A Model for Cluster Computing at Duke

Robert G. Brown
Duke University Physics Department
Durham, NC 27708-0305
rgb@phy.duke.edu

May 12, 2003

Last CVS checkin: Date: 2003/04/03 18:57:59 .

Contents

1 Introduction

Commodity cluster computing at Duke is finally coming of age. The early pioneers have well-established clusters, which have proven very beneficial and cost effective ways to support a variety of research initiatives. However, the old clusters have grown, and new clusters have proliferated, to the point where a number of infrastructure problems have begun to emerge.

Recognizing that clusters are likely to be the primary vehicle for high performance computing in the University environment for at least three to five years, the University is exploring models for providing the requisite infrastructure support. Its goal is to provide cost-effective support and growth pathways for both new and existing cluster-based research projects, in keeping with its primary mission of fostering education and research.

The existing organization of cluster on campus is purely driven by practical issues. Clusters are typically physically located in close physical proximity to (within the departments of) the groups that built and operate them. They are most often managed by departmental systems managers, sometimes augmented by cluster-experienced members of the research groups. There are notable exceptions to this rule of physical proximity, usually associated with a group that lacks adequate local facilities for their cluster. In these cases the cluster is most often located in a facility “belonging to” other groups also doing cluster computing.

For example, multiple clusters in physics, in the computer science department, in the engineering school, are physically located within the premises and are run by systems managers working for these various departments and schools. However, a cluster belonging to a group in the math department shares space over in computer science; a cluster belonging to a group in ISDS shares space with the physics clusters.

This white paper describes a model for University cluster support that recognizes the considerable benefits of this mostly-local physical and administrative organization, while extending it in ways that should yield clear benefits in scale, provide much better support to the local administrators that now do most of the actual work of cluster support without adequate training, and extend the advantages of cluster computing more consistently to groups that lack the local infrastructure to run their clusters on a departmental or group level.

The success of this model will strongly depend on its ability to meet a highly variable set of needs in a way that is perceived to be fair by the high performance computing community on campus. The University’s research is supported by grants from many agencies covering many fields of endeavor, with many distinct standards for what can and cannot be funded in terms of computing support. The cost recovery associated with the support model will need to be as flexible and as variable as these many sources of support.

An essential feature of the model is *adaptibility*. In addition to coping with a highly variable cost-recovery terrain (where some granting agencies prefer to fund systems including all support, others presume support to be already paid for in the indirect cost portion of the grant), the model will need to cope with the ever changing landscape of computer and networking hardware, and with cluster users ranging from tyro to expert. Any fixed and inflexible model will fail in time as the needs of the community evolve to where they are no longer being met.

The model presented is thus to be viewed as no more than a *beginning framework* for meeting the needs of the community. It is *expected* that it will be reviewed and revised as often as necessary to achieve its stated goals as the needs of the community change and as its design features are tested in application. The model is at heart an *evolutionary* model where new ideas can be tested and accepted or rejected as they prove viable.

2 A Model for Cluster Computing at Duke

The existing model for cluster computing at Duke is one of many *locally* centralized, generally autonomous, cluster computing operations. This model *works*, and it works for certain very good reasons. Well designed clusters, located in facilities that provide adequate infrastructure such as physical space, power, cooling capacity, and networking, scale extremely well in their system management requirements. That is, barring hardware failure a cluster node should require full-time equivalent (FTE) labor on the order of *an hour a year* or even less to install, update, and operate. In a department that already has a competent systems manager or systems management group, it is often possible to install and operate a cluster using opportunity cost labor provided by the local manager as just another aspect of managing the departmental LAN.

This is a particularly efficient solution, as the LAN manager *already* provides most of the core services required by the cluster (e.g. account management, disk and backup services, software installation and management services, and security) for the departmental groups utilizing the cluster resource. These services can be extended to the cluster nodes for essentially zero marginal cost, making the labor cost for installing and maintaining the nodes the *only* cost that scales with the size of the cluster, and this cost scales in a particularly predictable way.

This model is also efficient for a second reason. Since there are *many* clusters on campus, each engineered according to the needs of its local users and being perpetually built and rebuilt as new moneys become available, there is an *evolutionary* optimization that naturally occurs as new ideas are tried out, good ideas and bad ideas are discovered in *small* scale experiments, and these ideas and experiences shared across campus. This model works well in the rapidly changing world of computer and networking hardware, where "revolutionary" changes occur every year and are an accepted part of doing business.

This should be compared to the likely efficiency of a monolithic model where all cluster computer operations on campus where organized and managed by a single, centralized authority. Bad ideas would be costly on an *institutional* scale instead of a departmental or group scale; good ideas would have to diffuse into the institution from other institutions; change would necessarily proceed at a much slower rate. Worst of all, the cluster managers would likely become increasingly dissociated from their client base and increasingly narrow in their support of the wide range of user environments likely to be familiar to the cluster users. Accountability and flexibility would be lost.

These negative elements associated with monolithic models can all be observed *now* in those existing computer operations on Duke that are heavily centralized, especially in the realms of mainframe computing and in the generally homogeneous academic computing clusters[1]. Those of us who have been associated in some way with computing on campus over decades recall well the days of the Triangle Universities Computation Center (TUCC) and its campus equivalent (DUCC), and the inefficiencies that actively drove the primary computer users on campus to abandon this model altogether in favor of organization at the departmental scale.

For all of these reasons, the model proposed herein for improved institutional support of cluster computing *remains* a model that is centralized *locally*, at the departmental level where that makes sense and in a number of distributed cluster sites where it does not make sense. It *avoids* the creation of any sort of monolithic centralized cluster facility that might become the Duke Supercomputing Center (DSC) to mirror the North Carolina Supercomputing Center (NCSC) as DUCC once mirrored TUCC. It relies on institutional organization and coordination enabled by technology to achieve the desired support at the institutional scale while retaining the flexibility and cost efficiency of the localized management model.

The primary features of the proposed model are thus:

1. Mostly decentralized clusters, in a number of "cluster facilities" in reasonable physical proximity to their users, where those users themselves tend to be clustered, e.g. physics, math, computer science, chemistry, engineering, other science and engineering mileau with long-term needs for High Performance Computing (HPC). This simply recognizes that the existing model is fundamentally sound and should not be radically changed.

2. As an ¿¿extension¡¡ of the model, one or more cluster facilities (both existing ones and new ones) can successfully house clusters belonging to otherwise isolated groups that ¿¿don't¡¡ need to be in immediate proximity to their clusters. Again, as the examples of Math and ISDS, this is a viable model but needs to be promoted by Duke at the institutional level where a cost-benefit analysis or lack of local infrastructure make it appropriate. There are two possible models for managing these remote clusters. Both are likely to make sense for different kinds of clusters and cluster owners.

[1]This is not intended in any way as a criticism of academic computing, which does a truly spectacular job with extremely limited resources and has my greatest respect. Rather it refers to the fact that it is *so* large, and the base that it serves *so* diverse, that its ability to transform itself rapidly has been to some extent compromised, sensibly enough choosing to optimize reliability of service in established ways even when new models emerge that might reduce marginal costs.

