**CC*DNI Engineer: Data Safety, Transfer, and Performance**

**Current, Ongoing, and Proposed Projects requiring a CyberInfrastructure Engineer.**

**Introduction:**
Our CyberInfrastructure (CI) Plan emphasizes that UCI is working to raise both the floor as well as the ceiling in terms of effective use and availability of CI. We take our responsibility to the Social Sciences as seriously as to Engineering and are working to encourage all faculty to make use of our aggregate expertise to enhance their research productivity. For example, we are investing time in answering problems in file-sharing not only for Terabyte (TB) data sets (via GridFTP and bbcp), but also for small data sets (via OwnCloud).

As described below, the proposed CI Engineer (CIE) will support a wide variety of tasks, many of which are not generally thought of as being the responsibility of an Engineer. However, the responsibility of an Engineer is to all disciplines, not just one. This will undoubtedly make finding such a person more difficult, but certainly worth the search. We do not expect to find a single person embodying the expertise required for all these projects, especially since we will be competing against Google, Facebook, EMC, and the like for such talent. However our local support team already has significant expertise in these areas, and we will search for a CIE who has expertise which is compatible and complementary to our own. Each of these projects will be discussed and supported by the Research Computing Support group [1], as well as other local and remote experts in the domain on whom we can call for advice and critique.

Note that we include a number of hyperlinks in this document. These are meant as courtesy references since often the review panel is drawn from groups as heterogeneous as a university faculty and some terms in this document are admittedly obscure. As such, the links are meant only as aids in understanding the flow of the proposal, not as technical references. There are no peer-reviewed citations in this proposal, only documents relating to specific technical aspects. Where there are noted References with square brackets [#] that include URLs, they point to supplementary material showing specific experience and documentation relevant to this proposal.

Following are the projects with which we will task the CI Engineer to work upon.

**1: Outreach and Consulting with Researchers**

There is considerable expertise in our Research Computing Support (RCS) group, which makes a real difference in how researchers deal with data when they realize it is available. The CyberInfrastructure Engineer (CIE) will endeavor to meet with school and department heads across campus to make them aware of this expertise. If the CIE does not have the expertise to address any particular topic, they can call other RCS staff to assist in the areas of data storage and flow, application-level reconfiguration, algorithms, databases, and specific applications.

The CI Engineer will edit and publish a monthly newsletter that provides updates on available campus CI resources.

The HPC Cluster is one of the main resources where we interact with and respond to researcher about their analyses [2].

Partly in response to the availability and local configurability of the HPC cluster, we have had at least 3 graduate classes taught using cluster resources; the students use their own laptops or lab PCs and log into the HPC cluster to run simulations and do symbolic math using the OSS SAGE system. These classes have used all 3 main user interfaces - the web interface, the command line interface, and the X11 graphical interface via the free X11 compression utility x2go (like VNC or NoMachine clients). This serves as an opportunity to promote cluster computing as a teaching platform as well as a research platform. This is also one of the advantages of our 10Gbps LightPath [3] network (also below and our CI Plan) as it enables the multiple high bandwidth, low-
latency connections necessary for fluid GUIs in teaching. The CIE would also be responsible for reaching out to instructors to make them aware that this extremely cost-effective platform exists and can be integrated into their curriculum.

In addition to spreading the word and facilitating the use of best practices with UCI researchers, the CIE would also reach out to peer institutions to coordinate conferences (virtual and physical) to establish relationships, foster information sharing, and even hardware lending to improve overall speed of research. We currently have one XSEDE Campus Champion at UCI, but we would expect the CIE to rise to the point of being a second one.

While not the primary role of the CIE, they would also engage companies to enable UCI in testing and evaluating technologies to stay ahead of the technology curve. Interacting with vendors and other researchers about best practices and approaches for data analysis, integrity, tracking, provenance would provide good feedback and 'reality checks' for our own processes.

