

Make a donation to Wikipedia and give the gift of knowledge!

Computer cluster

From Wikipedia, the free encyclopedia

A **computer cluster** is a group of tightly coupled computers that work together closely so that in many respects they can be viewed as though they are a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks. Clusters are usually deployed to improve performance and/or availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.^[1]



An example of a Computer cluster

Contents

- 1 Cluster categorizations
 - 1.1 High-availability (HA) clusters
 - 1.2 Load-balancing clusters
 - 1.3 High-performance computing (HPC) clusters
 - 1.4 Grid computing
- 2 Implementations
- 3 History
- 4 Technologies
- 5 See also
- 6 References
 - 6.1 Further reading
- 7 External links

Cluster categorizations

High-availability (HA) clusters

High-availability clusters (also known as failover clusters) are implemented primarily for the purpose of improving the availability of services which the cluster provides. They operate by having redundant nodes, which are then used to provide service when system components fail. The most common size for an HA cluster is two nodes, which is the minimum requirement to provide redundancy. HA cluster implementations attempt to manage the redundancy inherent in a cluster to eliminate single points of failure. There are many commercial implementations of High-Availability clusters for many operating systems. The Linux-HA project is one commonly used free software HA package for the Linux OSs

Load-balancing clusters

Load-balancing clusters operate by having all workload come through one or more load-balancing front ends, which then distribute it to a collection of back end servers. Although they are primarily

implemented for improved performance, they commonly include high-availability features as well. Such a cluster of computers is sometimes referred to as a server farm. There are many commercial load balancers available including Platform LSF HPC, Sun Grid Engine, Moab Cluster Suite and Maui Cluster Scheduler. The Linux Virtual Server project provides one commonly used free software package for the Linux OS.

High-performance computing (HPC) clusters

High-performance computing (HPC) clusters are implemented primarily to provide increased performance by splitting a computational task across many different nodes in the cluster, and are most commonly used in scientific computing. Such clusters commonly run custom programs which have been designed to exploit the parallelism available on HPC clusters. HPCs are optimized for workloads which require jobs or processes happening on the separate cluster computer nodes to communicate actively during the computation. These include computations where intermediate results from one node's calculations will effect future calculations on other nodes.

One of the most popular HPC implementations is a cluster with nodes running Linux as the OS and free software to implement the parallelism. This configuration is often referred to as a Beowulf cluster.

Microsoft offers Windows Compute Cluster Server as a high-performance computing platform to compete with Linux.^[2]

Many software programs running on High-performance computing (HPC) clusters use libraries such as MPI which are specially designed for writing scientific applications for HPC computers.

Grid computing

Grid computing or grid clusters are a technology closely related to cluster computing. The key differences (by definitions which distinguish the two at all) between grids and traditional clusters are that grids connect collections of computers which do not fully trust each other, or which are geographically dispersed. Grids are thus more like a computing utility than like a single computer. In addition, grids typically support more heterogeneous collections than are commonly supported in clusters.

Grid computing is optimized for workloads which consist of many independent jobs or packets of work, which do not have to share data between the jobs during the computation process. Grids serve to manage the allocation of jobs to computers which will perform the work independently of the rest of the grid cluster. Resources such as storage may be shared by all the nodes, but intermediate results of one job do not affect other jobs in progress on other nodes of the grid.

An example of a very large cluster is the Folding@home project. It is analyzing data that is used by researchers to find cures for diseases such as Alzheimer's and cancer. Another large project is the SETI@home project, which may be the largest distributed cluster in existence. It uses approximately three million home computers all over the world to analyze data from the Arecibo Observatory radiotelescope, searching for evidence of extraterrestrial intelligence.

Implementations

The TOP500 organization's semiannual list of the 500 fastest computers usually includes many clusters. TOP500 is a collaboration between the University of Mannheim, the University of Tennessee,

and the National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory. As of November 2006, the top supercomputer is the Department of Energy's IBM BlueGene/L system with performance of 280.6 TFlops.

Clustering can provide significant performance benefits versus price. The System X supercomputer at Virginia Tech, the 28th most powerful supercomputer on Earth as of June 2006[1] (<http://www.top500.org/list/2006/06/100>) , is a 12.25 TFlops computer cluster of 1100 Apple XServe G5 2.3 GHz dual-processor machines (4 GB RAM, 80 GB SATA HD) running Mac OS X and using InfiniBand interconnect. The cluster initially consisted of Power Mac G5s; the rack-mountable XServes are denser than desktop Macs, reducing the aggregate size of the cluster. The total cost of the previous Power Mac system was \$5.2 million, a tenth of the cost of slower mainframe computer supercomputers. (The Power Mac G5s were sold off.)