3. One is the "owner managed" model, where the cluster is remotely sited but still managed by a departmental LAN manager of the department to which the owning group belongs. This is the *only* remote management model possible and in use (by Math and ISDS) at this time. It is obviously successful, for obvious reasons (it retains most of the zero-marginal cost advantages associated with local cluster administration).

   There are some additional cost penalties, however. The cost of physically managing and installing the nodes is considerably higher than with strictly local nodes, as it takes a relatively long time for the departmental manager to travel *away* from their primary departmental LAN over to the cluster site to perform such maintenance and installation duties that require physical presence. During this time offsite, their management of their departmental LAN is obviously somewhat less responsive. Similarly, they are necessarily less responsive to the needs of the cluster owners when those needs require a trip off site over to where the cluster is physically located. At a guess, offsite management by the systems manager of the owning group is roughly twice as costly per node as onsite management by the local systems manager of the owning group.

4. An additional model proposed for the management of these offsite clusters is that they be managed by "the university". This alternative model is one that we wish to architect and implement for a variety of reasons. Some research groups that might wish to operate clusters are in departments that lack the human infrastructure to support an offsite cluster, or the departmental LAN infrastructure to be able to realize any sort of economy of scale if they did. In addition, groups may find advantages in the resource sharing that is enabled if they locate their cluster under a common, university-level administrative umbrella with several other architecturally similar clusters. The construction of a suitable university management model for offsite clusters is a primary focus of this white paper, although that should *not* be construed as any sort of abandonment of the local management model (onsite or offsite) where it makes the most sense.

5. The existing local management model is not without flaws. Local managers at some sites have in the past been relatively untrained graduate students or postdocs, who have sometimes proven spectacularly incompetent or untrustworthy. Even when done by competent and professional local managers and there the considerable advantages associated with zero-marginal cost extension of the existing LAN services is obtained, the labor cost associated with running one or more on or offsite clusters is not necessarily either trivial or acceptable in any given departmental environment.

   Running a cluster in addition to a LAN involves tradeoffs that affect productivity in many ways, the most obvious one being that in many cases an administrator must choose to do one or the other, performing a sort of a task prioritization or triage as needs for services and support emerge. If the LAN manager is relatively underutilized, this is not generally a problem. If they are already heavily burdened, it can easily *over*burden them and result in a reduction in the quality of services.

Also, these local systems administrators are (generally) well-trained in LAN administration but may lack expertise germane to cluster management per se (where it differs). The construction of a university-level mechanism to better support and to better train onsite and offsite local managers is *also* a primary focus of the model proposed in this white paper.

6. In order to accomplish these goals of providing clusters that are fully managed by the University (offsite as far as the cluster owners are concerned), providing operational support to both onsite and offsite local managers, and providing improved training for local managers, the University will clearly need some sort of centralized cluster organization. This organization can improve productivity and efficiency at the institutional level in many, much needed ways. For example, in addition to the above, it can also help manage: cluster siting and the building or remodeling of facilities as needed; cluster tracking and inventory, grant-writing support, cluster architecture and standards, personnel support (both centralized and owner/local), application support, information coordination and dissemination, cluster integration both on campus and off (at NCSC, for example) and the management of the university-managed clusters.

This, then is an outline for a campus cluster support model that is fleshed out in more detail below. In it, clusters will continue to be both managed and physically located locally where it makes obvious sense to do so, as this results in by far the greatest economies of scale. Nevertheless, a University-level cluster computing operation will be proposed that will remain at least partly delocalized itself, and which will be responsible for providing a variety of levels and kinds of support to groups operating or hoping to operate clusters for many purposes throughout the University.

3 Cluster Siting

In this section we will discuss the advantages of cluster decentralization (or rather, centralization at a department-local level) in more detail, doing a cost-benefit analysis (CBA) of local management-local siting, local management-remote siting, and remote management-remote siting for a variety of typical cluster environments. The numbers presented in this CBA are a "best guess" sort of approximation and should be refined with actual numbers where available.

It is difficult to discuss cluster computing at *any* scale in completely general terms. On the beowulf list, "your mileage may vary" (YMMV) and "it depends on what you are doing" are the standard warning and answer to nearly any complex question. A cluster that works optimally (in the CBA sense) for one computation won't work at all for a different computation. For that reason, we need to differentiate clusters, and cluster problems, at a very early point in the discussion into two very generic classes:

- Problems (and clusters) that are very sensitive to cluster architecture and design. Typically these are problems with a relatively large communication-to-computation ratio, although it might well include problems with any sort of "unusual" bottlenecks or requirements (very large memory footprint, specialized network, very large or specialized storage requirement).
- Problems (and clusters) that are *not* particularly sensitive to the details of cluster architecture and design and that do *not* have any special bottlenecks or requirements.

Silly as this distinction may be, it is a crucial one. Problems and clusters that fit in the former group for all practical purposes *must* be engineered and operated on a per-problem, per-cluster basis by the group that uses the cluster. At this point in time the University simply cannot provide meaningful support for this sort of cluster computing at the institutional level. As time passes and the cluster support described in this document is (hopefully successfully) implemented that may change. At this time, however, it would be a capital mistake for the University to even consider anything but a local management model for this sort of cluster.

In is at least possible to describe some fairly "generic problems" that fit the latter description, and to describe a "standard cluster" architecture that should do just fine to solve them. Remote, centralized cluster management makes the most sense when the cluster has a very "vanilla" architecture that will work successfully on a wide range of relatively simple cluster problems. We will therefore focus most of our attention on problems of this sort.

To make the discussion concrete, let us consider an "embarrassingly parallel" application such as a Monte Carlo computation consisting of many fully independent sub-computations. We will presume that only a small amount of data is required to initiate a sub-computation, which runs for a long time on a single CPU and then returns a small amount of data that represents the result. Such a computation runs efficiently in parallel on any number of processors, requires little in the way of network speed or local storage, and doesn't globally fail if a single node goes down in the middle of its sub-computation.

In addition, we will consider a more challenging but still fundamentally simple problem such as a "coarse grained" lattice decomposition of some sort. Each node works on a part of some large space (lattice). To advance the computation many of the nodes have to communicate results between nodes before they can proceed, and if a single node goes down in mid-computation the entire computation dies and must be started over from the beginning. *However*, each node still does a *lot* of computation for a little bit of communications, and the computation can thus be scaled up to many nodes with a very generic network architecture. Also, the computation has no particularly special requirements in terms of local storage or memory and can easily fit on a fairly standard node design. However, it does generate a fairly large set of results, output continuously throughout the computation.

Both of these computations will run efficiently on a very generic architecture. Let us now analyze the costs of the different ways of siting the hardware and managing it.

