations for a certain element range or vector row range, suffers from load imbalance and idle times at the end of each iteration. HiCon load balancing is able to better adapt the parallel execution and enable suitably meshed concurrent processing, by considering processing capacities and instant task load due to multiuser operation. Sophisticated load balancing is also better than simply assigning available tasks to the first idle server, mainly because the load control mechanism provides optimum resource usage even in situations of heavy load in the system.

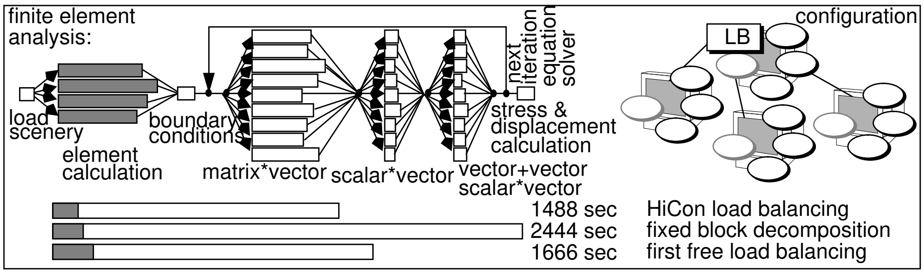

Fig. 2 Advanced load balancing for managing three parallel finite element calculations.

3.2 Matching Trade-off Between CPU Utilization and Communication Overhead

The second measurement looks at a single, parallel image recognition application, which consists of different phases with varying task profiles and execution profile structures (Fig. 4). In this application even small tasks operate on large sets of common data, where the reference patterns and task sizes are not static but depend heavily on the actual image structure. HiCon load balancing is able to consider communication cost due to cooperation and access of common data, and tries to match the trade-off between utilization of CPU cycles and communication overhead. HiCon load balancing performance is compared to a strategy that cares of CPU utilization only (Fig. 4).

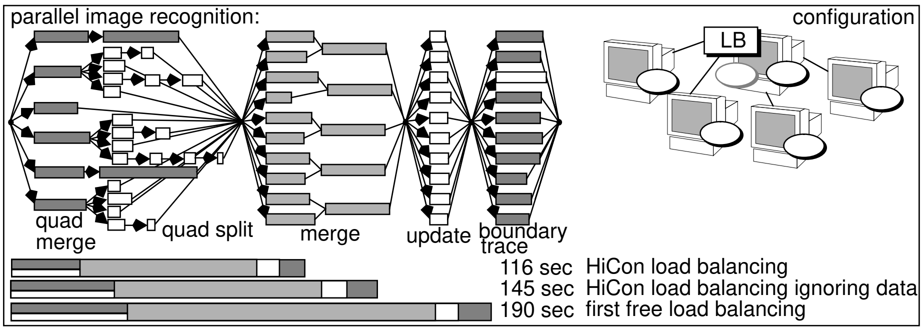

Fig. 3 Load balancing considering communication to manage parallel image recognition.

3.3 Scalability by Decentralized Inter-Cluster Load Sharing

The last trial shows a network of 28 servers under heavy concurrent application load by 9 parallel image recognition applications, under different load balancing control structures (Fig. 5). While the completely centralized structure suffers from congestion of the load balancing component, the completely decentralized structure had not enough information and overview to achieve a good workload distribution, and was unable to suitably exploit application internal parallelism. Hence, the HiCon intra cluster - inter cluster concept is successful and naturally fits into the network topology.

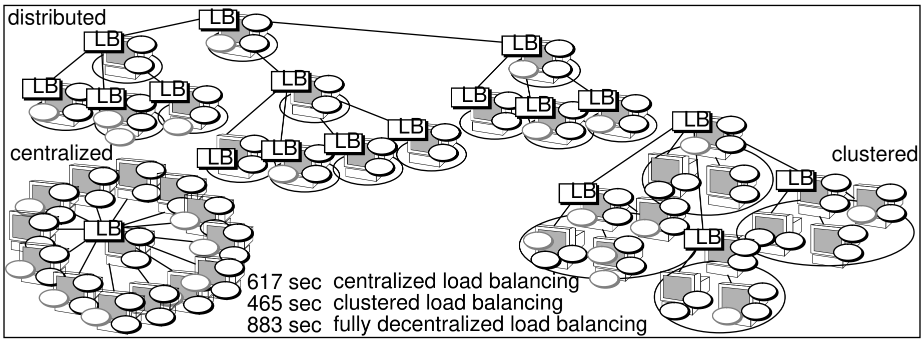

Fig. 4 Clustering structures for load balancing large, heavily load workstation networks.

