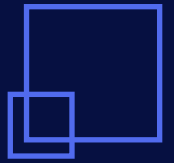


Clustrx®

**Operating
System**

Summer 2010



High-Performance Computing: Peta to Exascale

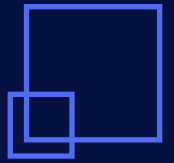
The next decade will mark a resurgence for parallel computing, with high-end systems moving to exascale and others massively moving to multicore parallel architectures and multiserver virtualization in datacenters. Unlike the previous smooth run of the HPC evolution, this shift will have a strong impact on the existing system software.

Major players in the HPC market have acknowledged the importance and difficulty of providing a sound viable business model of sales and support for the entire range of supercomputing systems which will enter the market during the next decade, be those the upcoming mainstream mini-terascale systems, petascale mid-range ones, or future exascale monsters with still undefined architectures and applicability. One certainty is that the HPC market will have to welcome the exascale monsters in the next few years, so initiatives like IESP (International Exascale Project) seem to have emerged right in time.

As HPC enters the petascale era, there will be a number of challenges to overcome before applications can take advantage of the upcoming new level of computational power. One of the most pressing challenges will be to re-design system software that will fit well into heterogenous petascale architectures to allow end-users to solve their everyday or previously unattainable scientific and business problems without painstakingly trying to accommodate themselves to a great variety of approaches which will be used to bring petascale to reality. Vendors' yearning for a leadership on the petaflops market that has been just born and continuous attempts to increase the overall productivity of each system by including computational accelerators inspired the design of an operating system that should become a firstling of the next generation of system software for HPC clusters.

For petascale/exascale, it is widely recognized that past approaches based on the reuse of existing toolsets previously developed for server versions of UNIX or Windows will not effectively scale to manage all power available in future high-end systems. The survivors in the next HPC years will be companies which have recognized today's challenges of the oncoming changes and successfully adapted their business for an efficient and cost-effective development of the new generation of hardware (new interconnects, task-specific accelerators, etc), system software and application software that will employ all the huge power of the future exascale computing for the common good.

A conclusion was made in the beginning of the OS Clustrx development: a petascale system software newcomer should use proven, well-tested technologies in every part where they can serve requirements, but other parts should be replaced by functional analogues which will scale up to exascale. This approach is a natural way to ensure a painless migration to petascale architectures.



Working Comprehensively

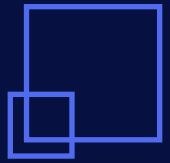
Clustrx is an operating system (OS) for high-performance computing (HPC) that takes up several novel approaches, both to serve the current computing needs and to scale smoothly onto future systems. The philosophy that ensures the next-generation scalability and fidelity of this OS is based on viewing an HPC cluster as a single supercomputing machine. The approach demands that an OS should cover the whole infrastructure of a cluster, of any complexity and heterogeneity – and Clustrx does. It represents a supercomputing system as a “blackbox” that aggregates the computing power of a great number of nodes (that can be totally heterogeneous in hardware and system platforms) into a comprehensible and scalable service which can be deployed, controlled, and distributed from a single point. This aggregated view is a critical step ahead from the “loose” view of a computing cluster as a set of more or less similar nodes with interconnections.

In a personal computer system, the operating system and applications share a single processor. However, as more and more processors and other hardware are gathered into a distributed computing system, this approach becomes invalid: now a great deal of hardware resources are shared between the operating system and application software in an increasingly complex manner. The complexity can grow infinitely, but, as the system has to remain manageable, we can take advantage of another level of organization: where application tasks are performed by one part of the cluster while the operating system uses another to provide control, both parts containing a lot of shared hardware resources.