A cluster supercomputer of any design is at heart a client/server LAN. Some of the costs of installing and managing a LAN scale with the number of servers. Others are fixed costs that don't scale at all. Still others scale with the number of clients, or the number of users. As is the case with any such LAN, primary costs for LAN construction, maintenance, and administration include items such as:

- Account management – creation, destruction, modification of fundamental access and groups privileges for all users of the system. Typically scales with the number of users independent of the number of clients, sometimes scales with the number of servers as well.
- Disk management – creation of shared server disk resources, their secure, authenticated exportation to LAN client systems, backup, retrieval. Scales with number of servers with a very weak dependence on number of clients.
- Network management – all aspects of managing both clients and servers on the network. Scales with number of clients plus number of servers.
- System installation – all aspects of installing servers and clients, depends strongly on operating system. In package-based linux, small cost that scales with number and kind of packages installed to get started, then scales with number of servers and (with an independent scale factor) number of clients.
- Software management should be a nearly fixed cost absorbed mostly into system installation and thereafter fully automated. Even so, there is at least a per-package fixed cost for setting up additions, modifications, updates to a "standard" list of software.
- Security – ensuring the integrity of all data and resource utilization. A large fixed cost associated with the entire LAN itself, with per server and per client costs (larger for the servers) and per user costs. Similar to, and related to, systems management.
- Systems management – monitoring status of all LAN elements, identifying and fixing problems, reconfiguration, and more. A large fixed cost associated with the entire LAN itself, with additional per server and per client costs (larger for the servers). Similar to, and related to, systems management.
- User support – dealing with the myriad of user problems that occur, teaching, hand-holding and more. A large variable cost that scales with the

number and competence of the users, the competence of the systems staff, the quality of the LAN hardware and design, the number of systems in the LAN, the number of tools in common use in the LAN and much, much more.

- Hardware support – repairing, replacing, disposing of all hardware as it ages out, arranging replacements for critical components in a proactive way, troubleshooting, and so forth. Scales with the amount of hardware, its quality, the load placed on it by all sources of hardware stress (users, programs, physical environment).

- Administration – paperwork and job related work of all flavors. A highly variable cost managed in different ways by different organizations. Scales at least weakly with number of systems and number of users both.

These are all services that must be provided and costs that must be paid for *any* LAN, including the specialized LANs we call a compute cluster or beowulf.

In addition, there are certain physical infrastructure costs associated with a LAN that must be tallied. These are not human or management costs (detailed above) but are nonetheless far from negligible.

- Power. Clients, servers, and network components are all electronic and consume electrical power. In very rough numbers it costs ~\$75 to provide 100 watts of electrical power twenty four hours a day for one year at \$0.08 per kilowatt-hour. In addition, any place more components are to be located than there is an immediate supply of electrical power will require remodeling and rewiring to achieve the required density in supply. This cost scales with number of components of any given power consumption, or total power consumed.

- Cooling. All the power consumed by any LAN component must be removed from the environment in a steady state way or it will build up as heat, damaging components and risking fires. Cooling occurs by many physical mechanisms in any environment including natural mechanisms, and the natural mechanisms vary in efficacy with e.g. the outside temperature, humidity, airflow, and details of the components physical location. We will assume (again in very rough numbers) that an electronic component that is consuming 100 watts of electrical power (all of which is continuously appearing in the immediate environment of the component as heat) will require roughly 33 watts of power, on average, to remove that heat. That is, \$25 per 100 watt component, per year. In addition, any place more components are to be located than there is local cooling capacity will require remodeling achieve the required capacity. This costs scales roughly with total power consumed by all components.

- Physical space. It is especially difficult to estimate the cost of space in a LAN environment. Every workstation location requires at least desk space for e.g. system unit, monitor, keyboard and mouse in an office/workspace environment. Servers and cluster nodes require space that is more typically fully dedicated to computers and provided with ample power and cooling. In that space, components can reach very high densities. The cost of the dedicated space may be "high" where it displaces humans or requires extensive remodeling (amortized, of course, over the lifetime of the space), it may be irrelevant (in new construction), it may be "low" when finding the space is a matter of cleaning out an unused supply room with plenty of power and cooling capacity relative to what you plan to put into it. There are additional nonlinearities in that small spaces may cost more or less, per component, than big spaces.

- Global network infrastructure. Access to the LAN backbone, and LAN access to the campus WAN backbone. The former scales roughly with the number of LAN environments or networked components, the latter is a fixed cost per LAN.

With these costs in hand at least by name, we are finally in a position to consider and compare the various location/management schemes.

3.1 Local management-local site

In a location that already contains an operational LAN, whose users include the owner/operators of a proposed cluster, *almost all* of the costs of installing and managing the cluster are already incorporated in the irreducible costs of setting up and running the LAN itself and supporting the cluster owner/operators as LAN users. Those users already have accounts, servers, desktop clients, managed filespace, security, authentication, software support, and routine maintenance provided by the local systems staff, and if that staff is competent most of those costs will *not* change tremendously if some network components are compute nodes instead of desktop LAN clients.

Exceptions (items which ARE additional costs for running compute nodes) are the cost of installing the compute nodes themselves, the cost of installing and maintaining any cluster-specific software, and the additional cost of supporting users with cluster-specific problems. There is a difficult-to-estimate cost associated with "FTE boundaries" that nonlinearly kicks in whenever the local manager(s) are pushed, because of the additional nodes OR workstation clients OR servers OR users past their capacity boundaries. If they're already working at absolutely full capacity without the cluster nodes, adding the cluster nodes will cost "more" than one expects from additions within their capacity. In addition, there are the more or less standard infrastructure costs of roughly $1 per watt per year plus the more esoteric costs associated with providing the space and networking required by the nodes.

Of these, the latter are for all practical purposes the same regardless of where and how the nodes are situated. There is generally no reason to expect *a priori* that space, power, and cooling will be more expensive in a nearby location (inside the same department) than far away. In specific cases it may be more expensive; in other cases it may be cheaper. We will therefore ignore the infrastructure costs altogether as being roughly equal for any physical siting of nodes, remembering that for any *specific* proposed space, we will need to reexamine the situation and see if that space is anomalously expensive or inexpensive relative to alternatives.

Thus the additional per node costs for a locally managed cluster in any environment where the cluster nodes themselves don't push the managers across a capacity boundary come down to the twenty or thirty minutes it takes to install the node plus the twenty or thirty minutes it takes to do all per-node management of the node in a typical year (including all hardware repair, software installation, software update). An FTE hour is a fairly safe upper bound on the yearly cost, per node, of running a standard cluster fully integrated with an existing LAN where the cluster owner/operators already have accounts and access to all the LAN resources.

There are some additional one-time costs associated with running any given cluster. Perhaps a cluster of 16 nodes requires a special compiler that it takes eight FTE hours to purchase and install (over several weeks) and later configure and support at the user end of things. Perhaps a cluster of 16 nodes has its own server, requiring an extra hour or two of setup time and a few minutes a week worth of attention to a backup device. It is reasonable to expect a cluster to take an FTE day or two *outside* the per-node costs to take care of this sort of thing. Altogether, however, a node will typically cost less, per cpu, than a LAN desktop even excluding the support of the humans that might use the desktop.

In the local management, local site model, absolutely maximal economy of scale is obtained except near FTE capacity boundaries. The system managers are on hand in the premises to take care of cluster nodes, so it doesn't significantly reduce their responsiveness to departmental LAN problems when they work on them. It takes a few minutes to walk to the cluster/server room to perform physical operations and maintenance instead of as long as an hour. Many physical operations can often be bundled into a single trip. Managers have maximal flexibility in choosing when to work on nodes and when to work on the LAN in general. The LAN management aspects of running the cluster are largely inherited from the LAN they are already running and not cluster specific.