**Implementation**

The CIE will meet with department chairs to alert them to the resources on campus, as well as to encourage them to make this known to their faculty. Further, we will seek the support of the chairs in communicating with faculty to make them aware of available resources and encourage them to sign up for monthly updates. The CIE will create and distribute a monthly email to the faculty list about recent changes in CI and upcoming events that might be of interest.

**Milestones**

The CIE will contact all chairs within several months of their hire to initiate the process described and will continue to reach out until all department chairs have been contacted. After that period, the CIE will send out an email each month about CI changes and events. They should also be posting to the relevant listservs: XSEDE, UC-Research-Computing, as well as other forums that are identified.

**2: LightPath / Science DMZ**

UCI was recently awarded a NSF CIE grant to implement a 10 Gigabits per second (Gbs) Science DMZ for large data consumers and producers [3].

We have purchased, installed, and configured the major equipment, and large data sources and consumers have started to use it, but we have yet to get as much use as we would like due to:

- lack of user education and awareness
- complexity of network configuration
- protocol and port filters blocking routine tasks (no CIFS/SMB protocols allowed through the firewall preventing 'normal' desktop fileserver access; port blocking on the main firewall masks software licensing requests)
- some user-side security concerns due to concern with lack of a firewall
- few individual systems can supply or digest such very high data rates.
- few compelling interactive applications that require or could exploit such data rates.

UCI's core network is behind a firewall, unlike the LightPath DMZ (see below)

*Figure 1. 10Gbs LightPath Science DMZ relative to UCINet*

The diagram below shows how the LightPath DMZ connects to the Internet as well as its relationship to the core UCINet. Note that traffic going to into UCINet must pass a border firewall, whereas the traffic to LightPath does not; it is protected only by an Access Control List at the Core LightPath Core router (in green in diagram). Diagram courtesy Jessica Yu of
The issues described above are limiting the use of a valuable resource. The proposed CI Engineer would in harmony with UCI Network Engineers to help in debugging problems and providing documentation on how to do various tasks inside the DMZ.

Examples

- identifying and configuring a high-bandwidth requiring application that uses a Graphical User Interface, such as VISIT, or one of the 3D image reconstruction applications such as freesurfer.
- Connecting, configuring, and teaching how to use a GridFTP client.
- Showing users how to use Globus Connect over the LightPath network.
- Setting up high-speed Network File System (NFS) mounts to remote sites.

Implementation

The CIE would assemble and prepare documents describing what our LightPath DMZ is and under what conditions it will be useful to end-users, both using it as part of other resources such as our HPC cluster (which uses it as an high-speed link to external data sources), and as an end-user link if the user has a requirement for high-speed data or low-latency interactions.

If faculty desire further information and/or connection to the LightPath network, the CIE would assist in their determining wiring and hardware requirements in harmony with UCI’s Networking group. Once the hardware was in place, and if documentation was not sufficient to describe how to configure the connection, the CIE
would assist in the configuration and modify the documentation to address the problem.

Milestones
The CIE would create an initial document on effective LightPath use within 2 months of hire, and then expand on it in wiki or blog form as more feedback was received or the usage increased, to address each of the points that are mentioned above, with hyperlinks to relevant other institutional documents.

3: Campus Storage Pool

As indicated in our CI plan, UCI recognizes that a key resource for both research faculty is reliable, scalable storage. Previous attempts with such technologies have not been very successful and current commercial products, while viable, tend to be quite expensive when scaling up and can limit future choices and flexibility due to vendor lock-in. There are Open Source, Proprietary, and Hybrid systems available to support flexible storage pools and facilitating their availability and use would be a priority for the CIE. The CIE would examine and test the options available and provide a written evaluation of such technologies that would be shared with other academic institutions.

We have dedicated hardware towards storage evaluation, with a test cluster composed of several older compute and storage nodes with 1 Gbs Ethernet and Double Data Rate (DDR) Infiniband to accurately replicate the working environment. However, we lack the time necessary to exploit this resource for extended testing.

