Enabling Moab's Adaptive Computing for Cray XT3/XT4

Michael A. Jackson, *Cluster Resources, Inc*.

ABSTRACT: *Cluster Resources, Inc.™ has ported their Moab Workload Manager® to work with Cray XT3 systems (on TORQUE Resource Manager, PBS Pro, runtime, and other resource manager environments). Moab™ offers an extensive suite to facilities, allowing sites to intelligently and automatically adapt resources, workload and policies to a changing environment. Moab can apply the adaptive computing in areas ranging from QoS/SLA enforcement to automated failure recovery to machine-learning based optimization. We will discuss how Moab can help you apply next generation technologies today.*

KEYWORDS: Moab, XT3, XT4, batch, scheduling, adaptive, utility computing, workload management, Cluster Resources

1. Introduction

This paper provides a technology review and scenario-based case study of the use of Moab Workload Manager® in an adaptive computing infrastructure for use with Cray XT3/XT4 or other current or future Cray architectures. Moab is currently utilized at Oak Ridge National Laboratory (ORNL) on Jaguar (#10 on Top500), a Cray XT3 which has more than 18,000 cores & on ORNL's 1,024 processor Cray X1E, as well as Sandia National Laboratory's Red Storm System (nearly 13,000 CPUs) and at another leading Cray site with nearly 19,000 cores.

Coverage will be given to Moab's high-level architecture and how this design is able to quickly accommodate the growing complexities of Cray systems and to increasingly make such systems highly optimized and able to adapt to changing needs and conditions intelligently. The result is a more efficient and more flexible system, a system that has improved uptime and decreased administrative overhead.

Cluster Resources, Inc.

Cluster Resources, Inc. is a leading provider of workload and resource management software and services for cluster, grid and utility-based computing environments. Cluster Resources is recognized as a leader in innovation and as an organization that delivers a high return on investment through its popular Moab™ product lines. With more than 5,000 sites worldwide using Cluster Resources' developed and/or maintained software and more than a decade of industry experience, Cluster Resources delivers the services and software products that enable organizations to understand, control and fully optimize their compute resources.

2. Needs for Adaptive Computing

The XT3/XT4 and other Cray platforms have a number of compelling advantages that make them a platform of choice, such as scalability, high efficiency, performance and effective performance analysis. However, the complexities associated with such platforms can require customer sites to develop or acquire deep expertise and apply a reasonably high investment in administrative management of such powerful systems.

Adapting such environments from one usage scenario to another, adapting to failure conditions or simply optimizing usage in the face of technology complexity can be overwhelming, causing sites to either not adapt, to not fully respond to failures or to not optimize the available resources and technology capabilities that were acquired.

3. Management Evolutions

High Performance Computing (HPC), inclusive of both clusters and supercomputers, is going through a number of management evolutions. Some sites leap past one or two levels in this evolution based on the nature of the organization and the growing management complexity that they are experiencing. The management evolutions are proceeding in a clear direction towards adaptive computing, though the reasons of advancing towards the mature adaptive computing are different based on the nature of the organization and their resource and process complexity. The foundation drivers are fairly simple: An attempt to get more work accomplished, improve ease-of-use, increase organizational agility, improve reliability, reduce human management burdens, enable new services, and other base level drivers. The management evolution phases are listed below:

- **A. Simple Supercomputer or Cluster Management** (Management centered only on the compute resources, all other resources are managed in stand alone silos.)
- **B. Holistic/Broadened Supercomputer or Cluster Management** (Management extended from compute resources to cover additional resources in an orchestrated manner. Multi-resource management may add orchestrated management of storage, software licenses, network resources, satellites, tape robots, chillers, and any other resource or service, or all of them in a holistic approach.)
- **C. Local Area Grid** (Management of multiple clusters and/or super-computers inside of a unified administrative, data and/or security domain. For example unifying multiple HPC systems that share storage and are used within a single organization.)
- **D. Wide Area Information Grid** (Management of multiple clusters and/or supercomputers that have multiple administrative, data or security domains for the purpose of unifying usage information, capacity planning, cost sharing or in preparation of resource consolidation. For example a multi-departmental organization that shares some resources, but does not yet migrate workload between systems.)
- **E. Wide Area Management Grid** (Management of multiple clusters and/or supercomputers that have multiple administrative, data or security domains for the purpose of unifying management of resources, policies, and other such capabilities. For example a multi-departmental organization that shares some resources, desires to reduce the administrative burden of independently managing multiple siloed systems, but does not yet move workload between systems.)
- **F. Wide Area Workload and/or Data Grid** (Management of multiple clusters and/or supercomputers that have multiple administrative, data or security domains for the purpose of unifying management and resource access. For example a multi-departmental organization or even multiple independent public organizations share some or all resources for the purpose of service level control over all resources, reducing end user barriers to access resources, optimizing resource utilization or to increase collaboration.)
- **G. Adaptive / Utility Computing** (Management of a single or multiple clusters and/or supercomputers and/or data center resources where management mechanisms orchestrate multiple or all resource types and interact with process logic to adapt to changing conditions or needs. For example applying management to adapt to resource failures or bottlenecks, dynamically changing policies to meet changing needs, adapting software environments or configurations to optimally meet workload demands, etc.)

Initially clusters and supercomputers were created to solve singular problems and only after they were successful did the effort to make them more generally applicable get applied to any great degree. Also management of such systems began with a fairly simplistic start, focusing more on enabling the computational capability and less on making such systems highly manageable. Further, management focused primarily on the compute hardware and left the other elements of the environment to be managed in silos and not as part of a unified environment. Thus resource managers such as TORQUE, LSF, PBS Pro, SLURM, etc. focused their value on job submission and compute node monitoring, and similarly cluster deployment kits and cluster management tools initially focused on compute hardware.