It is hard to beat this model, which is why it is so popular and the obvious cluster model of choice where it works. The *marginal cost per node* for management is on the order of 1-2 FTE hours per year, or a real dollar cost less than $100 per year even for fairly skilled and well-paid managers. Better yet, as long as the cluster doesn't cross the FTE capacity boundary for the department, this 1-2 FTE hours per year per node is *free* – opportunity cost absorbed into the irreducible cost of the systems managers already working for you. You just use a larger fraction of the capacity you are already paying them for.

3.2 Local management-remote site

This model is very similar in cost to the former model. It presumes that the campus network backbone can be arranged and routed so that the remote site LAN can be made "local" as far as network routing, security, and network management are concerned. In particular, the most cost effective local-remote schema will be ones where NFS servers in the departmental LAN can be transparently, securely, and efficiently mounted by the compute node clients in the remote site. The network is "flat" across the two locations.

This can easily enough be set up between most departments and potential offsite but on campus cluster locations. The University wisely provide a large amount of overcapacity in terms of fibers and switching when designing the University backbone, so it is fairly straightforward to set up a pipe from (say) the physics server room directly to (say) the ISDS LAN so that the ISDS cluster in the physics server room is completely within the management boundaries of the ISDS LAN.

In this case also, then, all the LAN-specific aspects of setting up a generic cluster and making its nodes available to the owner/operators within the departmental LAN are *also* zero marginal cost, per node. Again, the irreducible charges are node installation and maintenance, cluster specific modifications and software management, and dealing with the special problems and needs of cluster users (in addition to the standard infrastructure cost of $1 per watt per year).

The one important additional cost is the extra time required to do anything physical with the nodes themselves. Since the nodes are in one location and the manager in another location altogether, the manager must leave their LAN and go to the node location to do things like physically boot a node with the power switch or reset button, pull a node that appears to have broken and diagnose the problem, install a new node or set of nodes. The time spent in transit, in particular, is lost relative to the local-local model. However, this makes it *more* likely that an FTE boundary will be encountered, especially if there are a *lot of nodes* at the remote site so that frequent trips are equired.

In addition, there are nonlinear costs associated with the lost productivity of individuals working in the department LAN who require immediate service when the LAN manager is offsite. In the worst case scenario, the LAN crashes the minute the LAN manager has left. Even if they return immediately once the reach the offsite cluster location and notice that the LAN is down or some crisis has occurred, it can easily cost an hour *more* of downtime and lost productivity for an entire department's worth of LAN users during the delay due to transit. This is one of many reasons that LAN managers don't like to have jobs split across locations, clusters or not.

It is worth noting that *many* of these additional costs and inconveniences can sometimes be avoided if there is a small surplus of FTE management capability at the physical site. In that case, it may be possible to just send email requesting that a site-local manager visit the server/cluster room and toggle power on a downed node, or even do the physical installation of a node (which may be nothing more complicated than racking the node, cabling it, and turning it on).

On a more organized basis, once some sort of centralized management entity is created, it may be possible to achieve something very similar at a higher level. The central management group might be running clusters in some of the same sites. As a concrete example, the physics cluster room might host both an ISDS cluster and a public cluster in addition to physics' own local clusters. It might well be that the public cluster is managed by one of the physics LAN managers (who is partially funded by the University for doing this particular job) or by the physics LAN management group (where all or part of an FTE in that group is provided for managing the public cluster). It would be a simple matter indeed to make an additional responsibility of that individual the physical (not LAN level) care of the remotely sited ISDS cluster, so that any routine work of pulling downed nodes for service or installing new nodes was done by this group without necessarily requiring the physical presence of the ISDS LAN manager.

3.3 Remote management-remote site

In the previous two cases the particular model of the parallel program being run didn't much matter. In either case, the program was being run "locally" and so the user's standard NFS filespace and/or any special project space was likely mounted and immediately available on the user's own workstation and the cluster node alike. Backup of any critical data was already arranged within the preexisting backup paradigm of the department. Even things like visualization were likely transparently supported on the LAN workstations and the cluster nodes alike. There were essentially no additional costs associated with the management and secure utilization of multiple accounts or transport of data across LAN boundaries.

When the cluster in question is *remotely* managed, by a centralized University entity, this is no longer true. Cluster access and data transport will necessarily cross LAN administrative boundaries, and boundaries of trust. Cluster users will require a *cluster-local* LAN infrastructure to support their cluster computation, *cluster-local* accounts, authentication, security, and fileservices. Cluster users will require mechanisms for accessing the cluster and transporting their data to and from their department local LANs. Furthermore, since the remotely managed cluster may well be a shared entity with access split among many groups, the cluster itself will likely need a variety of cluster management tools to be installed that facilitate e.g. remote monitoring of jobs, batch job submission, job level accounting, and muchmore.

In essence, a core management LAN structure has to be created for the cluster. The cluster manager will need to manage accounts, security, users, *and* coordinate those structures with those of a variety of departmental LAN managers, where the cluster users have their primary accounts and the workstations through which the cluster will likely be accessed. Finally, the cluster will require more tools to be installed to support remote cluster access and monitoring, and making those tools effective will require more user support and training.

At best, a remotely managed, remotely sited cluster will thus require *substantially more* FTE management than will a locally managed cluster, however it is sited. It thus behooves us to consider ways that we can minimize this additional expense and recover a reasonable fraction of the cost-efficiency of managing a cluster within the cluster owner/user's LAN.

The obvious approach is to piggyback the LAN management aspects of the cluster on top of an existing LAN that already offers services to the entire University community. This approach permits us to realize substantial economies of scale. Although not as great as the economies that exist for LAN-local management (since the cluster will willy-nilly not be directly integrated into the owner/user's LAN) such an approach permits the University itself to gain important dual benefits, described below.

The unique solution thus appears to be to fully integrate the cluster with the existing Academic Computing group (acpub). Any member of the University community already can get an account within acpub, and for a variety of reasons most faculty, staff, students, and even many postdocs already have done so. Acpub provides at least one mechanism for institution wide disk authenticated access (AFS on top of kerberos) that is already, for the most part, supported to the departmental LANs likely to host the owner/users of centrally managed cluster facilities. One presumes that any additional LAN resources required by the cluster (e.g. local server disk pools, backup mechanisms, additional cluster management tools) could be scalably provided within their existing LAN support framework, minimizing cost and maximizing integrability.

This does create certain organizational issues that must be dealt with. The staff that runs the public cluster(s) cannot just "be" the acpub staff, as the acpub staff likely lacks core expertise in cluster construction and management (the same problem that exists in the locally manged clusters out in the departmental LANs). Also, at this point acpub is not *primarily* based on linux, and although their staff is far from incompetent in linux, neither are they the campus's primary experts. They are thus unlikely to be able to scalably extend their existing staff and services to clusters without augmenting their staff with one or more cluster and linux experts. Those experts would need to have the freedom to support the clusters "semi-autonomously" – integrating with and drawing upon the scalably extensible services acpub can provide for account management, access, authentication, and possibly disk services, while not being forced to strictly conform to the acpub workstation model.

