Green Data Center Computing:
A Demonstration Project
NYSERDA Agreement # 22899

Period of This Report: Apr 28 – July 15 2012

Introduction

The project's aim is to demonstrate the feasibility of deploying a network of Performance Optimized Datacenters (PODs), geographically distributed to exploit the availability of renewable energy for their operation. Such a distributed system has the potential to significantly enhance the energy efficiency, reliability, security, and overall performance of data centers by several means, including optimizing the utilization of the available renewable power for computing by intelligently redistributing computational load depending on the availability of renewable energy and minimizing losses associated with power transmission by placing the PODs near the power source. This concept provides data center operators the means to avoid performing expensive utility upgrades as the availability of wind power through New York State and other states grow, keeping the infrastructure and Transmission & Distribution (T&D) costs low, thus making it possible to use the wind power that is currently stranded, i.e. not-delivered to the grid due to the T&D constraints.

Short Summary

This report presents details of the project work performed to date:

- Task 1 deals with the PODs and the Operation Management Unit design and assembly, along with the development of the Energy Management Unit and Power Supply. This task is in progress and during this reporting period the full system design was performed and several components for the in-house laboratory demonstration were acquired. The renewable energy emulator architecture was modified consistently to the changes and updates made in the complete POD architecture.
- As part of Task 2, the characterization of the subsystems and the assembly to determine the correlation between Input Power (to POD) to Wind Variability is currently under investigation. An algorithm implemented in a numerical environment was developed to model the power commitment scheduling. The particular scenario included in this report is analogous to transfer of workload between two PODs to minimize the overall system operation cost. Accordingly, the algorithm was used to demonstrate power usage and net balance between demand (workload) and power availability, either from renewables or from grid.
- Work on Task 3 dealing with the development of a system level model using experimental data to be used in design optimization will start in October 2012.
- Task 4 aims to characterize the type of workloads suitable for the POD architecture. A detailed review is underway to study existing virtual machine migration technologies and other tools and strategies for managing downtime at one POD datacenter and/or shifting work between PODs as to take advantage of available renewable power. A number of other tools and strategies for making POD hosting a viable alternative for a larger range of workloads are also under consideration.

A portion of the hardware required for the laboratory demonstration was acquired, a list of which is provided in the next section. A summary with expenditures is included at the end of the report along with a schedule showing a percentage completed and projected percentage of completion.

a. Progress of Project to-Date

TASK 1 - Progress to Date

From an electric and control standpoint, this project was divided into seven major tasks. Selecting the subsystem architecture and topology capable of providing high quality electricity to servers that are co-located is the first major undertaking. Different strategies for monitoring the available power from different intermittent sources of power will also be considered. Engineering design and integration aspects of the project necessary for a renewable-powered POD should be finalized by the end of the summer. This includes; (1) developing the hardware and algorithms necessary to directly power the POD from a renewable energy source (wind turbine), (2) the ability to migrate computational load from one location to another, (3) study of the wind characteristics of the selected location and the inertial response of the turbine to variation in the instantaneous wind power profile, (4) determination of the expected short- and long-term power variations (based on wind characteristics and availability, and specifications of the UC), and (5) the identification, selection and sizing of all the subsystems required for a laboratory and field demonstration. In particular, Subtask 1.1 – Assemble the PODs and the Operation Management Unit, and Subtask 1.2 – Development of the Energy Management Unit and Power Supply, are both in progress. In support of Task 1, a full system design has been performed and several components for the in-house laboratory demonstration have been acquired. The inventory of the currently purchased equipment is listed in Table 1.

I. Hardware Acquisition:

Company	Product	Quantity	Total
Chroma	DC Power Source	1	$4,440.00
Iomega	Storage Unit 12TB	1	$3,283.20
HP	DL165 G7 Opteron6220	2	$7,547.60
	DL385 G7 Opteron6220	1	$5,264.98
NetGear	Switch	2	$164.74
Unipower	Sabre 3KVA (-48Vdc to 208Vac)	1	$3,141.00
APC	SmartUPS SUA3000RMT2U	1	$1,502.39
Grand Total			**$25,343.91**

Table 1 – Inventory of purchased equipment

II. AC and AC Architectures Updates:

1. Comparison between HP based AC and DC architectures:

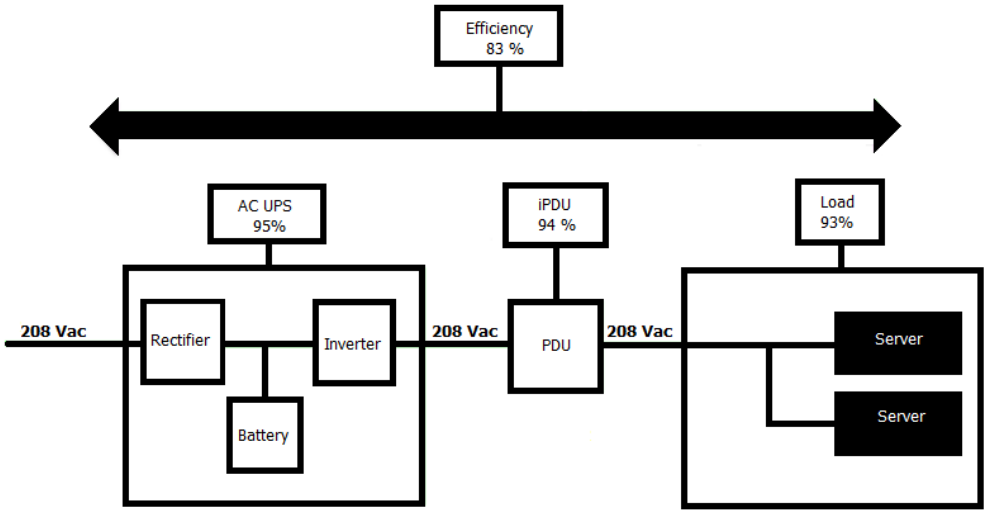


Figure 1a - Updated AC Architecture

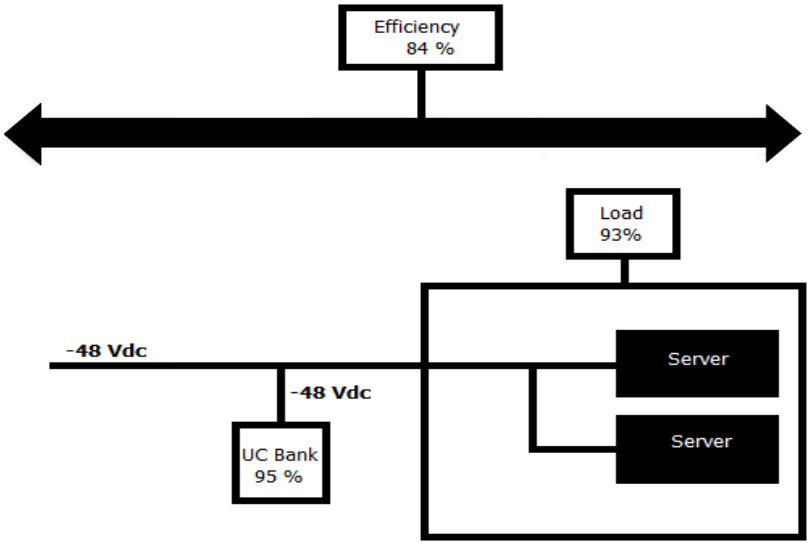


Figure 1b - Updated DC Architecture

Power efficiency estimation of the proposed architectures favors the use of DC servers (84%) instead of AC servers (75%). Hence, the use of DC server is the best solution to increase the energy efficiency of the system because of the reduction of losses due to conversions along the line. Also, DC servers have their advantages. For instance, voltage control is made easier. In addition, backup storage can directly be connected to the PDU without any conversion ensuring faster response. On the other hand, AC servers will also be studied to confirm our theoretical efficiency estimation.