Other resource types such as network, storage, software licenses, etc. were set up, and while their availability and proper function was an important dependency to availability and optimization of the systems, they were left in their initial states and not coordinated or orchestrated in conjunction with the compute resources and workload conditions. Some resource manager vendors added individual add-on products to address other resources, but by their nature kept their management to more of a silo-based approach.

Schedulers such as Maui Scheduler solved some of the issues as they were abstracted from a policy perspective to cover any resource type but still had integration points with only one resource manager and thus inherited their limit in resource and information scope. What was needed was an abstracted policy engine that could simultaneously interact with multiple resource managers and information sources. Cluster Resources' Moab product line was created with this in mind—to allow the next evolution in management.

After achieving the ability to orchestrate multiple resource types and to unify policies across them, organizations are now able to turn to unification of management across multiple systems. The easiest resources to unify are those that share administrative domains—those that are managed by the same organization and under the same administrative infrastructure. Further technologies such as shared storage, unified security mechanisms (i.e., unified file systems, LDAP, etc.) facilitate quick and easy unification of resources. In this case the management toolset is able to simply interact simultaneously with resource managers on each system and apply a single set of policies that span the set of resources. This Local Area Grid (LAG) unifies management so that in essence the systems look and are managed as one from an administrative, data, security and policy basis.

Many organizations have compute resources that span departments or were initiated in a disparate manner, leaving silos of resources, management, data and security infrastructures. To solve management issues in this area organizations do not need to try to overcome resource, technology and political issues all at once but can approach it in smaller phases.

In particular some organizations may elect to move to an information-only grid for the purposes of gaining capacity planning information or to set up cost allocation or sharing mechanisms. This lets them gather usage information and unify reporting across separate systems. In this case there is no requirement for data staging, unified security or grid level policies, rather just a policy engine that can integrate with multiple information sources and aggregate this information. Also useful in capacity planning and other decision making areas of an information grid are simulation capabilities to apply historical patterns of resource usage against profiles of resources and then the ability to change simulation settings to evaluate resource changes or policy impacts to such environments.

The next level of Wide Area Grid (WAG) management moves from information and decision making to policy management. In a WAG, resources can be managed in a unified, peer or mixed manner, as opposed to the LAG that only allows the unified manner. You may unify all resources from a single tool, or you can manage some centrally and delegate other elements of management down to the local administrator. You can also set up a model that has no central management, but simply allows peer organizations to reduce complexity by standardizing on a shared management tool. This phase is often used by organizations that maintain separate owners that do not intend to collaborate, but the central organization seeks to reduce complexity and increase manageability of such resources. This phase is also used by government organizations that have security requirements that disallow workloads to move across resource boundaries, but once again seek to improve management capabilities and reduce costs.

The final level of Wide Area Grid evolution is the full workload and data grid phase. In this phase political needs and environmental constraints are able to be applied in policy logic to create the desired experience. Resource owners can continue to maintain sovereignty over resources while increasing collaboration and optimization. By allowing data and workload to migrate between systems the intelligent policy engine can optimize around workload surges, to further guarantee service level commitments, improve end user ease-of-use and increase overall productivity. Moab Workload Manager for Grids is able to deliver any one of the three WAG phases, but the purpose of this paper is to focus on adaptive computing and so it will not be addressed in any detail.

The adaptive computing management evolution can either be a natural step after any one of the WAG phases or may occur in conjunction with, the individual supercomputer or cluster that is moving to broadened or holistic resource management. It does not have a prerequisite of any of the previous phases, but is rather an increase in intelligence and interactive capability that matches to any of the previous phases, as well as commonly to data center environments.

The remainder of this document will focus on further describing adaptive computing, the underlying adaptive computing component capabilities, adaptive computing purposes and application areas in HPC and specific example scenarios.

4. Market Adoption

Market Adoption of Adaptive Computing is difficult to measure due to the multiple uses and applicability of the term to different markets. However, International Data Corporation (IDC), has provided a market figure of approximately $5 billion for the Utility Computing space (it is understood that this area has a significant association in definition to many elements of adaptive computing). In comparison with the $45 billion worldwide traditional server market and $11 billion HPC market, the $5 billion dollar level for utility computing is indicative of a less mature, but nonetheless substantial part of the IT market. What is not clear is the division between Adaptive Computing used in HPC as opposed to data centers. Also worth note is that the Utility Computing space is growing more than 10 times faster than the general server market and approximately three times faster than the general HPC market.

5. Adaptive HPC

Cluster Resources' Moab product line is able to be applied to multiple computing models. Adaptive computing can be applied to data centers to optimize business processes, improve reliability and utilization and other such values, as well as be applied to set up hosting centers. Moab-based hosting centers can be used both in internal form and for external use. Internal hosting centers are often used to unify resources and then set up billing relationships to internal departments so that IT organizations can act as internal service providers and manage costs around cost sharing models. Some organizations now use the term "full economic costing" in conjunction with this cost sharing approach, as they incorporate personnel costs, power costs, facility costs and other such costs in their cost sharing and hosting model with other internal organizations. In a similar model this is used to host access to resources and charge customers for use of the resources, application use, access to the services on top of the hosted resources, etc.

Other organizations may seek to apply Moab's abilities to become a resource importer. Organizations may seek to work with a hosted resource provider to guarantee access to additional resources or service levels above what they can achieve with their own resources. Such organizations may desire to have an external organizations manage their resources, or to get access to unique resources, or to set up surge protection or disaster recover capabilities. Moab can orchestrate the service level management, policies, reporting and other required capabilities in the workload, policy and event management space for these usage scenarios.