One advantage of providing centralized cluster management within the acpub hierarchy is that it will provide acpub with a straightforward route for migrating from student clusters based on proprietary hardware and operating systems (e.g. Sun and Solaris, Wintel) to linux clusters. This is a desirable migration for many, many reasons (tremendous direct cost savings in software, greater security, the ultimate degree in system installation and management scaling, and open standards for a variety of document and data protocols that facilitates e.g. long term archival storage and retrieval of critical data). Acpub will virtually "inherit" the ability to build and manage scalable linux workstation clusters from the ability to build and manage scalable linux compute nodes; as noted above, compute nodes are just a specialized variant of a workstation from the point of view of installation and management.

It should be pointed out that this model for cluster support already works for at least the embarrassingly parallel class of tasks described above. The author of this document ran embarrassingly parallel Monte Carlo computations on the entire acpub collection of workstation clusters for close to a year some years ago (with the permission and support of the acpub staff), getting a phenomenal amount of research computing done before weaknesses in Solaris forced this experiment to be terminated. This extremely simple model of account+workstation or node access would likely fail for distributing true coarse grained parallel tasks in a multiuser environment, but a *simple* extension of it very likely *would* work, and that is what is proposed below.

This document does not attempt to suggest the details of how cluster and linux management might be integrated with the existing acpub staff and group; only that this is by far the most desirable way to proceed as (once a modest cost penalty for the initial startup is paid) it maximally leverages existing modes for delivering University level compute services to University personnel in all venues.

It will have the temerity to suggest that this be done delicately, in a way that carefully avoids crippling either the existing acpub staff or the cluster staff that would be working with them, and with full respect to the FTE capacity boundaries that undoubtedly already exist within acpub. As in, don't try to make acpub simply absorb the additional burden of cluster support with their existing staff. It will *also* have the temerity to suggest that it be done in a way that integrates existing methodology for providing the core linux installation and support services *outside* of acpub (as they are now, via e.g. the dulug site, with the integration of a "virtual staff" of linux experts and cluster experts that are not "in" acpub per se but are still charged with providing support and training services), crossing boundaries of administrative control.

A major point of the model proposed for cluster management on campus is that it remain *decentralized* as much as possible with support mechanisms that cross boundaries of administrative control even where it attempts to provide a basis for centralized management and access. It centralizes where one can see an immediate and clear CBA advantage in terms such as zero marginal cost extension of existing LAN management structures.

This is not at all paradoxical – linux based cluster management and support *currently* spans the entire globe, with linux and cluster specific development, instruction, and training coming from an international community. This model for support provides the greatest possible basis for experimentation, evolutionary optimization, and the rapid dissemination of the best and worst solutions throughout the institution, and encourages an open consensus model for technology engineering that ensures that the broad needs of the community are continuously met. If you like, it keeps the customers of any given service in close contact with the service providers, as the two groups are mixed.

In the next section we will discuss in some detail the economies of scale associated with a truly distributed support model. This section will be quite specific in its suggestions for how to integrate a "centralized" cluster facility into the existing linux and cluster community.

4 Centralized Cluster Installation and Management

Before discussing how to integrate a centralized cluster organization, loosely connected to acpub (utilizing at least acpub's existing LAN mechanisms for account access, authentication, and so forth) with the existing collection of linux-based LANs and departmental clusters, we need to describe the way linux is currently installed and managed in a typical LAN environment. This will explicitly illustrate its phenomenal scalability in the existing University support environment, and indicate clearly how and why it is essential to *preserve* this in any sane clustering model.

4.1 The DULUG linux installation site

The key to scalable linux installation on campus comes down to several things:

- Red Hat linux, a linux distribution based on self-contained packages of software called "rpms".
- kickstart
- install.dulug.duke.edu
- yum

In a nutshell, Red Hat's current distribution(s) are mirrored to install.dulug.duke.edu and augmented with rpm (re)packaging of Duke's site license software and any other software found to be useful that is missing from the primary Red Hat distribution. Certain tools and configuration features are customized for the Duke environment. install.dulug.duke.edu is accessible only to duke.edu addresses.

From install.dulug.duke.edu, any member of the University community can install a working version of linux on nearly any networked computer, over the network, for free, in a matter of ten or twenty minutes. No particular expertise is required other than the ability to follow simple directions, although installing a computer located in a departmental LAN is best done with the support and cooperation of the LAN manager.

Kickstart is a Red Hat linux feature that permits the installation of a workstation or cluster node to be "scripted" from a straightforward template. Work done once developing a single LAN-specific kickstart script can then be used many times to install workstations, servers, cluster nodes. This is the installation tool of choice for LAN managers on campus that run any significant number of linux systems in their departmental LAN. With kickstart, to install a system (or reinstall a system) one simply boots it, either from floppy, from the local hard disk, or (using a PXE-enabled ethernet card) from the network. Ten minutes later (or less, depending on bandwidth to and load on install.dulug.duke.edu) the node reboots itself into a fully installed and configured state.

Yum is a tool authored by Seth Vidal (the systems manager of the Duke Physics Department, who also maintains install.dulug.duke.edu) that fully automates various aspects of the maintenance of workstations, servers, and nodes after their original install. It is automatically installed and configured to run nightly on any Duke system installed from install.dulug.duke.edu that does not explicitly choose to exclude it. With yum, updating every linux-based system on *campus* can be as simple as dropping the updated package in the primary archive on install.dulug.duke.edu – yum will automatically install the updates from a nightly cron script.

This is *crucial* feature as it *significantly* increases the security level of every linux system on campus equipped with the feature – security updates are distributed within as little as 24 hours of the discovery of a problem, even onto systems belonging to totally system-ignorant individuals who would otherwise never hear of the problem or think to update their systems. Yum also greatly simplifies all the aspects of rpm package management for systems or LAN managers as it forms a consistent interface to the entire package collection on install.dulug.duke.edu (and/or other yum repositories set up in departmental LANs with specific site-local packages). Installing, removing, listing and otherwise maintaining packages is reduced to a single, simple command with a few options. With it and a simple script, a LAN manager can (for example) install a new software package on every system in their LAN with a single command typed once at their own desktop.

As long as the primary job of maintaining install.dulug.duke.edu and yum continues to be done by Seth Vidal (with the help of various other linux experts on campus, but his primary responsibility), one can see that his labor enables linux installation and software management to be done at the practical limit of efficiency (and amazingly close to the *theoretical* limit in efficiency). It isn't *quite* true to say that to install a new, PXE-equipped system in a campus LAN one has but to turn it on, but only a small amount of work at the LAN level (developing a kickstart file, setting up PXE and dhcp, writing the LAN-specific post install scripts that make the system a trusted member of *this* LAN and e.g. mount the correct server directories rather than a standalone untrusted host) makes it true.

4.2 Cluster nodes

Workstations tend to be extremely heterogeneous in their hardware configuration. The incredibly wide variety of over-the-counter motherboards, cases, CPUs, memory options, sound options, video options, storage options, and even keyboard and mouse options, is wonderful from the consumer point of view and keeps prices low, but is a real headache from the systems support point of view. At this point, linux supports most PC hardware fully automatically in a kickstart install, but certain e.g. video cards do require some options to be specified or tweaking to be done on a per-system basis for a workstation.