In wide area connected clusters, where networks show significantly reduced bandwidth and increased latency, a suitable clustering concept is even more important. Local load balancing can manage accurate assignments and suitable parallel execution within applications, but between distant clusters only rough, coarse grained load equalization is feasible. The E=MC^2 project evaluates these issues [8].

4 Conclusions

In summary, the results from the HiCon project lead to the following conclusions of common interest.

For development of load balancing concepts for large distributed systems, not only scalability should be considered: centralized advanced load balancing has strong advantages compared to simple, distributed policies. These advantages will appear as soon as realistic heterogeneous system configurations and workload from more productional environments like research or industrial computing centers, are addressed. Results from former static scheduling approaches and transaction routing techniques from data processing may be integrated.

Upcoming high speed connections for wide area networks enable more fine grained and dynamic load sharing and better global resource utilization. It shifts the trade-off point between parallelism and data distribution and the inferred communication and synchronization efforts. However, existing load sharing facilities are still unsuitable for this challenge, and latency turns out to be a major limiting factor for distributed parallel computing. Load balancing has to consider this appropriately.

Simple but general concepts to integrate data communication, remote data access and synchronization into the load balancing model, are inevitable for distributed systems and non-trivial applications. The HiCon concept just showed one approach by explicitly observing access patterns and locations of global shared data, which proved to be appropriate for a wide range of applications.

Overall, the HiCon project demonstrates that it is feasible to automatically optimize the resource usage within heterogeneous parallel and distributed systems even by concurrent parallelized real world applications.

References

1. W. Becker, G. Waldmann, *Exploiting Inter Task Dependencies for Dynamic Load Balancing*, IEEE Int. Symp. High-Performance Distributed Computing (HPDC), San Francisco, 1994
2. W. Becker, J. Zedelmayr, *Scalability and Potential for Optimization in Dynamic Load Balancing - Centralized and Distributed Structures*, Mitteilungen GI, Parallele Algorithmen und Rechnerstrukturen (PARS), GI/ITG Workshop Potsdam, 1994
3. W. Becker, *Das HiCon-Modell: Dynamische Lastverteilung für datenintensive Anwendungen auf Rechnernetzen*, Informatik Forschung und Entwicklung Vol. 10 No. 1, Springer Verlag, 1995
4. W. Becker, *Lastverteilung in Workstation-Netzen*, BI Sonderheft Paralleles Rechnen, RUS, Universität Stuttgart, 1995
5. W. Becker, G. Waldmann, *Adaption in Dynamic Load Balancing: Potential and Techniques*, Tagungsband 3. Fachtagung Arbeitsplatz-Rechensysteme (APS), Hanover, 1995
6. F. Douglis, J. Ousterhout, *Transparent Process Migration: Design Alternatives and the Sprite Implementation*, Software-Practice and Experience Vol. 21 No. 8, 1991
7. D. Eager, E. Lazowska, J. Zahorjan, *A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Sharing*, Performance Evaluation Vol. 6, 1986
8. P. Huish (Ed.), *European Meta Computing Utilising Integrated Broadband Communications - Interim Report*, Deliverable CEC Project B2010 TEN-IBC E=MC^2, 1995
9. F. Lin, R. Keller, *The Gradient Model Load Balancing Method*, IEEE Transactions on Software Engineering Vol. 13 No. 1, 1987
10. M. Litzkow, M. Livny, M. Mutka, *Condor - A Hunter of Idle Workstations*, Int. Conf. on Distributed Computing Systems, San Jose, 1988
11. P. Yu, A. Leff, Y. Lee, *On Robust Transaction Routing and Load Sharing*, ACM Transactions on Database Systems Vol. 16 No. 3, 1991