Finally, adaptive computing is applied to pure HPC sites that leverage the added intelligence capability and orchestration to improve SLA enforcement, improve optimization, make such HPC sites more resilient to failure conditions, reduce administrative burdens through added automation, etc. This last area, adaptive HPC, will be the focus of the usage scenarios described later on in this document.

6. Architecture and Core Adaptive Computing Concepts

Adaptive computing capabilities can be significantly impacted due to the architecture of the underlying intelligence and orchestration engine. Rather than trying to describe the pros and cons in different approaches this paper will describe Moab's applied approach and the values of this approach.

Multi-resource Manager Support: Moab has a critical capability of being able to support multiple simultaneous resource managers. Not only does this refer to traditional compute resource managers such as TORQUE Resource Manager, LSF, PBS Pro and SLURM, etc., but it extends to hardware managers, license managers such as FLEX lm, network managers, storage managers, device mangers, tape robot managers, or any other manager of consumable or non-consumable resources. It is important to be able to integrate with multiple resource managers both for purposes of making holistic decisions, as well as the ability to unify resources under a single management structure. If the intelligence and orchestration engine did not have this capability it would be blind to resource failures or bottlenecks that impact workload, it would not be able to adapt to or change critical conditions and would make less efficient decisions.

Also valuable within the space of multi-resource manager support are the abilities to unify information from the different systems and the ability to translate resource manager languages to increase interoperability and allow sites to unify resources without ripping and replacing previous resource manager choices.

Abstracted Scheduling Management: Because Moab's scheduling concepts are abstracted, it has the ability to schedule virtually any resource type including compute resources, network resources and storage resources. Further, Moab has current and future time reservation management, floating reservation capabilities, time offset capabilities and other flexible concepts. Moab is able to schedule resource usage, as well as processes or events that are associated with an entire workflow or business process.

Event Engine: A very core concept that is a requirement for an effective adaptive computing infrastructure is that it has a flexible event engine. In the end the flexibility must exist for the event engine to virtually mimic human evaluation as best as possible and then to automate adaptive processes that have clean answers and to mitigate and escalate issues that require added evaluation. The event engine also must have the ability to impact both policies and settings inside the decision structure of the adaptive intelligence engine, as well as the ability to interact and change configurations, settings, send commands and otherwise adapt the environment over which it is applied.

Workflow and Complex Logic Structures: To enable adaptive computing for complex business processes there must be a workflow design that allows decision trees, feedback loops, cascading triggers and other such concepts. Moab has this advanced structure which allows near human level logic to be applied to adaptive scenarios.

Generic / Abstracted Resource Monitoring: Associated with an effective event engine is the ability to expand what can be monitored and included in what is being adapted or considered when evaluating actions. Moab is able to leverage its native resource manager interface to interface with virtually any information source. This allows Moab to interface and monitor network traffic within a supercomputer or cluster or between them, web services, command output, flat files, key word searches in event logs, xml strings, data base information, custom scripts, temperature readings, and many other information sources and then to use the event engine to drive behaviors that adapt to the added information.

Generic / Abstracted Event Definitions: Once Moab is able to monitor the environment and activity within the environment you can leverage its ability to identify a specific activity, threshold, output, etc. and define it as a generic event. Then you are able to trigger actions to take place that leverage or correct the observed event through use of the event engine.

Generic / Abstracted Metrics: It is also common to require more than a single event or series of events to take place before Moab can take appropriate action. This can apply when evaluating when to adapt to an environmental condition or to service level requirements or in a complex workflow situation. Moab has the ability to generically apply counters and track historical patterns of these conditions. These metrics allow decisions to be made and reports to be generated based on site specific environmental factors. This increases Moab's awareness of what is occurring within a given supercomputer or cluster environment, and allows arbitrary information to be associated with resources and the workload within the system. Uses of these metrics are widespread and can cover anything from tracking node temperature, to memory faults, to application effectiveness.

Dynamic Workload: There are two principle approaches to adaptive computing and it is ideal to be able to apply both, as a hybrid of the two will be most optimal. The first is to be able to adapt the environment to meet the needs of the application workload that the organization is trying to accomplish. The second is to be able to dynamically adapt the workload mix and the workload itself to better match to what is available in a constrained resource environment. Moab has both capabilities and is able to apply them simultaneously as appropriate.

Moab has a dynamic job concept that allows it to receive in a range of acceptable resource constraints and then is able to adjust and apply the constraining parameters based on its knowledge of what will result in the fastest turn around time for the workload. For example, if a given workload can be accomplished in 80 core hours and it does not matter if it runs on 2 cores for 40 hours, 20 cores for 4 hours or anything in between, Moab can evaluate and see that it has 2 cores now, and would have 20 cores in 20 hours or could apply 15 cores right now. In this case it may decide to apply the 15 cores now and get the work accomplished in 5.33 hours from now rather than leaving the end user to guess or lock into a non-ideal request.

Another dynamic workload example is where sites with multiple different Cray architectures can 'teach' Moab what constraints their applications have and then leave it to Moab to optimize placement. In this case Moab will consider which Cray platform it will apply the workload to, based on performance differences per platform, and on current and future platform resource availability.

Holistic Workload Concepts: Many resource managers in the HPC space can only work with serial and parallel workload. However, many full processes require elements to be accomplished on one or more sources such as a supercomputer, cluster, traditional datacenter resource and perhaps other elements to be made available via web services. Moab is able to manage transactional workload, serial workload, parallel workload, web services, and other forms and is able to interact with each of the systems to build a holistic adaptive process.