Heterogeneity thus adds to LAN management costs. Sometimes (when a particularly troublesome piece of hardware is encountered) this cost is significant. LAN managers encourage users to purchase systems with "approved" hardware (hardware known to be both supported and relatively headache free) to control this cost. As long as hardware from this list is used, a single kickstart file will typically suffice to use for any LAN workstation and costs are minimized.

The same is true for cluster nodes, which can be viewed as specialized, particularly simple workstations. As long as cluster nodes are engineered so that all components are well supported (ideally in a tested and trusted configuration) a single kickstart file can suffice to support all the cluster nodes in a LAN. Even as nodes with new hardware are added over a period of years, it is simple to make small changes in copies of the basic kickstart installation and support several generations of node at once.

The *basic* cluster node configuration (and hence kickstart file) is unlikely to change signficantly between clusters or departmental LANs. A basic cluster kickstart file (such as the one used in physics for cluster nodes) can thus be shared at the institutional level, with obvious LAN specific modifications. Cluster nodes can thus be installed for a very low average cost as long as hardware and configurational heterogeneity are avoided. Cluster hardware can be prototyped at the institutional level to ensure that nodes are scalably installable and maintainable.

From this we see that cluster nodes can be installed and their software configuration maintained, at the institutional level, for *very low cost* per node. *If* one utilizes acpub's institution-wide LAN configuration as the LAN basis, *and* one sticks to "proven" node hardware configurations, *and* use a standardized, widely shared "cluster node" kickstart configuration, one pays a fixed cost for integrating these elements for the first cluster, a small fixed cost for additional cluster-specific servers or hardware, and then a marginal cost of a few minutes per cluster node for installation and software maintenance per year. The only uncontrolled costs remaining are user support (which can be highly variable, depending on the skill level of the user and the complexity of their task) and hardware support.

Because the scaled (per node) costs can be so precisely estimated, it is possible to create a simple model for cost recovery on cluster node hardware, installation, and basic software maintenance that is likely to pass muster from the vast majority of granting agencies. It is left as an open question as to whether or not to attempt to recover physical infrastructure costs (the estimated \$1 per year per watt) for running cluster nodes. It is recommended that the University *not* attempt to recover costs for the basic LAN services and user support required, and instead view them in exactly the same way that acpub services are viewed now (and indeed, offer them as an *extension* of acpub's basic services).

It is my personal belief that this model will prove to be satisfactory and even attractive to virtually any researcher interested in doing cluster computing who is in an environment that simply will not support the local management model (that is still cheaper and more desireable). The model itself is described next.

5 The Standard Node Approach

The University already (re)sells computer hardware via the Duke computer store. When cluster nodes are purchased from a cluster integrator, or a value added reseller, one buys both the hardware and the "integration" – basically the charge for pre-installing linux in a suitable cluster configuration and building one or more server or head nodes.

As one can see from the description above, in most cases the "integrated" clusters thus purchased require further effort to actually integrate into a LAN environment. Accounts need to be installed, disk resources shared across the LAN need to be managed, users need to be supported, specialized software not pre-installed by the integrator needs to be installed. One then is left with a dilemna in long term maintenance – one particular snapshot of some linux distribution is installed on the hardware, and one has to work quite hard to arrange for this distribution to be updated or augmented. All this work is a considerable added cost on *top* of the integration fee charged by the cluster integrator. It is actually *considerably more costly* to the University to install a pre-configured "integrated" cluster from a vendor and perform the work required to insert the cluster into a preexisting LAN environment than it is to just reinstall the cluster nodes themselves from kickstart!

In the previous section, we saw that Red Hat, a campus wide installation archive, kickstart and yum enable anyone to install a "cluster node" once these tools are customized for their particular LAN environment. In the case of a University-run centralized cluster using the acpub LAN environment, this work amounts to precisely the same "integration" sold by many vendors *customized to perfectly fit the actual LAN environment and account scheme.* These costs are perfectly predictable (to a point) and nearly perfectly scalable once the initial development and deployment cost is paid.

A suitable model for cost recovery for the University is thus *to "resell" integrated compute nodes that can be inserted into the publically run clusters!* Utilizing a mix of local vendors such as Intrex and web-based vendors such as Dell and MicroWarehouse (ideally with pre-negotiated special pricing) to provide the actual cluster nodes and required hardware services and extended warranties, the University should be able to easily match the margins of any commercial integrator and achieve *true and full* integration using the schema described above in fair detail.

An example of the cost scale and cost recovery for a University resold compute node might be (noting that these prices are approximations based on a considerable experience purchasing more or less standard nodes):

- A "standard node" in a 1U configuration: $1000.
- Three year hardware service: $100.
- Integration fee: $100.

for a total price to the end-user of $1200/node exclusive of network switch capacity, cabling, and miscellaneous hardware. The "integration fee" would cover node installation (one hour FTE) and an expected two FTE hours of node-level attention, on average, during the node's three year expected lifetime, at roughly \$35/hour, and presumes that the hardware service fee eliminates "all" the cost of hardware service to local systems staff after perhaps diagnosing the problem.

This estimate will need to be adjusted to reflect reality on the basis of experience as the project proceeds. For example, nodes may cost only \$1000 *with* extended service, or may cost \$1500 in a higher memory configuration. Both single CPU configurations (presumed in the \$1000 price tag) or dual CPU configurations (which would likely cost about \$1600 for equivalent memory per CPU and hence provide slightly better cost scaling for certain classes of problems) are likely to be options. It may be that \$100 per node provies inadequate to recover the actual installation and management costs, on average.

These costs for an integrated node, resold on a break-even or win-a-bit basis, are highly comparable to the costs one would obtain on the open market, providing University researchers with an attractive alternative to doing it yourself or to having it done by a profit-seeking outside integrator. This is *essential* for any sort of centralized cluster effort to succeed. The University will *not* attract researchers and grant money to populate University-run clusters if the researchers perceive of the cluster nodes thus obtained as being significantly more expensive than market value of similar integrated nodes. Why should they buy a turnkey cluster "from Duke" if they can buy a turnkey cluster from any of a dozen vendors?

True, those turnkey clusters are far from ideal and actually have a number of hidden integration costs when inserted into an actual LAN environment. However, research groups will accept those risks and hidden costs if it means that they can buy signficantly more nodes and get hence get significantly more work done, and worry about dealing with the difficulties with the nodes in hand.

An extremely important feature of this "integrated node" approach is that there should be basically *no difficulty* in justifying the cost of integrated (turnkey) cluster nodes purchased from the University, any more than there is for integrated cluster nodes purchased from an outside vendor, as long as those nodes are cost-competitive.

A final important feature of this approach is that the purchaser *retains "ownership" of the nodes.* The nodes are basically prepaid for University level management in one of the various University cluster sites, but they belong to their purchaser. The purchaser can dictate to what extent they participate in any sort of resource sharing program. The purchaser can, if and when they have some alternative way of siting and managing the nodes, recover and move their nodes (at their own expense). This means that racks populated by cluster nodes that may well belong to several groups who may well choose to share them will not violate any laws or outrage the sensibilities of the groups that purchase those nodes.