The campus storage pool will be some form of Distributed File Systems (DFSs), as we have experience in this for HPC (Gluster & BeeGFS, and have documented these experiences [4]. We have to dedicate someone to evaluate a few more before we can responsibly make a decision that will be very long-lived. The DFS that we are now using for our HPC systems (BeeGFS) is quite fast but does not support the kind of failover and reliability that we would need for a storage pool that will be supporting a much wider storage role. The OSS file systems that we would like to test exhaustively are Lustre, DRBD, and possibly Ceph, as well as more testing of these partially (BeeGFS) or completely proprietary (GPFS) filesystems. These testing results will be openly published as we have in the past with our Storage Brick testing [5] and Gluster vs Fraunhofer testing [4]. This is especially important, since there have been very few real-world comparisons of such systems.

Implementation
Once the CIE has been initiated in the dependencies of our existing infrastructure, they will use our existing testbed cluster to test some provisioning technologies and then use the storage servers to test most of the technologies mentioned above. Each will be vetted for various performance metrics (under various loads, with tiny files, streaming, high IOPS, etc), robustness under various failure scenarios, interaction with multiple differing client loads, ease of backups, etc.

Milestones
This is a non-trivial task and one that will probably take at least a year to fulfill with the CIE's other obligations, even with our existing expertise and assistance from others on the team. However, we expect that the CIE can run through the provisioning tasks and start producing documents about testing the above filesystems within 3 months of hire.

4: Hybrid Backup System

Like all institutions, we need robust backup systems to support both our multiple research storage pools and the above-described Campus Storage Pool. 'Single' copies of data are soon 'missing' copies of data. There are
certainly commercial systems that can back up storage, but they are all expensive and lead to vendor lock-in, as we and others have discovered over the past few years.

We want to design a hybrid tiering backup system that takes advantage of the university's data dynamics to cache recently modified data to a local disk-based system and then forward more stable data to long-term, cheaper storage for archiving. We would like to design the back end for both disk-based and tape-based systems so we can eventually use any pools that are open to us, including Amazon Glacier, UCLA's Cloud Archival Storage Service (CASS) [6], and University of Oklahoma's PetaStore [7]. However, the effort to design, and implement this system is slower than desired due to lack of human resources.

Implementation
This project is at a very early stage. Besides the use of generic Storage Bricks using ZFS that we would use for the local backup unit, the logic has yet to be written, so the CIE would have fairly wide scope for design choices. Since none of the control logic needs to be especially fast, the logic would probably be an interpreted language like Python or Perl with hooks into bash scripts and the relational database (from the Robinhood Policy Engine that we're already using) for tracking file provenance. The logic for querying the Robinhood database is fairly easy, and the initial file movement to the local backup unit is trivial, but the logic for the remote transfer may be difficult, especially the business logic, since there are multiple , to connect to a commercial provider. We already have written a parallel rsync application called parsyncfp that can saturate the 10Gbs LightPath network, so pushing the data that fast is not a problem, but deciding which data to push is.

Milestones
Like the Campus Storage Pool described above, this is a non-trivial task. The CIE will have to come up to speed on a number of CI issues, but since many of them are overlapping (LightPath, the HPC cluster, Distributed File Systems, and the Amazon Elastic Compute Cloud (aka EC2) Application Programming Interface (API)), the learning curve should be fairly fast. We should have a simple proof of concept working within 6 months and a more fully debugged version at the 1 year mark.

5: Dataset Movement, Sync, & Sharing
Data movement is one of the main aggravations in the Big'n'Bigger Data age. All researchers frequently run into bottlenecks in moving data to/from endpoints around the world. We have been inspired to write up a fairly well-received document that describes some of the problems and solutions in this domain.