Configuration	48 Vdc	208 Vac
Efficiency	84%	75%

Table 1 – Efficiency of DC Servers vs. AC Servers

N.B: The chosen inverter is 90% efficient. Hence the total AC efficiency is (83%*90%=74.74%). This result is the one used in the table above.

2. Renewable Energy Emulator Design Updates

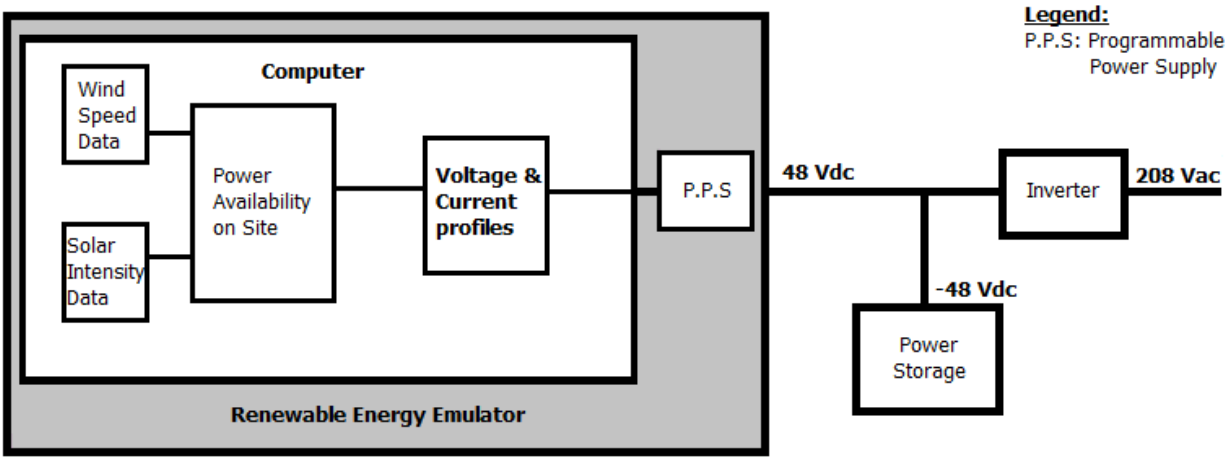


Figure 2 - Renewable Energy Emulator Architecture

Some hand-on experience with renewable power will be achieved by installing two solar panels and two wind turbines on the roof of one of the Clarkson University academics buildings. The complete system will include two wind turbines, two solar panels, two charge controllers, 40 batteries, and one inverter. Meteorological information will be also collected as to study the wind characteristic in our own region. A design of the foundation for the two small 900W wind turbines is in progress. A temporary solar panel stand was also built by an undergraduate student. One of the two 100W solar panel has been positioned on the Center for Advanced Material Processing (CAMP) roof at 40 degrees angle for the summer, and the panel can be easily turned for a 50 degree angle during the winter. Basic system architecture has been designed. The same charge controller can accept PV, wind power, and battery storage inputs. Currently available batteries are lead sealed acid, 12 volts (80 batteries available at our disposal). The solar panel is connected to a charge controller, batteries, and an inverter when in operation. Initial tests after setting up a solar panel with four batteries in series, provided 38.4 volts, no load applied.

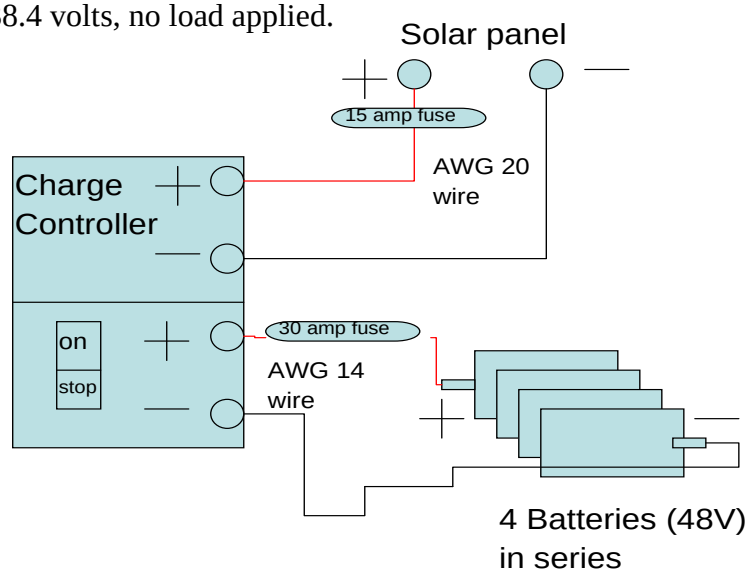


Figure 3 – Solar PV, Charge Controller and Batteries Setup.

III. Test Bench Organization:

Two electric panels will be used, one with 48VDC and the other with 208 VAC. Two experiments will be realized as shown in Figure 4a and Figure 4b.