6 User Support

If the University adopts the acpub general LAN configuration as suggested above, the only *non*-recovered costs for managing public clusters will be a very limited amount of one-time work developing cluster images consistent with the acpub LAN, some work on support-level infrastructure for the managed clusters (building and managing cluster servers and so forth, which will likely not always be paid for explicitly by node purchasers) and the extra human work required to support the cluster users.

If the University adopts a mixed model for user support such as the acpub help desk plus mailing lists and websites that range from campus-specific resources such as dulug and dbug (the Duke Beowulf User's Group) to international Internet-wide resources such as the beowulf list, the many beowulf websites and resource centers, the many linux and gnu lists and websites, and even to internal or external paid consultants (paid by the uesr) for particularly thorny problems. Using this mixed model and leveraging existing resources, even the highly variable cost of providing user support can largely be controlled. This support typically comes in two forms (motivating the mixed model). On the one hand there are requests for routine LAN services such as setting up accounts, arranging for resource access, or requesting help with more or less routine linux software, which the existing acpub help desk can manage at little to no additional marginal cost (presuming that the existing acpub staff is augmented by 1-3 cluster specialists working semi-independently on managing and running the public compute clusters).

The other kind of support needed is cluster-specific support for *how to use* a compute cluster – how to write parallel code, distribute embarrassingly parallel jobs efficiently, how to collect results locally at the nodes and retrieve them to the user's home LAN environment as needed, how to enable secure ssh access to node/cluster resources across the campus WAN. This latter need for support is much better provided at the user-group level, where all the cluster users and cluster administrators on campus pool their collective expertise and help out the less-experienced users, where one expert can offer up complex problems for input from other experts.

This distributed model of support is "the" model for virtually all of linux and open source software, and has proven to be shockingly effective at facilitating learning and development. It forms the critical information-exchange step in an ongoing process of genetic optimization that is responsible for the birth of the Internet itself and which continues to drive a staggering range of technological development today. It is also very nearly a model for "free" support. Most of the actual support (again, based on years of experience participating actively in many of these lists) comes from people who provide this support or are heavily involved in cluster or systems management *anyway*; the service is provided out of opportunity cost labor by employees and other members of the University community that are in some measure already being paid to provide this sort of service, or who are at least professionally engaged in some measure with cluster construction or operation.

It is important to remember that this sort of distributed support is *not* really free and that it *is* a highly valuable contribution to the University community, whereever it comes from. Although it is not generally necessary to explicitly pay for this kind of support with a line item charge, toplevel IT managers across the University need to be aware of the time being contributed to cluster support, both to be able to recognize the contribution when evaluating employees' performance and advancement and to be sensitive to the need to augment staff when participation pushes any given employee or employee group to an FTE boundary (where they no longer *have* "spare time" in which to answer cluster-related questions).

Since some of this distributed support will be provided by faculty, postdocs, and other non-staff employees, their contributions *also* need to be recognized and rewarded somehow. Cluster experts throughout the University are an important community resource and enable research to be done and grants to be obtained far beyond the boundaries of their particular departmental mileau.

7 Physical Infrastructure

As detailed above, a reasonable cost estimate for the recurring physical infrastructure (power and cooling) costs for maintaining a cluster node in *any* on-campus site can be estimated as \$1 per watt steady-state power consumption per year. This does not include the cost of renovation of a cluster site, the cost of racks or wiring trays at the site, nor does it include any sort of "rent" on the physical space provided at a particular site for a cluster node as these are all one-time capital infrastructure costs.

It is difficult to know what to do about these costs as far as cost recovery is concerned. The recurring physical infrastructure cost for operating a node is not trivial at roughly \$100 per CPU per year, but neither is it trivial for operating a desktop workstation within any department on campus, or for providing power and cooling for any piece of experimental apparatus in a lab. In some cases (such as powering the TUNL particle accelerators or the FEL) those costs are tremendous and must clearly be borne by a funding agency as a line item. In most other cases, they are considered to be part of the infrastructure already paid for in the indirect cost portion of a grant. To the best of my knowledge, the University doesn't put separate electrical meters on each lab space throughout campus and attempt to backcharge the researchers in each space for the power and cooling they happen to consume.

Similar considerations seem to hold for renovations to space required to site clusters within departments or elsewhere on campus. In many cases the University provides space renovations as part of the startup package offered to attract faculty hoping to build some sort of experimental program, or renovates space to meet the changing needs of established programs as they grow and alter their focus. Large programs or programs requiring new construction, however, might well fund the construction or renovation out of grant money. Again, it seems reasonable to assume that indirect costs already charged to most grants suffice to cover at least a "reasonable" amount of space and renovation effort if it is required to support a funded project.

Clusters seem to be right in the middle. For many researchers, they are just another piece of essential equipment. Grants to theorists, mathematicians, computer scientists, and statisticians in particular have paid indirect costs at basically the same rate as experimentalists for years while generally requiring little more than an office and access to the library to support their research. As times change and clusters become and integral tool for researchers in these fields, it is not at all unreasonable for them to expect the same sort of infrastructure support for their essential equipment as has always been given routinely to the experimentalists.

On the other hand, a 128-CPU cluster can cost on the order of *\$13,000 per year* to power and cool. Its space requirements are not huge (a few square meters of floor and rackspace) but the capital investment cost of renovating that space so that adequate power and cooling density is achieved in any given location may not be small (order of \$50,000 to \$150,000, amortized over perhaps ten years and split up among hundreds of nodes). The economics and CBA of indirect cost recovery from the grants that presumably will have cluster nodes housed in the facility over the course of many years is not as easy to determine, and a compelling case would likely have to be made to make at least some of the many granting agencies comfortable with supporting it as a line item in all but a few special cases.

It is therefore *suggested* that no attempt be made to recover this cost at this time, at least on a routine basis. In the case of smaller clusters (5 to 10 kilowatts), it would indeed be unreasonable and should be considered to be paid out of indirect costs on the supported research done with the cluster nodes thus housed. In the case of larger clusters (up to perhaps 25 kilowatts) it could be argued either way, but not *convincingly* argued without hard data that can only be collected deliberately, over time, by looking at the indirect cost balance of actual clusters operating *in situ.*

For clusters larger than 25 kilowatts (more than 256 CPUs, to use another measure) the capital outlay from the granting agency is *already* expected to be in the hundreds of thousands of dollars. For clusters of this size, granting agencies have to expect to pay various additional charges because operating a cluster of this size even within a pre-existing LAN with zero marginal cost extension of the LAN workspace is likely to require a significant fraction of an FTE manager. Although some of these costs may be recovered for centrally run clusters utilizing the integrated node approach above, within a departmental LAN they are not usually so charged even though the additional administrative burden can easily push the local administrator(s) across an FTE capacity boundary.

Finally, *indirect costs are not generally charged on capital equipment.* This can create an imbalance precisely *where* a low-overhead research project (one that funds only one or two salaries and the cluster) is concerned. Overhead on the salaries may not cover the actual cost of operating the cluster if the cluster is large, and there are no indirect costs charged on the cluster hardware.

For this reason it would not be unreasonable to *try* to recover recurring operational costs such as power and cooling at fair market value (exclusive of the cost of any renovations required, for the most part) for very large clusters where this imbalance is likely to occur and where granting agencies are likely to recognize this and tolerate the additional expense. However, it would be advisable to proceed here on a case by case basis, taking into account the kind of grant being sought, the amount of cluster node equipment being purchased, and the actual amount of indirect cost monies the grant will generate.