Also very key to this space is the ability to manage system requests that are not the final workload being accomplished but rather part of the internal interactions and directions being given that are necessary to accomplish the final workload. Moab is able to manage and schedule elements of the system level requests to help optimize the system and work within internal constraints and timing requirements.

Virtual Private Resources / Environments: Another valuable concept is the ability to aggregate the collection of resources from multiple sources into a virtualized resource package that is dedicated to an organization, application, project, etc. This virtual pool of resources, configurations and definitions may consist of an entire compute environment including the needed compute resources, storage, bandwidth allocation, provisioned software, configuration settings, VPN set up, event infrastructure, etc. This is key to be able to create adaptive environments that get changed based on application requirements, or based on which entity is using the system or on different phases of a large process.

This capability lets Moab cater to scenarios such as high security environments, optimization of environments for processes or applications, or different internal usage cases. These can typically be done simultaneously on different available resources so that it is the equivalent of creating a customized environment that is optimized for its purpose and being able to recall the packaged environment and apply it at any time in the future.

Aggregated Reporting, Billing: Moab has the ability to track and report not only its own data, but also to aggregate reporting information that it gathers from underlying resource managers and information services. Sites are able to teach Moab to trust one source over another when information is redundant and much of this information can be pulled in and aggregated into combined reports. Some organizations also seek to share costs or gain revenues by billing organizations or external customers for the use of their resources. When combining reporting and virtual private resource definitions it becomes natural to provide on-demand resources to end using entities, tracking usage and charging them for actual usage or usage plus margin rates.

7. Areas of Application for Adaptive HPC Computing

Adaptive computing can be applied to many different areas of HPC computing, though most organizations will only apply those capabilities which best match their organizational needs. This section will describe a few of the purposes of adaptive computing and a listing of example areas of its application but will not cover them in detail. The following sections will focus on a smaller number of specific scenarios with greater detail.

Adaptive computing can be overwhelming when considering the scope of the value and points of complexity that can be managed. Fortunately, the capabilities can be applied as simplistically or as holistically as desired and there is no dependency on applying the technology to an entire environment. Moab can be applied to solve tactical issues and then augmented upon as desired. Some adaptive scenarios are default behaviors, others can be set up in less than a day and even some complex scenarios can be handled in less than a week. It is common to tactically manage high value adaptive needs and then to incrementally apply improvements in phases thereafter.

Optimizing Resources / Investment / Work Accomplished: Adaptive Computing is often applied for the purpose of optimization of resource usage and optimizing workload that can be accomplished on available resources. A key example for Cray systems will be the ability to use Moab in a mixed hardware environment and to use affinity or requirement-based workload placement. This allows the mix of vector, scalar, MPP and MTA workload to be submitted to a single system and for Moab to optimize placement of the workload against the mixed resources in an intelligent way. This also allows different rules and conditions to apply based on the choice which is made. Other optimization areas can include mixed network conditions or types with workload placement, network re-configuration or condition resolution.

Moab can be used to replace manual activities. In many cases administrators identify conditions that require them to take a set of actions. The conditions may be intermittent, but the same actions may apply each time. Moab can be used to monitor for the intermittent conditions and then create an automated response that in effect patches the situation with event instructions.

This is often desirable to resolve issues that may have alternatively required vendors to provide actual patches or hardware to be updated, etc. This allows a virtual patch to be applied and allows the organization to return to non-interrupted service or to experience fewer service interruptions as they wait for the vendor to make the required changes. Indeed in some cases it may resolve incompatibility scenarios that are never fixed because no single vendor owns the issue. Another area under the purpose of optimization is raw application optimization through learning-based historical self-tuning.

Quickly Adapting to Changing Needs: Organizations experience significant changes in what they need to deliver and it is very valuable to have an infrastructure which can quickly adapt to those changing needs to help meet organizational objectives. Product companies can experience significant shifts in product delivery timeline pressures due to outside events or competitors. Mergers or acquisitions of other organizations can cause high levels of disruption that must be adapted to quickly and can cause revenue loses or budget over-runs that are only relieved upon completion of such changes.

In the areas of defense, threats can change rapidly and the resources that are delivered to assess and counter the threat can make the difference between strategic information leadership and being blindsided. Specifically, adaptive service level delivery and condition-based policy adaptation are important application areas to quickly meet changing organizational or application needs.

Further, security adaptation can allow security-differentiated workload to share elements of different resources, as long as the required policies and procedures specific to the security level can be automatically applied when appropriate.

When capacity planning is used, and new resources are being considered or new workloads are to be introduced or new rules of usage are about to be applied, Moab's simulation capability can be valuable to evaluate the impact of such changes virtually without needing to experience the results of poor or incomplete choices. Simulation capabilities aid organizations to look at what would happen in different scenarios and to make decisions faster and more confidently due to the precision and speed of the evaluation work.

Automatically Responding to Failure Conditions, Surges or Other Environment Changes: Nearly every Cray system deployment has a number of unique environmental elements that exist only at that given customer site. Additionally, due to the sheer scale of Cray systems the numbers of failure points increase. This complexity of unique environment factors that commonly push the edge of associated technologies and the failure points that exist due to scale and system complexity create an environment that can gain great value from adaptive computing. Moab's adaptive infrastructure allows it to adapt workload placement, policies and environmental elements in situations such as Lustre file system state issues, CPA feedback failures, network hotspots or failures, workload surges, temperature irregularities, or a host of other failure conditions, surges or environmental bottlenecks.