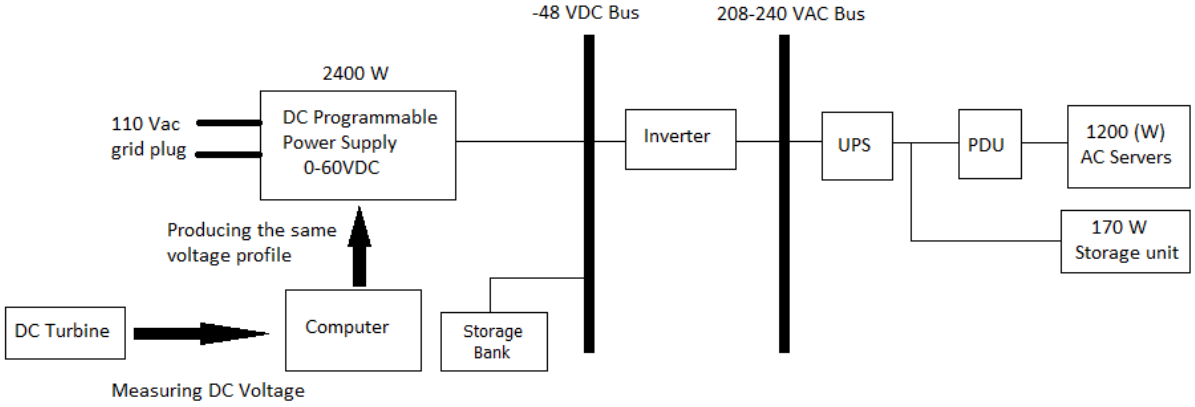


Figure 4a – AC Server Architecture

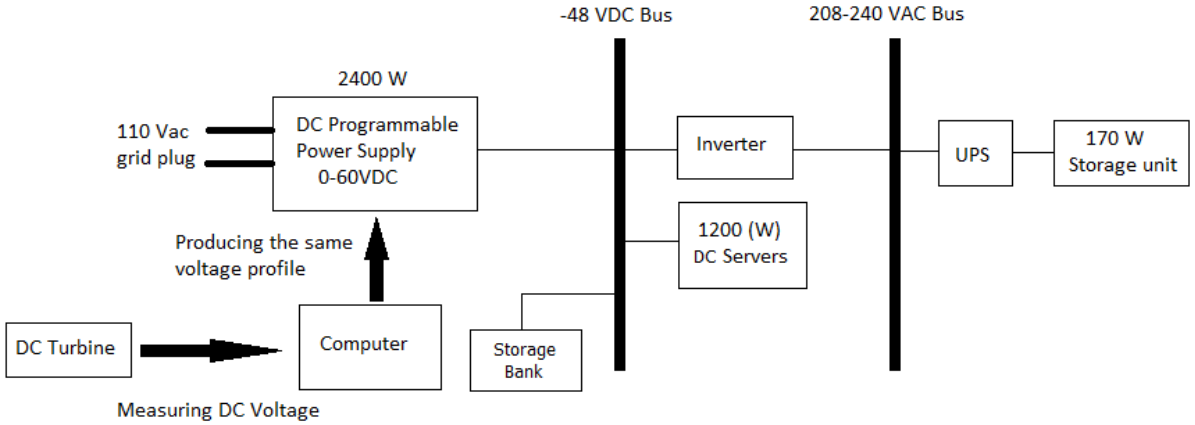


Figure 4b – DC Server Architecture

TASK 2 - Progress to Date

As a part of Task 2 – Characterization of the Subsystems and the Assembly to Determine the Correlation Between Input Power (to POD) to Wind Variability – a code was built to meet the demand and schedule the power commitment. The program gives a schedule of running several electricity units in order to meet a pre-deterministic workload. In electricity market operation, for the same distribution of power plants, electricity price computed based on UC[1] algorithm increases when constraints related to network transmission lines NCUC[2] are taken into consideration. The constraints of additional NCUC price-increases related to outage of transmission lines or outage of generators SCUC[3] is also considered. Therefore,

$$price(UC) \leq price(NCUC) \leq price(SCUC)$$

[1] UC: Unit Commitment
[2] NCUC: Network Constraints UC
[3] SCUC: Security Constraints UC

These algorithms will give the best distribution of power generation and will also give a schedule of units that should be turned on or off in order to achieve the lowest price.

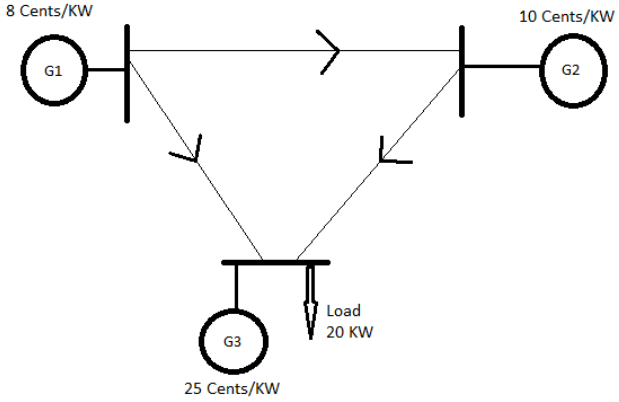


Figure 5 – Example of an Electric Network

For example, in Figure 5, even if the load is connected to the bus G3, it is preferable to drain electricity from the bus G1. Analogously, this particular scenario represents the transfer of workload from one POD to another as to minimize the overall system operation price, hence this particular code was used to discuss power usage and net balance between demand (workload) and power availability, either from renewables or from grid. In this respect, the developed code provides the best planning for a low price of electricity. The UC algorithm will be utilized in order to obtain the best schedule (scenario) with the lowest price. This will be achieved by taking into account expenses due to electricity transmission and security constraints, as well as when these are removed. Instead of performing this expensive electric power shift, the project looks into substituting the power shift by a workload shift.

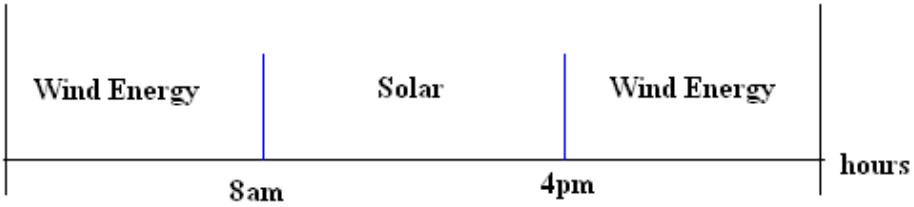


Figure 6 – Renewable Resources Availability

An example on how the algorithm works is provided next. Based on the power availability distribution, the day was divided into several time-slots. For instance, in Figure 6 above the wind energy is available from 12:00 am to 8:00 am and from 4:00 pm to 12:00 am. It is also assumed that solar energy is unavailable either because of nighttime or because there might be an outage. On the other hand, in between the wind energy cycles, from

8:00 am to 4:00 pm, solar energy is available and wind turbines are on outage. The algorithm described below was used to obtain the results for the chosen scenario and to obtain a scheduling plan to be used to manage the available renewable resources.

I. Unit Commitment

1. Unit Commitment Optimization Algorithm

a. Unit Characteristics:

The typical electricity power-cost function is given in Figure 7 below. The trend of the curve is quadratic. Hence, the cost function is given by:

$$Cost = cP^2 + bP + a$$

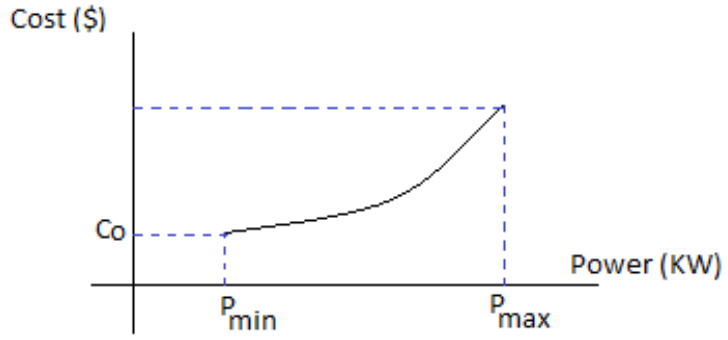


Figure 7 – The Electricity Power-Cost Function

The power unit characteristics which will be used to exemplify this concept are provided in Table 2.