Such bulk data movement is just one of a few related issues. Others are the problem of sharing and syncing data with collaborators and devices. The technology to share a few megabytes back and forth is now out of date and commodity tools such as Google or DropBox are not reasonable alternatives for multi-TB data sets. We have started to investigate both commercial and OSS solutions as described here [8].

In order to avoid steep transfer costs charged by commercial services such as Aspera, Signiant, and now even the previously free Globus, research communities have to set up local GridFTP nodes or alternative utilities and educate their users how to use them. This is a non-trivial task, since so many researchers are dependent on Graphical User Interfaces (GUIs) and simple authentication mechanisms, so the idea that you have to use a MyProxy credentialing service is quite foreign, especially via a command line client.

Implementation
Setting up a solid GridFTP site is non-trivial, and this would be one of the CIE's first tasks, since UCI is hosting a 200 TB data source for multiple users. They would also be responsible for testing OSS UDP-based file transfer utilities such as tsunami UDP. In any case, UCI must join the ranks of a first class data suppliers as well as being a first class data consumer.
We are already testing the OSS Owncloud file sharing software to the extent of several TB. For groups of tens of users, it seems to fulfill its promise as a private equivalent to Dropbox but we do not have a good idea of how it scales. The CIE would take responsibility for adding test groups to UCI's local instance and monitoring how well it survives heavy use, software upgrades, and especially the edge cases that academics tend to present. In addition, they would share authorship of our Data Transfer documents, to be updated as new utilities and protocols are vetted and decided to be useful.

Milestones
While it's non-trivial, it's also well-documented, so setting up a single-host GridFTP site should be completed within 2 months of hiring the CIE. The hardware and networking for a multi-host site is dependent on local funding and other organizations so a full multi-host setup may be delayed for as much as 6 months. Since there is already a current OwnCloud instance running, there's no setup time for the service, but there will be a short ramp-up while the CIE learns the administration (trivial) and the underlying architecture and performance bottlenecks (non-trivial). The CIE should be fluent in the site administration hours after first encounter with the interface, but the performance details will certainly not be as simple.

6: Teaching Linux and BigData analytical techniques.
One of the key responsibilities of the CIE will be to accelerate local Linux knowledge and use, especially for large datasets. We have seen some Linux-naive students being asked to analyze TB-sized datasets by similarly naive advisors and they have responded by trying to apply the tools they know - MS Word and Excel, with unsurprisingly poor results. Even those who know their way around Linux are still largely locked in data analysis techniques from a generation ago - the 'grep' and 'cut'-mediated slicing and dicing of columns and rows that were the standard for MB-sized data. Data is now so large that much data is being made available only as compressed binary formats (HDF5, netCDF, XDF, gzipped FASTA, FASTQ) and we need to impress upon researchers that this is both a good thing and isn't hard to learn. But without intervention, it's a long, hard path to figuring out how to do this.

In desperation, we have initiated several beginners classes for introducing students to Linux and Cluster Computing, bash, Perl, Python, and R programming languages, some popular aspects of Bioinformatics, and BigData [9]. These are not formal computer science classes, but essentially a 'drivers license' for using our compute cluster to protect ourselves against 'drunk data drivers'. These classes are typically one day long, with approximately equal parts lecture and tutorial, the latter done on our HPC cluster.

The CIE would be partly responsible for managing and teaching these classes to the level of his or her expertise, certainly the 'Introduction to Linux' and the introductory programming classes. They would also be charged with integrating other sources of material into our courses, such as the excellent Software Carpentry series, and the R/BioConductor documentation from Thomas Girke at UC Riverside.

Implementation
Since we already have a basic curriculum, there will be no delay in preparing content, but there will be delay as the CIE is introduced to the class, first as a spectator and then as a co-instructor, and finally as a solo instructor. They will then be expected to expand various aspects of the course in both lecture and tutorial aspects, coordinating with the other classes on R, and BigData.