Moab's adaptive capabilities allowed it to solve multiple customer issues in a few hours where vendor solutions did not arrive for months thereafter or in some cases still have not arrived.

Surges in specific workloads could trigger Moab to adapt an environment to the needs of a specific application, even to the point of re-compiling the application to work on other resource types, or allocating additional resources not originally specified for use to meet original service level requirements. Another application can include connecting up to machine room chillers or UPS systems or other critical infrastructure and setting in place automated safe shut down procedures to span multiple resource elements and workflows.

8. Usage Scenario: Mixed Vector, MPP, MTA & Scalar Environments or Systems

At the core of Cray's Adaptive Computing vision is the ability to join together mixed elements of best-of-breed computing hardware into a unified architecture and to have a software solution stack that can intelligently optimize in the face of a high level of complexity in a way that is transparent to end users. Moab has the foundational elements to specifically orchestrate workload, service delivery, environmental preparation and processes in an optimal way. While other solutions may eventually be able to provide simple routing services of workload to resources that match, Moab goes far beyond this ability in its adaptive computing approach. Moab has a mature adaptive computing architecture providing the core concept capabilities reviewed in section 6 above.

Specifically the ability to effectively apply adaptive computing within Cray's envisioned mixed hardware model requires multi-resource manager support to gather information from mixed sources. For example, gathering hardware, network, storage and file system information. Also, abstracted scheduling management will be important so that Moab can cater to the unique attributes of each architecture and the needs of each application. It is also important to be able to handle the coordination of the multiple simultaneous application and architectural requirements with their associated underpinning dependencies. The concepts of resource affinity and resource requirements are needed both on the application template and workload submission front.

The event engine is needed to drive the multi-step processes to adapt the environment to meet application needs, as well as to interact with external information systems and tools. Workflow and complex logic structures are valuable to enable applications to be optimized on different resources at different phases of their calculation work.

Generic / abstracted resource monitoring will allow Moab to track Cray specific attributes that may only apply to specific architectures and to provide the needed information for scheduling or event actions. Generic / abstracted event definitions become the tags that are used to build some of the workflow processes and Generic/abstracted metric tracking becomes the basis for application learning.

While not on the original concept list in section 6, Moab's application learning capability is valuable in its ability to benchmark and auto-tune workload placement across the different architectures.

Configuration Work: Initially, configuration would consist of applying node attributes to each of the compute nodes so that Moab can differentiate between the architectures. Next you would connect Moab's multi-resource manager mechanism to the appropriate information and management sources. Much of this may become the default functionality in future releases, similar to Moab's current ability on Cray systems to connect with TORQUE Resource Manager, PBS Pro and LSF, as well as to connect to CPA, the XT admin database and other information sources. For any new information element that is desired to be part of the policy decisions or to be an element of the adaptive environment, Moab can connect up to it via its native resource manager capability. It is important to understand that this can be a single additional source of information or it can be multiple simultaneous sources of information and management.

Some of Cray's architectures require the co-allocation of resources (e.g., one management node, one I/O node and x number of compute nodes). You can either leverage existing co-allocation logic or establish new co-allocation dependencies as well as order, timelines and associated processes.

Specialized events can be set up if required to improve information flow or to automate adaptation elements.

At this point you would want to set up "Application Templates," which identify resource ranges, default settings, dependencies and associated workflow activities. Application templates are not required but reduce end-user input requirements and improve the efficiency of resource utilization. They also prepare the way for application learning to be applied. Application learning aids by automatically tuning which resource will be allocated to an application workload to achieve optimal results.

As inferred above you would create "affinities" and "requirements" between the application templates and the node attributes. This is done so that applications that can run only on a specific resource are so indicated and applications that can run on multiple architecture types but have better performance on one type can be configured in favor of the faster platform. Then Moab can balance between requirements, affinities and affinity-to-availability comparisons on both a current and future timeframe basis.

An excellent capstone element of this is to configure automated application and node learning. You can configure Moab to learn which resources provide faster results for similar sized application-specific requests. Over time, Moab can be set up to observe completion time results for the application under similar workload sizes and auto-tune its affinity algorithm to favor placement to the best performing nodes. Moab is able to also balance this best performance affinity with prioritization and other factors.

Further, Moab is able to have nodes self-tune their own affinity by using their aggregate uptime and availability information, so that jobs are more attracted to nodes that are more reliable, leaving less reliable nodes to be steered around unless excess workload exists.

User Transparently Submits to One Location: The purpose of this usage scenario is to significantly simplify the experience for end users. End users are able to submit their workload to a single location without even having to know which platform will provide the greatest results; then Moab evaluates what the application constraints are; it also evaluates committed service levels to the individual, their organization, project, etc.; then further evaluates resource availability; it applies the affinities which will optimize placement; it potentially auto-provisions or adapts environmental aspects to better meet application needs and then it applies workload processes to execute the work.

In this scenario the end user experience is simplified as application templates fill in defaults, users are kept within permissible ranges and Moab applies all other logic requirements. Node learning makes sure their jobs avoid flakey nodes and application learning steers jobs towards the best performing resources.

Moab can also work with very complex adaptive scenarios where parts of applications may be triggered to run on one resource type, the next phase may be enacted on another resource type or even multiple resource types, and all of the integration, dependency management, co-allocation scheduling, etc. is orchestrated. This can potentially include future integration with Cray's adaptive compiling strategy which allows this per-platform optimization from a code perspective.

9. Usage Scenario: Failure Recovery & Ease of Use Automation

This element of adaptive computing is common to HPC and data center environments. Because there is such a large range of possible failure recovery situations and ease of use improvements, this scenario will be more of a limited detail overview that samples from various tactical situations.