Unit No	Unit Name	Min Capacity (KW)	Max Capacity (KW)	c (¢/KW^2)	b (¢/KW)	a (¢)
1	Wind	3	15	0.002	14.5	8
2	Solar	5	19	0.001	13	10
3	Grid	7	14	0.003	11.5	16

Table 2 – Units characteristics

N.B: The price is given by ¢/KW; it is also set that c=0 ¢/KW^2 for all units.

The optimization of the cost of electricity generation is done by linearizing the quadratic cost function and by using linear programming software such as GPLK or CPLEX, based on Figure 8.

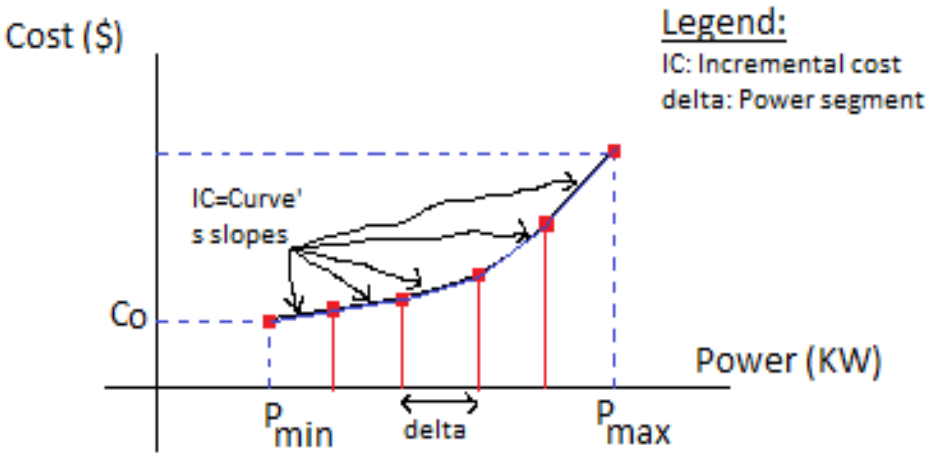


Figure 8 – Linearized Cost Function

$$Cost = Co.I + \sum_{n=1}^{Nseg} IC_n P_n$$

b. Power Requirements:

Table 3 provides for the example in questions the power demand and the power reserve.

Hour (h)	Load=P_t (KW)	SR_t (KW)
1	20.49	7.26
2	18	5.5
3	17.56	5.24
4	17.8	5.3
5	19	6.1
6	11.3	3.54
7	12	4.05
8	13	4.1

Table 3 – Power Demand and Power Reserve

c. Formulation:

The objective function is to minimize the cost of electricity power of all units (NG units) during NT hours. The mathematical model used to describe this system is a Mixed Integer Linear Problem (MILP). The program provides automatically the power generation required to meet the load demand and also units' status binary values that allows the energy workload manager to enable/disable the workload shift toward one datacenter or another. For each scenario we obtain also the price of electricity and the algorithm run time.

$$\min\{\sum_{t=1}^{NT}\left(\sum_{m=1}^{NG}(Co_m I_{mt} + \sum_{n=1}^{Nseg} IC_{nm} P_{nmt})\right)\}$$

Subject to: $\sum_{m=1}^{NG} P_{mt} = P_t$ (Meeting Power Demand)

Other constraints related to outage and hardware limits are listed in Appendix A.

2. Example

a. Scenario 1:

Unit commitment when the solar unit is unavailable, while only wind power and grid power are available. This implies that the availability vector is v = [1 0 1].

Hour (h)	1	2	3	4	5	6	7	8
P_{Wind}	13.49	11	10.56	10.8	12	11.3	5	6
P_{Solar}	0	0	0	0	0	0	0	0
P_{Grid}	7	7	7	7	7	0	7	7

Table 4.a – Power generation distribution

Hour (h)	1	2	3	4	5	6	7	8
I_{Wind}	1	1	1	1	1	1	1	1
I_{Solar}	0	0	0	0	0	0	0	0
I_{Grid}	1	1	1	1	1	0	1	1

Table 4.b – Power Generation Scheduling

Price ($)	23.2040
Time (s)	0.073601

Table 4.c – Electricity Pricing and Code Run Time

b. Scenario 2:

Unit commitment when the wind unit is unavailable, v = [0 1 1].

Hour (h)	1	2	3	4	5	6	7	8
P_{Wind}	0	0	0	0	0	0	0	0
P_{Solar}	13.49	11	10.56	10.8	12	11.3	12	13
P_{Grid}	7	7	7	7	7	0	0	0

Table 5.a – Power generation distribution

Hour (h)	1	2	3	4	5	6	7	8
I_{Wind}	0	0	0	0	0	0	0	0
I_{Solar}	1	1	1	1	1	1	1	1
I_{Grid}	1	1	1	1	1	0	0	0

Table 5.b – Power generation scheduling

Price $	20.8720
Time	0.069876

Table 5.c – Electricity Pricing and Code Run Time

c. Scenario 3:

Unit commitment when the wind and solar sources are both available, while the grid is unavailable, v = [1 1 0].

Hour (h)	1	2	3	4	5	6	7	8
P_{Wind}	3	3	3	3	3	0	0	0
P_{Solar}	17.49	15	14.56	14.8	16	11.3	12	13
P_{Grid}	0	0	0	0	0	0	0	0

Table 6.a – Power Generation Distribution

Hour (h)	1	2	3	4	5	6	7	8
I_{Wind}	1	1	1	1	1	0	0	0
I_{Solar}	1	1	1	1	1	1	1	1
I_{Grid}	0	0	0	0	0	0	0	0

Table 6.b – Power Generation Scheduling

Price $	18.2480
Time	0.05727

Table 6.c – Electricity Pricing and Code Run Time

The scenario for unit commitment when all renewable units are unavailable, while only grid-tie data centers are operational, e.g. v = [0 0 1], was not considered. The energy management unit was forced to operate only based on a combination of energy sources, with the requirement of using green renewables when available, but never grid-tie power alone. Clearly, the workload has to be transferred to another datacenter with similar architecture to be processed.

d. Discussion:

Figure 9 below explains a possible way datacenter can be managed. First, the total data to be process during a time slot is estimated.