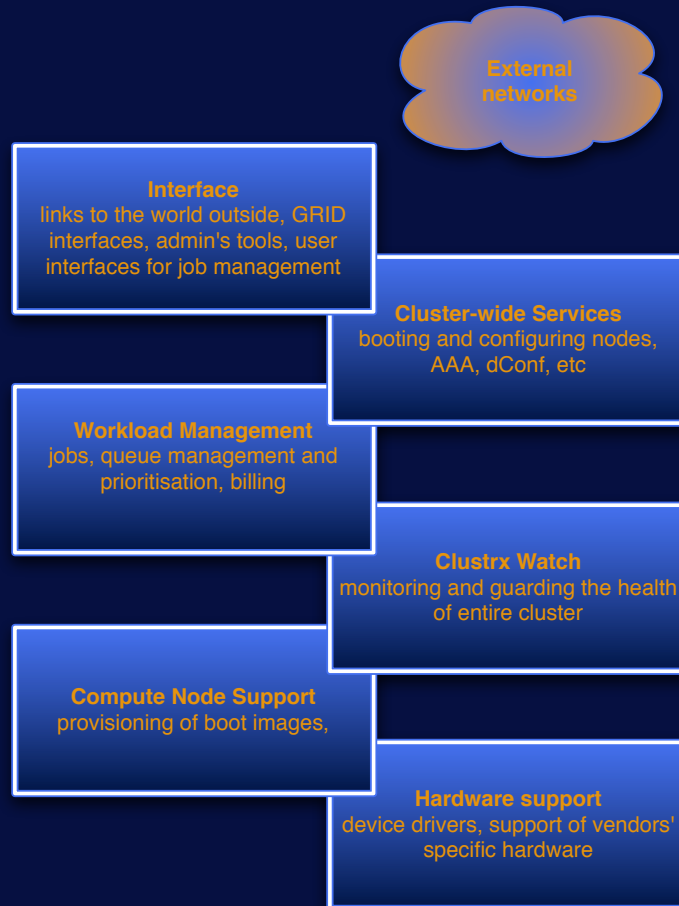
The key to the proper use of a united computing power is hierarchy. The primary level of it is dividing the cluster into a part that consists of compute nodes and one that represents an hierarchical management network. The latter is also a cluster that covers all management needs in a unified and scalable manner and is based on further hierarchies.

This is the approach that Clustrx uses to make a cluster system of any internal complexity look like a single computing device. In addition to scalability, it ensures a smooth and fault-free operation because hardware and software resources are redistributed by the system’s infrastructure transparently for the user, to achieve a stable and safe operation. This requires Clustrx to support all levels of infrastructure, from bootable operating systems on compute nodes to most abstracted interfaces – which it does. Importantly enough, the whole suite can be deployed in a very reasonable time: several hours between the hardware installation and the launching of computing jobs, the deployment being a unified procedure controlled from a single point and not requiring a lot of qualified personnel.

The hierarchical complexity of the Clustrx structure and workflow calls for multiple models through which to explain features. Each model will present either a particular task-driven look at the system from a certain logical focus or a multilayered architectural view.



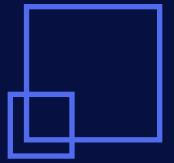
Layering the responsibility



Being able to see Clustrx-based cluster as a single supercomputing machine is achieved by using multiple layers of organization (see figure above). The Hardware Support is provided by Clustrx for an extensive (and extensible) list of vendors and devices. Linux is used as a basis to control the hardware. It supports a lot of devices at the kernel level, and it is open to add-ons developed both in-house and by third parties. There is a choice of MPI and RDMA implementations optimized for popular HPC interconnects. A single-package repository is used for all supported hardware architectures.

The Compute Node Support layer deals with an efficient boot and performance. Clustrx can use any combination of bootable OSs on compute nodes; any one can be booted either from disk or in a remote diskless fashion. The list includes a variety of HPC-oriented Linux, Windows HPC Server, and the natively developed Clustrx CNL (Compute Node Linux). The latter is based on Linux with a number of optimizations, patches etc. developed to deliver a highest possible computing performance. (Any boot image can be easily built and stored for further remote usage.)

Following the approach of fully supporting heterogeneity, application tasks launched in the system may run either under their native OS (they have been compiled in) or under



Clustrx CNL that supports a binary compatibility layer to run legacy 32-bit /64-bit and RHEL-based application software.

A set of kernels that allow the choice of memory managers (manageable by the cluster's resource manager) and CPU schedulers are used to achieve a maximum performance for a specific task. Features of Clustrx can be used to build a high-performance fault-tolerant disk storage that supports various parallel filesystems.