The central concept of a Beowulf cluster is the use of commercial off-the-shelf computers to produce a cost-effective alternative to a traditional supercomputer. One project that took this to an extreme was the Stone Soupercomputer.

However it is worth noting that FLOPs (floating point operations per second), aren't always the best metric for supercomputer speed. Clusters can have very high FLOPs, but they cannot access all data the cluster as a whole has at once. Therefore clusters are excellent for parallel computation, but much poorer than traditional supercomputers at non-parallel computation.

JavaSpaces is a specification from Sun Microsystems that enables clustering computers via a distributed shared memory.

History

The history of cluster computing is best captured by a footnote in Greg Pfister's *In Search of Clusters*: "Virtually every press release from DEC mentioning clusters says 'DEC, who invented clusters...'. IBM did not invent them either. *Customers* invented clusters, as soon as they could not fit all their work on one computer, or needed a backup. The date of the first is unknown, but it would be surprising if it was not in the 1960s, or even late 1950s."

The formal *engineering* basis of cluster computing as a means of doing parallel work of any sort was arguably invented by Gene Amdahl of IBM, who in 1967 published what has come to be regarded as the seminal paper on parallel processing: Amdahl's Law. Amdahl's Law describes mathematically the speedup one can expect from parallelizing any given otherwise serially performed task on a parallel architecture. This article defined the engineering basis for both multiprocessor computing and cluster computing, where the primary differentiator is whether or not the interprocessor communications are supported "inside" the computer (on for example a customized internal communications bus or network) or "outside" the computer on a *commodity* network.

Consequently the history of early computer clusters is more or less directly tied into the history of early networks, as one of the primary motivation for the development of a network was to link computing resources, creating a de facto computer cluster. Packet switching networks were conceptually invented by the RAND corporation in 1962. Using the concept of a packet switched network, the ARPANET project succeeded in creating in 1969 what was arguably the world's first commodity-network based computer cluster by linking four different computer centers (each of which was something of a "cluster" in its own right, but probably not a *commodity* cluster). The ARPANET project grew into the Internet -- which can be thought of as "the mother of all computer clusters" (as the union of nearly all of the compute resources, including clusters, that happen to be connected). It also established the

paradigm in use by *all* computer clusters in the world today -- the use of packet-switched networks to perform interprocessor communications between processor (sets) located in otherwise disconnected frames.

The development of customer-built and research clusters proceeded hand in hand with that of both networks and the Unix operating system from the early 1970s, as both TCP/IP and the Xerox PARC project created and formalized protocols for network-based communications. The Hydra operating system was built for a cluster of DEC PDP-11 minicomputers called C.mmp at C-MU in 1971. However, it was not until circa 1983 that the protocols and tools for *easily* doing remote job distribution and file sharing were defined (largely within the context of BSD Unix, as implemented by Sun Microsystems) and hence became generally available commercially, along with a shared filesystem.

The *first* commercial clustering product was ARCnet, developed by Datapoint in 1977. ARCnet was not a commercial success and clustering per se did not really take off until DEC released their VAXcluster product in the 1984 for the VAX/VMS operating system. The ARCnet and VAXcluster products not only supported parallel computing, but also shared file systems and peripheral devices. The idea was to provide the advantages of parallel processing, while maintaining data reliability and uniqueness. VAXcluster, now VMScluster, is still available on OpenVMS systems from HP running on Alpha and Itanium systems.

Two other noteworthy early commercial clusters were the *Tandem Himalaya* (a circa 1994 high-availability product) and the *IBM S/390 Parallel Sysplex* (also circa 1994, primarily for business use).

No history of commodity computer clusters would be complete without noting the pivotal role played by the development of Parallel Virtual Machine (PVM) software in 1989. This open source software based on TCP/IP communications enabled the *instant* creation of a virtual supercomputer -- a high performance compute cluster -- made out of any TCP/IP connected systems. Free form heterogeneous clusters built on top of this model rapidly achieved total throughput in FLOPS that greatly exceeded that available even with the most expensive "big iron" supercomputers. PVM and the advent of inexpensive networked PCs led, in 1993, to a NASA project to build supercomputers out of commodity clusters. In 1995 the invention of the "beowulf"-style cluster (<http://www.beowulf.org/overview/history.html>) -- a compute cluster built on top of a commodity network for the specific purpose of "being a supercomputer" capable of performing tightly coupled parallel HPC computations. This in turn spurred the independent development of Grid computing as a named entity, although Grid-style clustering had been around at least as long as the Unix operating system and the Arpanet, whether or not it, or the clusters that used it, were named.