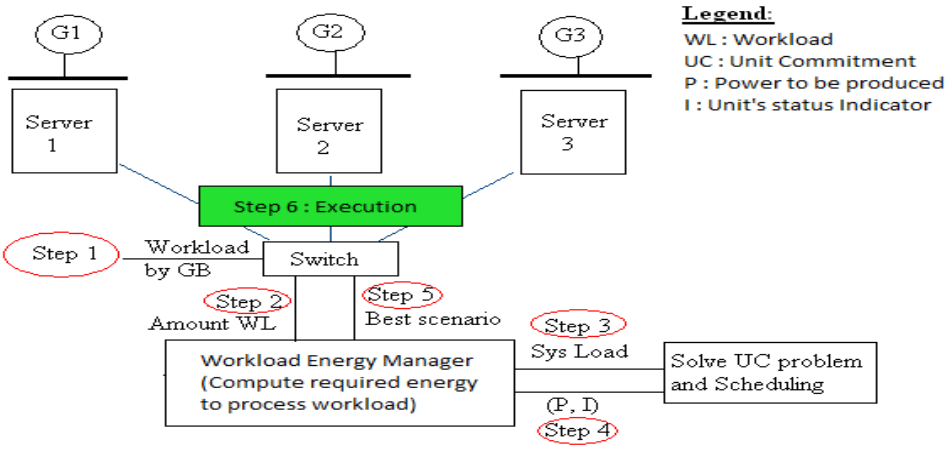


Figure 9 – The synopsis of the workload commitment builder

Second, the Workload Energy Manager (WEM) evaluates the necessary amount of electric energy needed to meet the demand and prepares different scenarios that are sent to a UC solver which prepares a schedule for each scenario. The WEM receives back all solutions and stores them in a database. When one of these scenarios occurs, the WEM orders the switch to move data from one server to another accordingly.

e. Conclusion:

In this phase of Task II a code was built in order to provide a schedule based on unit commitment algorithm. The main advantage of this method is that the electricity price is always kept to a minimum because there electric transportation is avoided with significant cost savings. In addition, unit commitment algorithms are faster than NCUC and SCUC algorithm. Hence, the control system has more time to utilize the best and correct solution. In a case where one unit goes on outage, other alternative scenarios to supply servers with power are available. If in a certain location renewable power is not available, workload shift from one site to another, such that the algorithm can be executed.

The table below provides a comparison between the four scenarios discussed previously.

Outage Units	Solar	Wind	Grid	Wind & Solar
Price ($)	2320.40	2087.20	1824.80	Inhibit
time (s)	0.073601	0.069876	0.05727	

Table 6 – Electricity Pricing for Different Scenarios

TASK 3 - Progress to Date

Work on Task 3 dealing with the development of a system level model using experimental data to be used in design optimization will start in October 2012.

TASK 4 - Progress to Date

The goal of Task 4 is to characterize the workloads for which hosting in a POD datacenter would be a good match. Part of this task includes a detailed study of existing virtual machine migration technologies and an evaluation of the role virtual machine migration could play in shifting workloads from one POD datacenter to another to take advantage of available power. In addition, we would like to explore a number of other tools and strategies for making POD datacenter hosting a viable alternative for a larger range of workloads. We expect that there will be some workloads for which POD datacenter hosting is a natural fit (e.g. hosting of static content) and other workloads for which POD datacenter hosting may never be a good fit (e.g. write intensive workloads with high availability and coherency requirements). In addition to virtual machine migration, we plan to explore other tools and strategies for managing downtime at one POD datacenter and/or shifting work from one POD datacenter to another according to available power.

I. Virtual Machine Migration

The first set of subtasks we have identified involves conducting a series of quantitative experiments of virtual machine migration. Most of the popular server class hypervisor/ virtualization technologies including VMware, Xen, and KVM offer a form of virtual machine migration. However, in many instances, it is assumed that the migration will take place between machines in the same datacenter and possibly even on the same network switch. In many cases, the machine initiating the migration must have access to the same network attached storage device (NAS) as the machine accepting the migrated VM and only the memory state is migrated from one machine to another. Some hypervisors also support live migration with storage migration where the disk state of the VM is transferred as well as the memory state.

Our first task is to perform a thorough evaluation of these migration technologies in a wide range of environments – between two machines in the same rack sharing a NAS, between two machines in the same rack not sharing a NAS, between two machines located in two different labs on campus with and without NAS and finally between an on-campus machine and an off-campus machine with and without NAS.

We are in the process of collecting a variety of interesting measurements of virtual machine migration including the total time to accomplish the migration (from start to finish), the total time the virtual machine is unresponsive during migration (typically during at least the last stage of migration), the total amount of data transferred to accomplish the migration and the total power required on the transmitting side to complete the migration.

We will also collect these same measurements of two extreme configurations – "cold" migration and a high-availability pair. In cold migration, the VM is suspended, the files transferred to other side and the VM resumed. This represents a worst-case scenario for the amount of time a VM will be unresponsive but makes the fewest assumptions about the environment (no NAS required for example). In a high-availability pair, VMs run constantly on both machines so full VM migration is never required. In some cases, data will be shipped from one VM to another to keep the two VMs in sync. The only thing that changes is where requests are sent (to both VMs if available, to just one, etc.).

1. Progress to Date

- Surveyed advertised features/requirements of several major hypervisors and cloud orchestration tools, and added this information into a master spreadsheet

- Using 9 Dell Optiplex 745 PCs from the Applied C.S. Labs for an initial migration testbed, current experiments are limited by existing hardware (e.g. we only have 3 GB of memory so are limited to small memory foot print VMs), but it has allowed us to develop and tune our test suite in anticipation of new hardware
- Set up a similar rack of servers in one of the labs in CAMP which can, in part, be used for testing cross-campus migration
- Worked with VMware to obtain licenses for a wide variety of their enterprise software offerings; Contacted Citrix to request the same.
- Wrote a set of automated data collection utilities for managing migration, collecting timings, observing the migration from the perspective of an external client machine, etc.
- Captured a substantial amount of data on live migration with KVM (both memory migration and storage migration) using existing hardware in our lab and began to collect data with VMware as well
- Identified the remaining hardware required for our final testing; Orders placed.

2. Next Steps

- Continue testing with current testbed until arrival of HP servers.
- Improvements to our test suite (more automation, more automatic correlation of measurements, modifications to investigate anomalies)
- Resolve problems with using CORE to limit network bandwidth at a larger range of network speeds
- Resolve problems with network trace collection at high bandwidth
- Obtain XenServer's license and set up Xen testbed and complete Xen-related quantitative testing.
- Need to obtain Windows Server 2008 copy to deploy VMware Virtual Center.
- Investigate a better network profiling device or tool to quantify the network traffic at high speed.

II. Current Status (More Detailed Status)

1. Description of Test Suite

We have developed a set of tests and automated data collection utilities for quantifying the performance of live migration from a variety of perspectives.