The backbone of the Clustrx-driven computing cluster is its monitoring, management, and control system, Clustrx Watch. This is a novel cluster-wide monitoring system capable of surveying millions of checkpoints (hardware sensors, SNMP datasources, traps, kernel and software metrics) in nearly real time, while scaling linearly. It features an hierarchical architecture of service data collection, aggregation, distribution, processing and logging that has been engineered to serve multi-petascale systems.

Clustrx Watch includes an advanced power manager integrated with its resource manager. Any unused hardware can be switched off quickly.

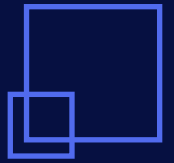
An emergency shutdown system guards hardware against critical failures when the cooling or power crashes. Nodes that the monitoring system resides on are mutually replaceable, i.e. as soon as some of those are found in trouble, they will exchange their roles smartly, smoothly and transparently. This contributes to an unbreakable architecture without a single point of failure.

The Task Management layer is connected with and relies largely on the monitoring system. Its job is to launch computing tasks and track their execution on compute nodes. If some of the nodes are found in a critical state by Clustrx Watch, the two layers work jointly to redistribute the computing load between the nodes.

This level features an integrated resource accounting automated by scripts. It can summarize resource usage logs saved by Clustrx Watch and send the summary outside for the purpose of precise billing.

The Services layer abstracts the cluster as a toolset of integrated services for the ease of administration. Most important services include AAA (user accounts, authentication, authorization), a highly controllable booting/configuring of compute nodes, a single configuration database named dConf (Distributed Configuration) that allows a customized access.

The Interface layer provides users with tools to add their computing jobs to the queue and monitor their execution. This layer includes communication links to the outside world that can be used for integrating the cluster into a GRID. The administrator has a unified point of access and control here. One of valuable administration tools is a cluster map that presents a real-time visualization of the state of all cluster objects to facilitate the administrative decision making.



Scaling Made Easy

The Clustrx architecture is designed to ensure manageability of clusters of any size and heterogeneity, thereby addressing the petascale and upcoming exascale challenges. The linear scalability is achieved by dividing the cluster into two distinct parts: a management network that runs OS services and a computing network that runs user jobs.

Multiple approaches are used to improve manageability of distributed services as the number of nodes grows. The administrator can use open APIs to control OS services at various levels. To facilitate the distributed administration, OS contains a rule-based engine that helps formulate complex administration tasks launched automatically as needed.

The management cluster gathers a lot of monitoring information from the nodes in real time. It creates an extensive data flow through the management cluster that sometimes can overload the system's bandwidth. To avoid clogging the management cluster, in case it is very large and data-intensive, an hierarchy of transit monitoring nodes is introduced. The transit nodes gather and process intermediate data, make decisions at their level of responsibility, and pass only important summaries of the data on to the next level of data aggregation. The hierarchy can be of any depth thus enabling the monitoring of millions of compute nodes. Note that all hardware components, including SNMP equipment and data storage, are monitored in the unified way.

UNIX daemons that were suspected to cause bottlenecks were replaced by rewritten scalable functional analogues and fully adapted to work in a dynamic distributed environment. To reduce the traffic and achieve a scalability and reliability appropriate for a management network, Clustrx has a transparent mechanism to migrate functional units between the management nodes not interrupting any service. (On small clusters, a single server node can be used to provide all functionality of the management network.)