Also, as this is not a specific scenario, but rather a scenario grouping, virtually any one of the core adaptive computing concepts could potentially apply and therefore will not be re-listed here.

Configuration Work: The first step in applying this capability is to specifically identify the failure condition. Next determine manual methods of identifying the issue via commands, scripts, tools, monitors, etc. Then apply Moab's native resource manager to import failure/state information found from the script, flat file, CLI, XML, web service, etc. This information can be associated with a generic event or measured by a generic metric to more easily create triggered actions. Then identify manual steps, commands to run, people to notify, workflows to apply, etc. that are required to resolve the issue. Finally apply Moab's event mechanisms to interact with systems, remote tools, commands, scripts, notification mechanisms, etc. via a workflow like logic tree to solve or mitigate the issue and apply desired notifications. Generic metrics are also valuable for tracking the frequency and location of such failures over time.

Example Results: The following are simplified failure recovery examples.

A. Lustre Multi-state Phases Confuse the Resource Manager: In Cray's case, Moab integrates with Cray tools (currently CPA & XT Admin Database) to evaluate file system state, but can also integrate with the file system itself if needs be. Without this integration and added information, the underlying resource manager was at times submitting workload to Lustre when Lustre was still in a state that could not accept workload, as it was cleaning up after previous workload.

When jobs were submitted at this time there was not a proper feedback mechanism to inform the resource manager that submitted workload could not run since the file system was not ready. These cases block nodes, as the resource manger thinks they are running and they are not, and more and more nodes get blocked due to incorrect information. Because Moab can track multiple sources of information it was able to validate the file system status before submitting jobs, thus eliminating the point of failure.

B. **Traffic within a Cray Network Creates Hot Spots that Cause Potential Bottlenecks and should be Avoided** While not yet implemented, a reasonably simple addition would be the connection of Moab to Cray or Cray user network monitoring tools that track where network hotspots are within the Cray supercomputers. Then there would be two obvious potential paths to pursue.

The first would be to evaluate if there were methods to direct the path that an individual job would take via the internal paths within Cray's specialized networks and then apply this within Moab's scheduling algorithms so that it avoided hotspots. In essence you could set a generic metric which measured the acceptable thresholds and then uses the event engine to drive commands which help steer the workload down the desired path.

If there is not a method to instruct Cray's network on which network paths to take then you may be able to simply optimize job placement on node locations that will avoid such hotspots and minimize latency in general.

At previous CUG events Cluster Resources has observed fairly advanced internal network hot spot monitoring tools. It is also very reasonable to integrate with these tools and use the supplied information to better avoid hotspots and maximize the balancing of network traffic in ways that favor faster job execution.

C. **Machine Room Chiller Failures** Moab is very generalized in how it integrates with resource managers. A machine room chiller can also be considered a resource can be coordinated with as long as such equipment has network accessible monitoring tools. Moab can monitor chiller status and room temperatures and take actions that avoid the equipment being damaged or the workload being corrupted. For example, after notifying the administrator of the chiller failure, Moab could checkpoint workload where possible, begin to pre-empt workload and then power down nodes before they overheat.

D. **External Power Failure, UPS Triggers** In this simple scenario, Moab would monitor UPS state, and when the UPS was triggered, Moab can be configured to take critical actions. After using the event engine to trigger a notification to the administrator, Moab can begin a series of mitigation steps. Other mitigation steps may be chosen, but the following is just one example.

Low priority workload could be check-pointed and then preempted. Those nodes which now no longer had workload running could be instructed to power down via trigger commands. If the UPS device allowed network access to current battery reserves data, Moab could then use a generic metric to monitor the threshold and once remaining power was below 10% the remaining high priority jobs could be check-pointed where possible and pre-empted and the associated nodes powered down.

These steps would help avoid corruption in data, extend the life of the UPS, maximize the life of high priority jobs, increase the likelihood that power is restored before additional high priority workload could get accomplished. It would also be possible to measure expected completion time due to wall clock estimates and maximize cycles towards those that may finish in the few minutes of remaining UPS duration.

E. **Compute Node Temperature Exceeds Desired Threshold** Multiple sites have expressed interest in monitoring temperature and applying Moab's logic to work around machines with excessively hot processors. The hotter a node or processor is, the more likely it will fail or experience other undesirable issues.

By connecting Moab up to temperature monitoring tools via Moab's native resource manager and setting generic metrics which create a baseline acceptable temperature, Moab can be configured to steer new workload to other nodes while the hot nodes have a chance to cool off.

Alternatively Moab can be configured to simply create an affinity so that it first tries to run on cool nodes and then runs on hot nodes only after there is no other choice. If workload is properly identified it may also be possible to route low CPU intensive workload to the hot nodes (i.e., jobs that are more network intensive or I/O intensive, etc.) that do not use the CPU 100% of the time. This way the node can continue to be used while intelligently applying workload that lessens the cause of risk.

In this scenario you might select two generic metrics for different temperatures and simply use the mitigation steps on the lower of the two and on the higher "very hot" temperature you might send a notification to the administrators and set reservations on those nodes that block additional workload until their temperature issue has been resolved and then automatically remove the reservation when temperatures normalize.

F. Storage Manager Reports Warning In the case of a storage manager reporting warnings, it of course would be relevant which warnings were being reported. Moab can be connected to a simple script that searches out specific warning messages and takes different actions based on the warning scenarios.

In this case let's assume that the warning was sufficient to believe that jobs dependent upon the specific volume of data would fail or otherwise be corrupted if they continue. In such a case Moab could be configured to suspend jobs that currently need the volume and to block queued jobs that would otherwise fail.