- One aspect of our test suite is the creation of a collection of VMs of various dimensions. We vary the amount of memory configured into the VM, the size of the disk space configured into the VM and the guest OS contained in the VM,
- Another aspect of our test suite is a collection of programs that run inside the VM being migrating to simulate workloads of various kinds. We vary the amount of memory in use within the VM, the amount of that memory that is written vs. simply read, the amount of data read from disk and the amount of data written to disk.
- Another aspect of our test suite is a collection of utilities for quantifying aspects of the VM migration process. Our scripts coordinate launching the programs that need to run inside the VM during migration with launching the actual migration. We also launch a set of network tests including pings to and from the migrating VM to quantify the downtime experienced during migration. We take a set of network

traces to quantify the total amount of data sent over the network during the migration as well as the pattern of the network data transfers.

2. Hypervisors Under Test

We are currently testing the live migration capabilities of KVM3.2.0 and VMware vSphere5.0.

KVM supports two types of live migration:

1. Migrate CPU and Memory but Storage stays on shared datastore (virsh migrate --live)
2. Migrate all CPU, Memory, and Storage. Storage will also be migrated together with CPU and memory. (virsh migrate --live --copy-storage-all)

VMware supports Host Live Migration and Storage Live Migration.

1. Host Live Migration only migrate CPU and Memory, storage has to be on shared datastore(MigrateVM_task)
2. Storage Live Migration only migrate Storage while CPU and Memory stays on the host(RelocateVM_task)

Thus, VMware Live migration requires shared datastore as the media. If VMware wants to migrate all CPU, Memory, and Storage, it has to be split into two separate phase: first host then storage or first storage then host.

3. Live Migration Stress Test Design and Testbed Setup

Generally, activities inside a VM can be classified by CPU, Memory, Disk R/W. We are trying to study how VM activities will affect the live migration, which will help us build a clear picture in estimating migration time when GDC is facing power shortage or outage.

i. Stress Test Design

We designed a series of stress tests for CPU, memory access (read and write) and disk access (read and write) to identify how each of them will affect the migration.

The CPU stress test performs intensive math calculation (FFT) to stress L2 Cache but uses only a trivial amount of memory and disk I/O. It can be configured to use different scheduling priority 1 or 10 which is corresponding to 75% user CPU cycles or 100% system CPU cycles.

The Memory stress test is programmed to allocate a specific memory size and then dirty a specified portion of that total allocation. We can control both the amount of memory that is written or dirtied as well as the rate at which data is dirtied.

The Disk Write stress test is programmed to keep writing data of page unit into a file to a specific size and then rewrite new data from the beginning of the file. After that, it is programmed to write data to the empty file and then rewrite again. It is stressing the disk write and rewrite operation to the extreme.

The Disk Read stress test is programmed to first create a file with specified data, reading data by page unit from file into memory sequentially, and then iteratively read from the beginning of the file again.

ii. Testbed Architecture

In our testbed, the actual migration traffic is isolated from the network we use to control the tests and to monitor the state of the VM being migrated.

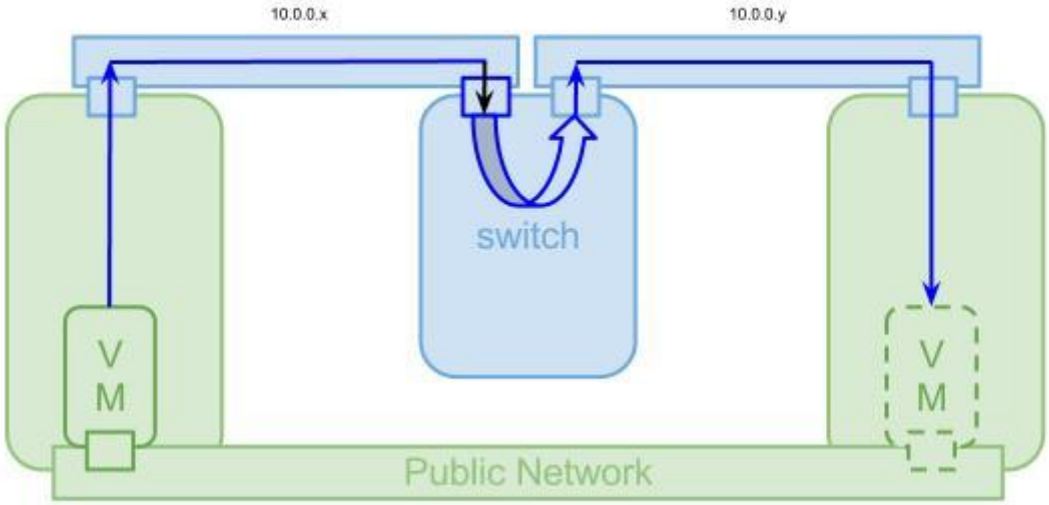


Figure 10 – Testbed architecture

A third machine simulates switch connecting two Hosts on a private network. We can use either an actual physical switch or a third machine running the CORE network emulation software to represent a variety of network conditions.

iii. Test Measurement

To collect live migration statistics we mainly care about how long it takes to accomplish the live migration, how much data will be transferred over the network, and how much downtime will be introduced during the live migration. To measure those statistics, our analysis scripts use the following tools:

IPTraf, a network tool which can count the total transferred data. It can collect the transferred data by each connection. We tried to collect the data from both sender and receiver. It turned out that at low speed 10/100M, it captures the packets correctly but at high speed 1G, some packets may be missed. It needs to be further investigated. Currently we use IPTraf to record the total amount of data transferred during live migration.

Tcpdump is a network analysis tool that can record all the packets traversing a network segment. We have also used this to record the total amount of data transferred and to see the pattern of that data transfer. It has similar problems with high-speed network links.

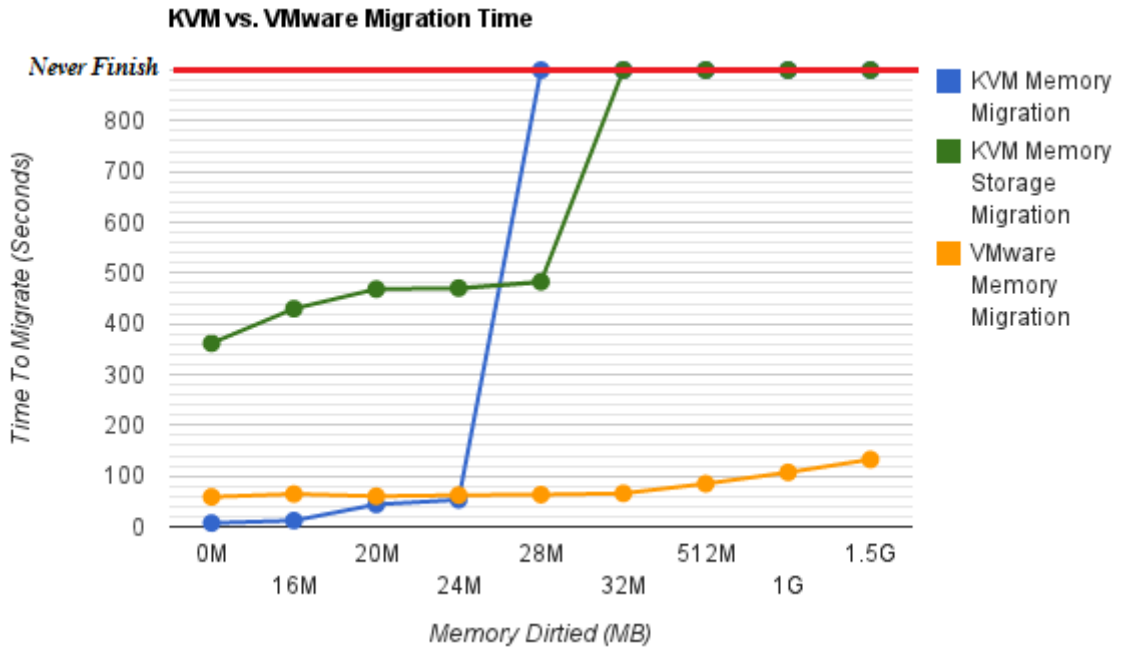
ping is a conventional icmp protocol tool to detect network connection. We control the interval and rate of the pings to pinpoint the downtime experienced during migration. While VM is down and shifted to destination host, VM is not able to respond to the ping request and those request packets will be missed. The VM includes its own timestamp in its ping response. From this we can see how migration impacts the VMs own concept of time.