Milestones
The CIE will attend the first classes given as soon as they are hired, and will be expected to help out as a tutorial assistant at that point. Since the courses are given about every 2 months, they should be ready to co-instruct the class after that and then progressing to solo instruction. Simultaneously, they should be creating content to address missing topics, such as more advanced (but still basic) programming using scripting languages, understanding and using the SGE scheduler, profiling and debugging code.
7: Cluster Cloudbursting

Cloudbursting is the use of a public cloud service to address overloads in private cloud processing. It is not a novel idea, but marshaling the necessary logic and network channels to support and integrate it into our systems is a non-trivial task. We plan to use it to offload highly compute-bound tasks running on our HPC cluster, if the user is amenable to paying real dollars for the acceleration. In many situations, a group has not spent the ~$10,000 to purchase a large compute node for the cluster, but still needs jobs to run at an accelerated pace. Even though the HPC Cluster scheduler is highly efficient and we use checkpointing and cycle scavenging to provide as much efficiency as possible, it is becoming a victim of its own popularity.

Using our cluster scheduler, this would not be possible, but if the PI was willing to set up an Amazon EC2 account, the CIE could bypass the priority settings on our cluster and run much faster on an Amazon virtual cluster. Our part would be to provide the logic so that this would be transparent to the user.

Using the Distributed Resource Management Application API (DRMAA) for our local cluster resource manager (which is Son of Grid Engine (SGE), an OSS Sun Grid Engine descendant), we expect the CIE to create the programming logic to offload some of our current local compute load to remote resources like Amazon's EC2 for those jobs that are primarily compute-bound. Jobs that are disk-bound would stay local where our DFS is optimized to handle that load, but code for compute-intensive jobs such as complex simulations, iterative sampling, and the like could be transferred to a remote system for execution. As long as the output was reasonably sized (a key criteria, since many service providers charge considerably for data egress), the output could be transparently and economically returned to our cluster storage. This would require SGE setup and integration with Amazon's EC2 as well as Google's AppEngine. Other people here have very deep experience with SGE so that part of it is mostly covered. Digging through the different APIs and especially figuring out the charging models will be much more complex.

Implementation

Since we use Son of Grid Engine (SGE), which has built-in support for the DRMAA, we have to 'spin up' an SGE cluster instance in whatever cloud we want to burst into and enable the appropriate DRMAA channels to accept our request. Like many such technologies, the actual technology isn't the problem so much as the administration surrounding the technology. In this case, getting and setting the billing information for the user, and then making sure all the notifications are made before a simple bash mistake causes thousands of dollars in billing is non-trivial. It may even be non-viable if the downsides and complexity exceed the upsides. Nevertheless, the potential upsides could be quite high, since this would allow significant expansion of our cluster's apparent power with a reduction in Data Center space and energy use.

Milestones

This is a secondary task, so it would probably not be started until 6 months after the hire, until the CIE has learned a significant amount about all the dependent technologies and our implementation of them. Like many such projects, this would require considerable research, prototyping and trial and error to work out how the API calls actually translate to actions. We expect that a proof of concept or a document detailing that is is not worth pursuing is about 1 year away from hire.

Sustainability of Campus Cyberinfrastructure Engineer Position

While this proposal only requests funding for a two year period, it is expected that the position will become a permanent staff member within the Office of Information Technology afterwards. Since the PI is also the campus CIO and OIT department head, he has the authority and operational funding resources to continue funding the position assuming the project succeeds to the point where the CIE is making substantive
improvements to the state of UCI Research CyberInfrastructure. The CIO has been interacting with campus leadership and faculty to ratchet up planning mechanisms to address future campus RCI needs and implement appropriate strategies. We are working in parallel with a broader effort within the University of California that will leverage a system-wide research computing conference in late March to help build a 5-year UC RCI strategy. Establishing the CIE position is clearly a step in the direction that the campus needs to head.

References

[1] The Research Computing Support group consists of 3.5 FTEs and one exceptional student. 