Notifications could be sent to administrators and this behavior would continue until the storage manager warning messages were resolved.

Moab has a re-arm mechanism on its triggers so that you can have an intermittent issue be evaluated every x period of time and then Moab re-arms itself to re-evaluate the situation after a desired time. Moab also has an event counter mechanism to help avoid thrashing situations; for example, you can cause Moab to cease checking after 10 times and to keep the jobs in a protected state until released by the administrator.

G. Compute Node Local Disk Fills Up Moab can be set up to monitor available disk on a compute node, and when the available disk is full Moab can execute a script to purge temporary files and other unnecessary files. It is also possible to have Moab evaluate job status on the node to ensure there is no current job running that may be dependent upon the current temporary files. This scenario leverages Moab's generic metric capability, event engine, state evaluation capabilities and multi-resource manager support.

H. Effective Node Throughput Drops Below Desired Threshold This scenario may be indicative of a few different situations. As such situations are identified, specific scripts which evaluate symptoms can be launched, and corrective measures issued. The key to this scenario is the workflow and complex logic structure which is supported by Moab and its event engine. Rather than repeating individual diagnostics for each potential cause, you can structure a decision tree that triggers additional tighter diagnostics in the particular area of interest.

Resolutions may consist of re-provisioning due to corrupt software, recycling the node, running diagnostic commands, reporting status information to a database, or many other such actions. This can become a strong aspect of self-healing for Cray systems, and over time can significantly reduce the knowledge required to maintain successful experiences within the complex environment.

I. **Major Network Failure** This scenario is dependent upon various organizational decisions and the nature of the workload. Moab has hosting and resource import capabilities that would allow a site that has a major internal network failure to leverage Moab technology to dynamically set up a peer relationship with an alternate organization or system (internal or hosting provider) and to begin to migrate and execute workload on the remote system without disruption to the end user's experience.

To overcome trust issues and service level issues Moab is equipped with a number of key capabilities. Moab's Virtual Private Cluster technology allows a holistic environment definition, inclusive of configuration, Virtual Private Network set up, authorization mechanisms, policies of which workload would be allowed to be applied on the remote system and other inhibitor-removing capabilities.

This same technology can be used to also guarantee services levels during surges**,** as opposed to just failure conditions. For example, you could have a scenario where high priority workload could never stay in a queue for longer than 30 minutes, as Moab's policy mechanism would move workload that passed that threshold over to remote systems.

When all workload can execute on the local system within the service level threshold, Moab's policy engine would enforce it to do so to remove the costs or burdens to the network of accessing external resources.

This capability can also be applied to partial failures so that if a given node or rack of nodes, etc., were to fail the end using organization might not experience the loss of functionality. While the nodes are inoperable and replacement nodes are being provided the user continues to utilize what they had originally purchased.

10. Usage Scenario: Application and Resource Optimization – Automated Learning

Application and resource optimization can be quite broad, but this section will focus on just one element - specifically automated learning. Moab has an automated learning mechanism that is able to observe application performance on different nodes, as well as historical node reliability and is able to use both aspects to optimize application specific workload placement.

Configuration Work: To implement this capability you would ideally apply known template defaults and requirements to applications. This is accomplished by identifying the application, selecting input parameters and setting defaults or ranges of possible responses. Optional parameters may include which platforms an application can run on and which platform will run the application faster, the number of resources (e.g. nodes, memory, disk space, I/O requirements), dependencies, associated co-allocation of resources, the range of acceptable durations and other such parameters.

Within Moab you can configure the application learning mechanism to track specific metrics to equate to a performance rating and configure Moab to allow automatically learned optimizations to be applied or require manual validation to be applied. If automatic optimization is allowed, Moab would fine-tune application template definitions and set optimal resource affinities so that jobs migrate to nodes that historically execute workload of similar sizes of the specific application faster.

Associated with this learning is also node reliability learning that allows Moab to monitor uptime/availability of nodes and is able to set an affinity for workload to favor being applied to nodes that are reliable and avoiding nodes that historically have been flakey.

Application Learning Mechanism: Moab's Application Learning Mechanism currently consists of the following steps and elements:

Users submit workload to Moab. Moab applies the equivalent of an evaluation filter which looks at the job submission and evaluates which application template or templates that the job can qualify for. Moab then applies the application template(s). This results in Moab setting specific parameters identified in the template(s) and then applies additional defaults as appropriate.

At this point, Moab engages the added tracking stats that have been assigned to this application, so that additional historical results can be applied to future workload. Each application may have a different factor being tracked that best measures its performance and resource requirement. Moab applies previously learned affinities and applicable policies and scheduling rules. The automated learning engine evaluates the job upon completion and fine tunes the node affinity associated with the application template, preparing the way for the heightened optimization.

User Transparently Submits Workload with Minimal Detail and Optimal Performance: The user experience with application templates is simplified as the administrator guidelines and the historical affinities are applied, leaving the user to act within ranges of optimal parameters and only filling in details that can not be set as defaults specific to the application. To gain the learning capability the user does need to identify a single additional sizing or load measurement that allows the automated learning mechanism to compare workload on an equal footing.

Conclusion

Adaptive computing for HPC is not a mysterious unachievable goal but rather a natural management evolution to a more intelligent and capable level. This capability area is already available and can be applied today via Cluster Resources' Moab product in conjunction with Cray's current and future architectures and advanced middleware. Moab provides an intelligent adaptive computing foundation which is able to adapt the environment to meet application and workload requirements in an automatic and optimized user transparent way. It is also able to reduce failure conditions, increase manageability and broaden usability. There is no need to apply the full capabilities of adaptive computing in the short term, but rather apply the foundation and implement valuable tactical elements one at a time and evolve your management experience forward. This paper discusses a number of valuable applications. The purpose, however, is to open the readers understanding of the principles and purposes that can be achieved and then to initiate in them a desire to evaluate their own usage scenarios and how these concepts can greatly enhance their experience.