8 The (De)Centralized Management Group

In this section the primary duties and responsibilities of the core group responsible for supporting cluster operations on campus are articulated. As noted in the various sections above, this group will be responsible for:

- Coordination of all cluster operations and management throughout campus, especially the group providing the services below. (Bill Rankin)
- Maintaining the primary linux infrastructure (Red Hat, install.dulug.duke.edu, kickstart, yum) upon which *all* scalable cluster installations on campus will primarily rely. (Seth Vidal)
- Augmenting and extending these fundamental resources as needed by special groups, with e.g. SGE tools or Scyld licenses. (Seth Vidal and Bill Rankin)
- Constructing and maintaining integrated public clusters made out of "standard nodes" as described above, in several physical locations. (Bill Rankin, the Academic Computing group, and possibly others in the various schools e.g. Feri Zsuppan, Sean O'Connell)
- Providing cluster training and all forms of support for LAN managers that run or expect to run locally managed clusters (whereever they might be sited). (Bill Rankin, Seth Vidal, Robert Brown, Jeff Chase, Sean O'Connell, Chris Cramer, and a cast of thousands coordinated by the above. Possibly special training programs coordinated by e.g Wake Tech as well.)
- Coordinating user support *channels* (not necessarily directly providing that support, although in many cases). (Bill Rankin, Robert Brown, acpub help desk)
- Providing more direct consultative cluster user support as permitted by their other responsibilities, helping users locate paid consultants or programmers where appropriate. (Bill Rankin, Robert Brown)
- Testing and prototyping node hardware and software configurations (Bill Rankin, Robert Brown, Seth Vidal...)

- Establishing (with prototypes), negotiating and outlining purchase procedures for standard nodes resold as integrated nodes in the centrally managed clusters (Bill Rankin, Seth Vidal, Robert Brown...)

In parentheses after each chore a list of names is given for the expected initial assignment of responsibilities. It should be noted that most of these "assignments" de facto recognize that this work is already being done, for the most part, by these individuals or would be clearly expected to devolve to one of them once this model is approved and adopted.

The first entry, coordination, is perhaps the most important task on the list. The University *has* an amazing amount of the infrastructure required to support cluster computing on nearly any desired scale in place. What it primarily lacks is a clearinghouse, a central entity that can connect up this infrastructure and support mechanism with those that might wish to participate. It also is very much a *de*centralized operational model. Individuals from the University (Bill Rankin), OIT (the acpub group), Arts and Sciences (Seth Vidal), and even the faculty (Robert Brown) all play key roles in establishing the initial "centrally managed cluster". Support on a broader basis comes from the entire University cluster community. Initially, at least, very little in the way of additional staff should be required to get the project off the ground, but coordinating the staff and contributions we have (from all over campus) will very definitely require someone working very hard to make it so.

However, there is one very important caveat to this very optimistic staff layout. Some individuals who play key roles in this project (notably Seth Vidal) are obviously *immensely* valuable to the University already and essential to the success of the model. *Because* of their value they are already heavily overburdened. Great care must be taken to prioritize their task assignments and provide additional support to their local management groups to in some measure protect their time and sanity. Seth can and does manage install.dulug.duke.edu. He very likely can help a great deal working with Bill to come up with a universally accessible, nearly fully automated cluster node kickstart file (for example) and floppy or PXE images that automatically access it. However, Seth also has primary responsibilities managing the physics LAN and secondary responsibilities helping out systems managers in many other University departments. He simply cannot be spread indefinitely thin by adding extensive cluster training and support responsibilities to this list *beyond* what he already does voluntarily and as time permits on e.g. the dulug mailing list, at least not without further augmenting the physics system staff to partially free some of his time.

The same is true to a greater or lesser extent for all the decentralized participants in this organization. Robert Brown already provides ongoing cluster computing support to groups all over the world via the beowulf list and there is no reason to suppose that that support wouldn't extend to anyone at Duke who needs it *time permitting* (given his other responsibilities of teaching, doing research, and raising children). Jeff Chase is similarly engaged in teaching and research. Sean O'Connell (and the various other systems managers with cluster experience who would almost certainly participate in the decentralized staff) has LAN responsibilities that are primary, but otherwise would cheerfully contribute time and energy via a list or cluster support group.

This point has been made repeatedly above and is worth repeating yet again. The "decentralized" model for centralized support proposed above is entirely consistent with the philosophy and *reality* of cluster management as it has evolved on campus, providing greatly improved and somewhat centralized support for (almost) "free" (out of opportunity cost time and zero margin scaling of FTE effort already being paid for around campus, plus the cost of a coordinating central group that can "fill in the cracks" created by the decentralized approach). However, whereever providing that support pushes key individuals to FTE boundaries, the University and its participating schools and entities have to be prepared to increase staff or redirect responsibilities to make it work. The model is scalable and largely self-supporting, it is not free. The way this is funded and accomodated may require some genuine cooperation and honest accounting for value between many disparate branches of the University; it is the view of this plan that this is all to the good.

It is also the editorial opinion of the *author* of this plan that this is actually possible at this point in Duke's IT history. Ten years ago this model would have laughably and expensively failed as someone or other attempted to build a centralized "cluster computing empire" and have it funded out completely out of proportion to the services delivered. Today the situation is entirely different, with genuine cooperation and coordination between all the various levels of IT across campus and a sense of collegieality and commonality of purpose that transcends funding models and empires. This model will likely test the University's ability to cooperatively implement things efficiently across many administrative boundaries, but given the success of netcom, of acpub, and of other groups that operate in just that matter there is no reason to doubt the possibility of success here.

Regardless, it is very likely that implementation of this plan will require adding one or two more FTE's (probably under the direction of Bill Rankin and/or Rob Carter) over the course of its first year, at least if it is at all successful in "selling" public cluster nodes. If it is *not* so successful, it minimizes startup investment in the first year in any event and can easily enough be modified to reflect experience at the end of a year.

9 Conclusion

In the sections above, a model for providing centralized (University operated) clusters within the generally decentralized University cluster environment has been proposed. The design maximally leverages existing administrative resources, improving cluster-specific support to local administrators and using the already existing LAN structure of academic computing as a base that should significantly lower overall administrative costs. It should be capable of providing remotely sited, remotely managed cluster resources to groups physically located anywhere on campus at a cost that is both fair and *perceived* as being fair by granting agencies and the research groups that own them. Finally, the model itself is one that can fairly easily be reevaluated on an annual basis and adjusted as necessary as actual measurements of cost and utilization are developed and a proper CBA becomes possible.

The model is scalable, and should be able to accomodate anywhere from a hundred to thousands of University managed nodes, with direct recovery of most of the actual costs of installation and operation, recovery of recurring physical infrastructure costs from indirect costs charged to the associated grants, and perhaps a gradual investment in an overall increase in the core infrastructure provided by OIT to the University as a whole (funded out of all the University's revenue streams) in a way that is entirely equitable as clusters become more and more common and more and more important in the University's grant-funded research efforts and educational mission.