time is a common utility to measure a program's run time. We use command line instead of GUI to trigger live migration and when the live migration is done, the command will terminate.

iv. Some Initial Results

Our initial testing suggests the following high-level conclusions:

- KVM migration minimizes downtime during migration but in many cases the migration will **never** complete. Contrary to published descriptions of their algorithm, KVM does not identify when data is being dirtied faster than it can be sent over the network. VMware migration, on the other hand, always completes, but typically experiences more substantial downtime during migration. We would recommend that all hypervisors expose this choice tradeoff to the user (i.e. allow users to specify a zero downtime migration vs. a stop and copy migration). We would also recommend that KVM at least detect the unsuccessful migration case and fail gracefully.
- The total amount of memory configured into the VM does not impact migration time in either KVM or VMware. Both hypervisors transfer actual memory in use. On the other hand, when doing storage migration, it is the total amount of disk space configured into the VM that matters and not the amount of that space that is actually being used. Visibility into the file system to only transfer disk blocks containing live data would be a huge improvement.
- Migration time is not sensitive to the amount of CPU activity in the VM being migrated.
- Migration time is highly sensitive to the amount of memory being dirtied during the migration. For KVM, we see that migrations with as little as 64 MB of memory dirtied fail to complete. This must be fixed if KVM live migration is to be used in a production environment.
- For live migration with shared storage, reading data from disk dirties memory as data is brought into the file buffer cache and therefore its impacts on migration time are much like the memory dirty stress test results. Workloads that write to disk are more complicated. Writing data to disk does not dirty memory as much as reading but it does dirty the disk and increase migration time



v. Workload Characterization

The second set of subtasks in this portion of the project relates workload characterization towards the goal of identifying a wide range of techniques that can be used to enable different workloads to run effectively in a distributed POD environment. Virtual machine migration as discussed in the previous section is one very powerful tool, but there are some workloads for which other techniques may be more effective including high-available pairs (both hot-hot pairs and hot-cold pairs) and workload summarization (e.g. caching the most popular results or providing a smaller summary when the full dataset or service is not available).

We plan to characterize workloads according to a variety of characteristics including size of data set required, rate of change in dataset, tolerance for inconsistency, tolerance for downtime, etc. and use these characteristics to suggest a range of techniques to make the distributed POD environment effective. The micro-benchmark results in the previous section are part of this (i.e. categorizing workloads based on the amount of memory read/written, disk read/written, CPU used, etc.) However, we would also like to look at some higher-level workloads and some techniques beyond virtual machine migration.

a. Progress to Date

- Identified a set of workload characteristics to collect/record about important data center workloads
- Identified a set of workload modification techniques that may be applied including VM migration, high availability pairs, workload summarization, predictive downtime and request queuing.

b. Next Steps

- Identify a concrete set of representative workloads that we can use as the targets for our workload characterization and modification techniques (e.g. SpecWeb to represent web server workloads).
- Develop a set of VMs that represent each of these workloads.
- Identify a set of tests and measurements that we can use to evaluate the effectiveness of our workload modification techniques

vi. GDC Demo Architecture

In this subtask, we are approaching setting up a future possible GDC architecture under the renewable power supply condition, with the goal of demonstrating how the possible computing services can survive in GDC at the end of this project. For feasibility of running computing services in GDC, the architecture mainly requires Power Supply Management, Network Management, and Service Scheduling Management to work together, providing a reliable environment for computing service.

a. Progress to Date

- Order the essential equipment required for setting up mini-scaled datacenters to represent the real-world Datacenter service.
- Set up a mini-scaled datacenter supplied with renewable energy as Solar/Wind turbine Powers, running computing services inside the datacenter

- Build the power/network/schedule components to manage the services in the green datacenter
- Demo the strategies of how to schedule and manage the computing services in case of GDC power shortage and outage.

b. Proposed Demo Architecture

With different characteristics of Renewable Energy, we can decompose a datacenter into Computation, Network Flow, and Data access. Computing servers consume most of the power but can be leveraged by Virtualization or other Computer Techniques to shift the services across datacenters while network and storage equipment take less power. This is critical to the users, but needs long-term access so computing servers can be supplied with wind turbines or hydro dams which provide the power at KWs level, while network and storage equipment can be supplied with Solar panels which provides the power at 100Ws.

For computing servers, as the renewable energy is not dynamic and inconstant, Virtualization can be leveraged in both shortage and outage cases and Service Scheduling Components will be in place to manage the computing services:

1. In power shortage cases, Virtualization can dynamically consolidate the services in fewer Servers while shutting down some servers
2. In power outage cases, Virtualization can dynamically migrate the services to another Datacenter to escape the shutdown datacenter

For storage equipment, the low-powered but long-lasting renewable energy can still maintain the data access through network.