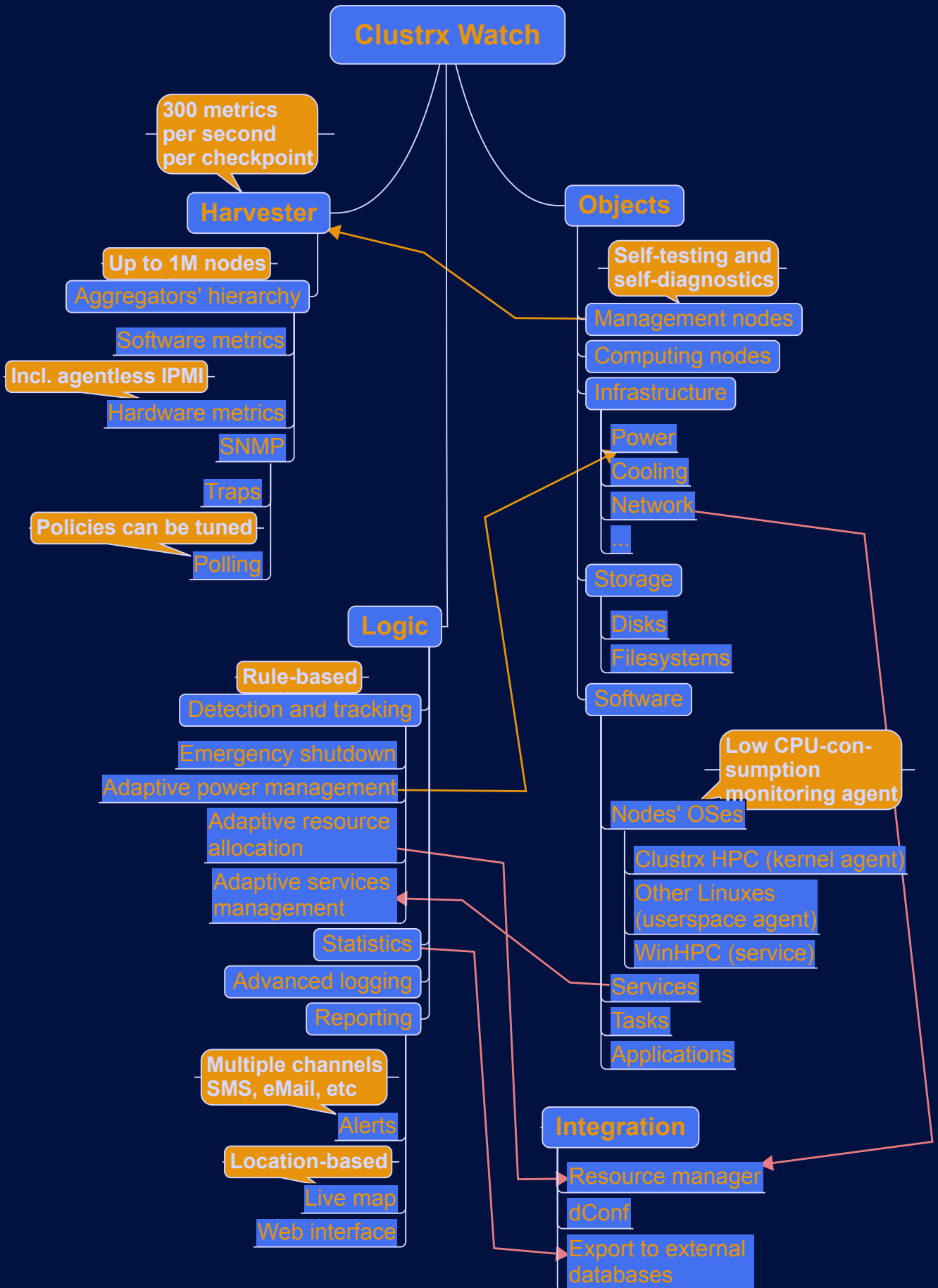
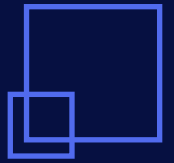
The management cluster uses a single distributed hierarchic database of configuration information – dConf (Distributed Configuration) – that stores all configuration of both management and compute nodes. All OS services interact with this database.

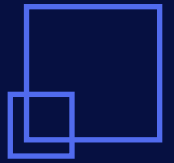
In addition, dConf is accessible by user applications via network and open APIs. Both the administrator and the users are allowed to write custom scripts to control any amount of operations on any number of compute nodes. Large trunks of data in dConf can be modified via transactions; the support for this is provided.