Technologies

MPI is a widely-available communications library that enables parallel programs to be written in C, Fortran, Python, OCaml, and many other programming languages.

The GNU/Linux world sports various cluster software; for application clustering, there is Beowulf, distcc, and MPICH. Linux Virtual Server, Linux-HA - director-based clusters that allow incoming requests for services to be distributed across multiple cluster nodes. MOSIX, openMosix, Kerrighed, OpenSSI are full-blown clusters integrated into the kernel that provide for automatic process migration among homogeneous nodes. OpenSSI, openMosix and Kerrighed are single-system image implementations.

Microsoft Windows Compute Cluster Server 2003 based on the Windows Server platform provides pieces for High Performance Computing like the Job Scheduler, MSMPI library and management tools. NCSA's recently installed Lincoln is a cluster of 450 Dell PowerEdge™ 1855 blade servers running Windows Compute Cluster Server 2003. This cluster debuted at #130 on the Top500 list in June 2006.

DragonFly BSD, a recent fork of FreeBSD 4.8 is being redesigned at its core to enable native clustering capabilities. It also aims to achieve single-system image capabilities.

See also

- Distributed data store
- Flash mob computing
- Peer-to-peer
- Symmetric multiprocessing
- Two-node cluster
- Computer cluster in virtual machines

References

1. ^ Bader, David; Robert Pennington (June 1996). Cluster Computing: Applications (<http://www.cc.gatech.edu/~bader/papers/ijhpc.html>) . Georgia Tech College of Computing. Retrieved on 2007-07-13.
2. ^ "Windows invades Linux territory" (Roger Howorth, IT Week, 12 Apr 2006) (<http://www.itweek.co.uk/itweek/news/2153996/windows-invades-linux-territory>) . Retrieved on 2007-07-13.

Further reading

- Karl Kopper: *The Linux Enterprise Cluster: Build a Highly Available Cluster with Commodity Hardware and Free Software*, No Starch Press, ISBN 1-59327-036-4
- Evan Marcus, Hal Stern: *Blueprints for High Availability: Designing Resilient Distributed Systems*, John Wiley & Sons, ISBN 0-471-35601-8
- Greg Pfister: *In Search of Clusters*, Prentice Hall, ISBN 0-13-899709-8
- Rajkumar Buyya (editor): *High Performance Cluster Computing: Architectures and Systems*, Volume 1, ISBN 0-13-013784-7, Prentice Hall, NJ, USA, 1999.
- Rajkumar Buyya (editor): *High Performance Cluster Computing: Programming and Applications*, Volume 2, ISBN 0-13-013785-5, Prentice Hall, NJ, USA, 1999.

External links

- IEEE task force on cluster computing (<http://www.ieeetfcc.org/>)



Topics in parallel computing

General High-performance computing
Parallelism Data parallelism • Task parallelism

Theory	Speedup • Amdahl's law • Flynn's taxonomy (SISD, SIMD, MISD, MIMD) • Cost efficiency • Gustafson's law • Karp-Flatt metric
Elements	Process • Thread • Fiber • Parallel Random Access Machine
Coordination	Multiprocessing • Multithreading • Multitasking • Memory coherency • Cache coherency • Barrier • Synchronization • Distributed computing • Grid computing
Programming	Programming model • Implicit parallelism • Explicit parallelism
Hardware	Computer cluster • Beowulf • Symmetric multiprocessing • Non-Uniform Memory Access • Cache only memory architecture • Asymmetric multiprocessing • Simultaneous multithreading • Shared memory • Distributed memory • Massively parallel processing • Superscalar processing • Vector processing • Supercomputer • Stream processing • GPGPU
Software	Distributed shared memory • Application checkpointing • Warewulf
APIs	POSIX Threads • OpenMP • Message Passing Interface (MPI)
Problems	Embarrassingly parallel • Grand Challenge • Software lockout

Retrieved from "http://en.wikipedia.org/wiki/Computer_cluster"

Categories: Cluster computing | Parallel computing | Concurrent computing | Supercomputers | Local area networks

-
- This page was last modified 16:49, 5 October 2007.
 - All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.)

Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a U.S. registered 501(c)(3) tax-deductible nonprofit charity.