Acknowledgments

The author wishes to acknowledge the contributions from David Jackson, Cluster Resources' CTO for general guidelines and review, Jess Arrington for resource manager research, Trev Harmon for network, I/O and processor centric consultation and Jonathan Ryskamp for document editing.

About the Authors

Michael Jackson is President of Cluster Resources, Inc. and manages strategic customer and partner relations with leading sites and partners such as ORNL, Tri-Labs and Cray. He is a co-founder of Cluster Resources, Inc., which formed in 2001. Jackson can be reached at michael@clusterresources.com or +1 (801) 717-3722.

Appendix A:

Moab Evaluation Modes

Because Moab is often utilized in environments with mission critical production requirements and aggressive project roll-out needs, it has been designed with a large array of advanced evaluation features. These features allow Moab to be installed and executed on production resources transparently and safely evaluated in a risk-free manner. These modes include the following:

Monitor Mode – In *Monitor Mode,* Moab imports all information, processes all policies, makes all decisions, and reports back on actions it would have taken, any issues it uncovers, discrepancies in environmental information, and other factors. However, in this mode it does not directly impact workload in any way.

Simulation Mode – In *Simulation Mode,* Moab uses real-world workload traces it has collected from the supercomputer environment to create a full simulation environment where Moab schedules and tests policies while taking into account actual resource and workload configurations, timings, failures, policies, admin interactions, and other factors to predict scheduling behaviors and system performance based on historic information. In this mode Moab does not directly impact workload in any way.

Interactive Mode – In *Interactive Mode,* Moab fully schedules, but asks administrators for explicit permission before taking each action.

Partition Mode – In *Partition Mode,* Moab is able to only view a subset of resources, users, queues, and jobs. All other information is completely masked away. Inside of this sandbox partition, Moab fully schedules as if in full production. This lets sites test on the actual environment while controlling the scope of what workload and resources are under its direction.

Normal Mode – In *Normal Mode,* Moab acts in a full production manner.

Overview of Moab's Standard Characteristics

Moab abstracts resource management from policy management layers allowing it to intelligently optimize and manage almost any type of resource and any type of workload. It provides advanced optimization and quality of service features together with highly flexible resource ownership and resource management policies. It is designed to interact with a large array of services orchestrating their activity according to a high-level set of mission objectives. It is designed to detect and react to a wide variety of failure events, performance events, workload events, and administrator driven events and intelligently adjust its behavior accordingly. With its flexibility, Moab can dynamically adjust policies, dynamically adjust workload, or even dynamically adjust resources in accordance with its objectives, allowing a highly optimized compute solution.

Moab is a policy engine with support for simultaneous interfaces to multiple information sources and can tie new middleware with legacy applications and tie together heterogeneous environments of varying hardware architectures, operating systems, and resource management tools.

For example, Moab can utilize specific resource management APIs such as those found in LoadLeveler, SLURM, LSF, PBS Pro, TORQUE, and others, and can also simultaneously interface with node monitors, management tools, databases, information services, etc. This is valuable in establishing the foundation for creating a unified view or resource information and management.

As a further example, Moab can be used to create a grid with Cray systems mixed with non-Cray systems. In fact this heterogeneous support allows Moab to be selected for some of the most complex environments, as it supports virtually every hardware architecture (including Cray XT3/XT4, Cray X1E, Cray XD1, Intel Xeon, Intel Pentium, Intel Itanium, AMD Athlon, AMD Opteron, SUN Sparc and UltraSparc, HP Alpha, IBM PowerPC, SGI MIPS, and others), a broad array of operating system environments (including Unicos, all Linux flavours, Mac OS X, AIX, IRIX, SGI ProPac, HP-UX, TRU64, Solaris, BProc, Scyld, BSD and Windows), all popular interconnects (including XT3 Interconnect, Crossbar, Fast Ethernet, Gigabit Ethernet, Infiniband, Quadrics, IBM HPS, Myrinet, etc.) and all major message passing tools (including Cray MPI, MPI, OpenMPI, MPICH, MPICH2, MPICH-GX, MPICH-GM, LAM, Scali MPI, MPI Pro, PVM, VMI, Open MP, SHMEM, MLP and others).

Moab's architecture allows easy integration with external services though the use of generic C, Java, and JSP API's as well as script, database, web service and file based interfaces using XML and flat text semantics. These integration methods can be used to coordinate interaction with system services (e.g. identity managers, databases, hardware monitors, resource managers, network managers, switches, security systems, allocation managers and many others.)

As an integration platform Cluster Resources' products act much like a hub for workload management and service level centric processes. The more that the work being accomplished is connected up to additional software and informational services, the more intelligently and streamlined the processes can be. For example, if Moab is integrated with a hardware monitor it can make intelligent decisions as to how to apply the workload around problems. If integrated with a license manager, Moab can optimize workload submission around availability of licenses and report back the cost of time wasted due to waiting for licenses or the over-purchase of licenses due to a lack of requests.

Moab also allows flexible integration of command and control services from multiple sources. For example, job information within a supercomputer may be imported from one source, while resource information may be merged from two different sources. Job execution may be assigned to one interface, while network scheduling and resource provisioning may be assigned to other services.

Both Web-based and desktop graphical tools are included with the base package and are specifically designed for end-users as well as for administrators and managers.