Watching Out

One of most important jobs of the management cluster is to monitor the health of the whole computer cluster and make automated decisions to save the hardware from physical damage due to a cooling or power crash. A distributed subsystem responsible

Massive Solutions





for this is called Clustrx Watch. Together with dConf, it is a spine of the whole cluster. Its philosophy goes beyond guarding the physical health – it is a sophisticated control and decision-making system. Therefore it will be the next focus to view the whole system from.

Clustrx Watch controls an extensive set of hardware and software resources that keep up power supply, cooling, and network infrastructure to ensure both the physical safety of the cluster's devices and the successful completion of computing jobs. To permit the monitoring of their safety, compute nodes run software agents. (Clustrx CNL has one built in its Linux kernel.) The agents are designed to minimize the processor time consumption. Some components of the system work with the support of SNMP (Simple Network Management Protocol), and there the agents are not needed.

Each monitoring checkpoint (an agent, an SNMP source) gives a few hundreds indications per second on one node. The data flow to decision-making management nodes via the hierarchy of transit nodes (see figure above) which do intermediate processing. They are used to perform an analysis and make critical decisions such as the emergency shutdown of an overheated node, reboot, hibernation etc. The processed data of monitoring are stored and can be used for further statistical analysis.

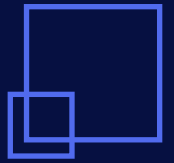
Clustrx Watch has access to the rule-based engine that facilitates automated response to events occurring on the cluster. The power and other resource consumption policies are also defined by rules that can be modified. Using sets of heuristics helps tackle the complexity and subtlety of the management. In particular, deciding on the emergency shutdown of a compute node requires a communication with the task and resource manager subsystem because the node might experience a slight overheating normally, due to a high computing load, and no physical danger might be involved. To resolve the issue jointly, both Clustrx Watch and the task manager use and update a knowledge base.

If a decision is made to shut down a certain device, Clustrx Watch makes sure it happens. There are multiple ways of power supply management for some devices, so the system uses smartly whichever is available to ensure the execution of the shutdown decision.

In addition to automated response tools, the administrator has an extensive manual control over the monitoring and its data: scripting, the access to information at the level of any transit node, statistics etc.

Managing Workload

Managing the computing tasks and workload is a function not assigned to a single particular subsystem but integrated seamlessly into the whole system with Clustrx Watch as its backbone. (Clustrx Watch supervises the workload management to a great extent, especially when some critical decision making is needed.) Therefore the current focus is the workflow as a whole, supported by OS and administration activities.



The queue of computing jobs, the user access rights, and assigned limits of computing resources are controlled via a single administration interface. The user, too, has the right to formulate specific requirements to the compute node boot image that he needs to perform his task (such as the presence of certain libraries or a particular third-party operating system). Open APIs and the access to dConf are available for this purpose.

A library of boot images is stored in the single configuration database, dConf, where they are taken from by appropriate system services and booted onto compute nodes. Clustrx is capable of booting nodes both from disk and remotely via network. As the booting of a great deal of compute nodes at a time creates a network overload, the boot data can be cached on transit management nodes (the same used for aggregation of monitoring data).

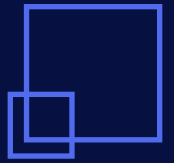
The user-defined requirements to compute nodes go to its job profile; it may include also specific requirements to hardware such as the presence of accelerators. Clustrx implements such hardware handling. The Resource Manager, which is common for the whole cluster, puts the job on nodes that have such hardware (if available). Generally, when a job is put on the queue, its profile is handled by the Resource Manager and appropriate resources are allocated.

When the setup of the computing environment on the nodes is done, they are ready to accept jobs from the queue. The execution is controlled by the resource limits assigned for a particular computing job and by data of monitoring collected in near-real time. If Clustrx Watch and the task manager together find it necessary, any node can be rebooted, shut down, or hibernated, for example, to save power.

If an emergency is signaled (such as overheating), an automated analysis is performed additionally. For example, if the overheating is caused by a high user workload on a compute node, it will stay powered, but an attempt at lowering the temperature will be made, say, by decreasing the workload intensity. If the reason for overheating is a cooling crash, the node will be shut down immediately. The next priority to guarding the health of the hardware is keeping the user job running. As more rules are accumulated, the accuracy of the response improves.