*Harry Mangalam, PhD* (UCSD), one of the coPIs of this grant, and whose Biosketch is included in the proposal.

*Joseph Farran*, an experienced systems administrator and programmer whose expertise with cluster computing in all its facets has contributed to making the HPC cluster the largest and fastest growing research compute facility at UCI, and with 900 post-graduate users from all Schools, one of the most widely used research facilities at UCI, period. He single-handedly integrated the Berkeley Lab Checkpoint/Restart checkpointing into our scheduler system and enhanced it with the CPU-scavenging system that allows all users to effectively gain more resources than they paid for.

*Tony Soeller*, an expert on Geographical Information Systems, whose talents have advanced a number of UCI research programs that depend on geolocation and integration with geographic data sources. He has also been key to fostering interactions with the *Faculty Research Computing and Networking Advisory Group* mentioned in our CI plan that assures faculty oversight of CI resources.

*Garr Updegraff*, PhD (UCSD), a ½ time programmer and extraordinary database expert, who wrote much of UCI's Registrar's system and who we are lucky to have on staff due to his interest in research problems and algorithms. He was going to retire until we were able to retain him by offering him this ½ time position which was much more engaging and challenging.

*Edward Xia*, an undergrad Information & Computer Science student assistant who has such a tireless work ethic and cutting curiosity about all things digital that he is teaching us as much about modern web programming and data techniques as we are teaching him about cluster and high performance computing.

[2] The physical and software resources of the HPC Cluster are summarized here. Almost 40 individual PIs have contributed compute and storage hardware to the cluster, in addition to the Schools of Biology and Medicine for priority use by their faculty and by OIT itself for general use. We extract and analyze the scheduler logs to record the usage of the PI's research group, as well as by School. (The 2 previous links are to external copies, since these records are not generally available to non-UCI viewers.)

[3] LightPath is our name for the Science DMZ created with funding from a previous NSF grant (NSF Proposal # 1341038). An internal presentation describing the project can be viewed here.

[4] Distributed File Systems (DFS) are fairly difficult to both set up and test in realistic conditions. We took the opportunity with incoming hardware to explicitly test our existing Gluster filesystem against another DFS, Fraunhofer (now renamed to BeeGFS). The results of these tests, mostly on identical hardware, is documented here.
In researching cheaper storage for our research cluster, we documented the creation of the Storage Brick, a chunk of storage that could be used as the basis for a variety of storage technologies, including testing hardware array controllers, filesystems, and tuning parameters. The report is fairly old, but it demonstrates some of the details that we undertake when we document a technology.

Our interaction with UCLA's IDRE group, especially in the investigation and use of cross-campus resources such as their Cloud Archival Storage Service, is an example of how we wish to make the most efficient use of resources. Our installation of the LightPath Science DMS (see [3] above), shows how one technology opens up more efficient channels for collaboration and analysis in others.

Like our use of the CASS system (see [6] above), we intend to collaborate with and use the University of Oklahoma's PetaStore facility to act as an archiving system for the data that has timed out of our active research data pool. This hybrid, distributed backup system is one of our proposed projects for this proposal.

Due to increasing sizes of datasets and the inability of commercial entities to provide sync/sharing services for them, we have started to investigate how to provide these services to our researchers in ways that are both convenient and scalable. See this document for details.

We have been offering introductory classes on Linux on the HPC Cluster, and Bioinformatics for over a year and have also started teaching some Linux and BigData handling courses in association with Padhraic Smyth's Data Science Initiative, out of the Information and Computer Science School. The classes have both a lecture as well as a Tutorial section. As well as referencing useful material from the Internet, we have also developed our own material. Some of these are listed below:

- A Linux Tutorial for the HPC Cluster-based
- Hands-on session with Basic RNA-Seq data analysis
- BigData and its Analysis on Linux
- Manipulating Data on Linux
- How to transfer large amounts of data via network.