The tight integration between the task manager and Clustrx Watch allows advanced response approaches to be used, such as a correct termination of a computing job on a hazardous node.

Cumulative data from the task manager and Clustrx Watch enable a very precise accounting of resources that a user job consumes. This feature makes it possible for independent software vendors to lend their expensive applications instead of selling. This approach can be attractive both for consumers and for service providers in many situations.



Going Virtual To Keep Clustrx Up

Services of Clustrx are fully virtualized and partitioned from the hardware environment to achieve a highest possible stability, security, and scalability. Clustrx gives out all computing power from a single point of control, without having to care about the low-level distribution of the workload between hardware resources. As the system contains a great deal of components, its administrator should take a more convenient look at it as a single tool that provides all the control functions needed to manage the cluster and keep it up to execute computing jobs.

The virtualization of the system's services contributes to a system of no point of failure. Data communication between the services creates a level of controllability never achievable with earlier-generation systems where the HPC stack stayed clearly apart from the operating system.

The Clustrx infrastructure is highly optimized, its services are distributed between management nodes depending on hardware and other aspects in an automated manner, using a cumulative knowledge base. Business logic is moved around between Watch nodes as needed to reach a top reliability and scalability. As more nodes are added to the cluster, it can be resized live and transparently.

Another view of the system is that of an administrator who can use his functions as fully virtualized services coming from a "blackbox" that has an underlying architecture. Queries and commands related to user access, handling computing jobs, monitoring etc. come to the management blackbox that generates a response or action. The whole cluster can be controlled from one administrative console:

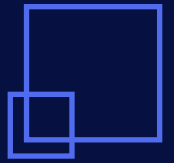
User access rights. The users can be allowed to add their jobs to the queue, view its execution status, access their data. However, it is the administrator who controls the access to these activities based on rights and rules. Clustrx system and network configuring. This information is stored in dConf and therefore can be controlled by queries and scripts. Scripting gives the administrator the level of control over the system configuration that allows, for example, to make modifications to thousands of nodes at a time.

Health control. This includes power supply, cooling, network infrastructure for all objects of the cluster. Both the manual and automated control is fully centralized, and the infrastructure of the cluster is optimized automatically to save power, which is both an economic and safety issue.

Compute node control. This includes handling boot images and remote boot issues. They are stored in dConf and are accessible both via its open APIs and manually.

Firmware update. Low-level issues like BIOS, networking, slots can be controlled on management nodes.

All administration activities are supported by Open APIs. Any subsystem can be accessed automatically by scripts or plug-ins so commands can be executed on



multiple nodes at a time. This scales the administration to any cluster size. An important administration interface tool is a cluster map that visualizes the state of all cluster objects interactively with respect to workload, health etc.

The multilayered architecture conceals physical locations where all the above tasks are handled. To achieve a proper consistence in this approach, the architecture contains no legacy code and has been redesigned from scratch.

Opening the blackbox shows the topmost architectural layer, that of system services which look monolithic. However, their architecture is more sophisticated and involves layering. Going one level deeper shows that the Clustrx services run in virtual containers or “slots”, that can be moved around between hardware nodes. Each service is a distributed system that integrates components launched in the slots and can be seen from above as a single virtual system object.

In their turn, the slots run in the Clustrx CBIOS environment. This is a Linux-like operating system that runs on the management node hardware (the dark purple layer), abstracts the containers from particular hardware resources, and thereby ensures safety.

Summarizing

Virtualization of the architecture, highly automated management, powerful monitoring and health support, infinite scalability, integration with existing solutions... this is what contributes to making Clustrx the choice for high-performance computing of the new decade.

Contact Us

The company runs a big research and development office in Kyiv, the capital city of Ukraine, and a representative office in London, UK.

Development

2a, Sribnokilska str.
Offices 90-91
02095, Kyiv,
UKRAINE
+380 44 5748665
+380 98 4360771

Representative

1, Lymington Court,
Lavender Hill
Enfield EN1 8NE,
United Kingdom
+44 845 508 61 37

info@massivesolutions.co.uk

<http://www.massivesolutions.co.uk>



Massive Solutions