So the proposed Demo Architecture will look like the diagram as below:

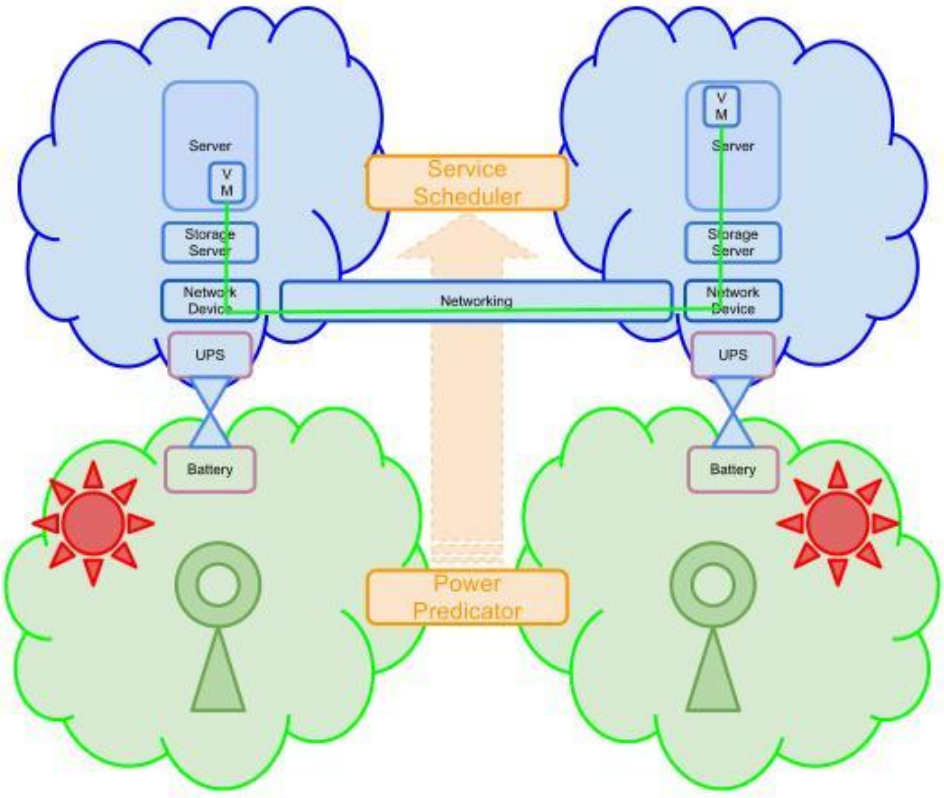


Figure 11 – Proposed Demo Architecture

In the above proposed architecture:

1. Each GDC center has been equipped with Networked Attached storage.
2. Power Predicator (Power management) will keep updates of the green energy stats information to the Service Scheduler. It has two main roles: one is notifying the Service Scheduler in advance when Datacenter's power outage is coming and the other is predicting the power supply strength.
3. Based upon the updates from the Power predicator, the Service Scheduler (Service Management) will make a decision on how to move the services across datacenters and what strategies will pertain for each service in the power shortage and outage.

Based on the proposed architecture, the protocol between Service Scheduler and Power Predicator is focused on how quickly the power predictor can notify the Service Scheduler of the power outage/shortage and how long the power supply can last in current strength. Those two statistics are critical for enabling the Service Scheduler to make a good decision about migrating the computing service.

c. Proposed Hardware Specifications

With the current project budget, we plan the following: each rack-mount server represents a mini-datacenter with virtualization deployed. One mini-datacenter supplied with Green Energy, and the other mini-datacenter supplied with Grid Power. Power Predictor will predict the power of Green-energy-driven datacenter and Service scheduler will decide when to shift the computing service onto and off GDC. This system will include:

1. 1x HP Proliant DL385 with 1200w DC as Green-DC
2. 1x HP Proliant DL165 with 750w AC as Grid-DC
   - Both of them include two AMD Operon 6220 3.G CPUs and 64G Memory.
   - AMD Operon 6220 supports AMD-V Virtualization Technologies and VMware EVC mode.
3. 1x Iomega StorCenter px4-300r Server Class NAS
   - Px4-300r supports both NFS, Samba, and iSCSI
   - Maximum Power Assumption at 170w
   - px4-300r will be equipped with GDC, hosting VM images

d. Current Status

- Finished the order of NAS and got the quote for the HP Proliants
- Obtained VMware vCenter and ESX Academic licenses

vii. Note on Staffing

Along with one PhD student in Mechanical and Aeronautical Engineering (Aitmaatallah) and two undergraduate students working on Task 1 and 2, for the summer, we have had one PhD student (Hu) and two Masters students (Hicks and Zhang) working fulltime on Task 4 of this project. We have also had the input and collaboration of a second Ph.D. student who is interested in the KVM live migration results and two remote students, one of whom will be coming to Clarkson in the fall. The Ph.D. student who was involved in this project during the last academic year (Wilbur) has been replaced.

In the fall, one of the PhD students will continue full time (Hu) and one of the Masters students will transition to be a part-time researcher and a part-time TA (Hicks). We hope to continue partial support of Zhang to participate in this project as well in the fall. In addition, J.

Matthews is planning to incorporate this project into her CS 644 graduate operating systems course in the fall. Hicks and Zhang will take this course as will several other students. Our quantitative study of live migration is a great match for the research project component of CS 644 and it will be a great way to continue the momentum of the summer and to involve more people in the project.

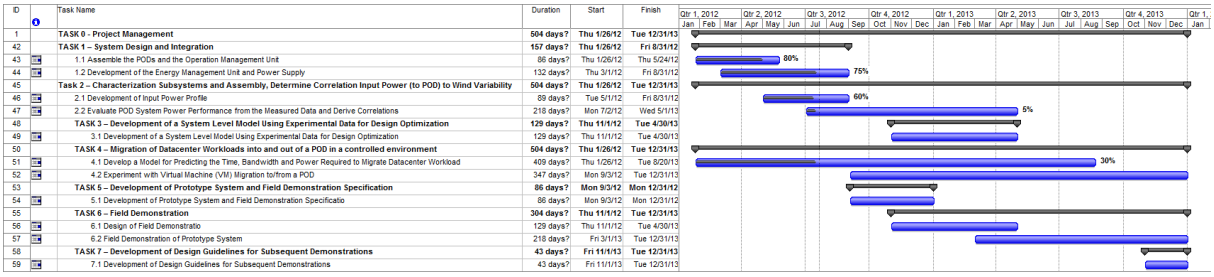
b. Identification of Problems & Planned Solutions.

Tasks 1, 2 and Task 4 are on schedule.

c. Ability to meet schedule, reasons for slippage in schedule, unforeseen obstacles, promising research directions not originally identified in current SOW

No significant slippage or unforeseen obstacles encountered in the project as of now. Work is going according to the plans. A set of HP servers with AMD processors and other equipment has been place on order and will be used as testbed for our initial demonstration. As already indicated we are currently making use of older machines for an initial testbed, but it substantially limits the range of configurations we can test.

d. Schedule - percentage completed and projected percentage of completion



ID	Task Name	Duration	Start	Finish
1	**TASK 0 - Project Management**	**504 days?**	**Thu 1/26/12**	**Tue 12/31/13**
42	**TASK 1 – System Design and Integration**	**157 days?**	**Thu 1/26/12**	**Fri 8/31/12**
43	1.1 Assemble the PODs and the Operation Management Unit	86 days?	Thu 1/26/12	Thu 5/24/12
44	1.2 Development of the Energy Management Unit and Power Supply	132 days?	Thu 3/1/12	Fri 8/31/12
45	**Task 2 – Characterization Subsystems and Assembly, Determine Correlation Input Power (to POD) to Wind Variability**	**504 days?**	**Thu 1/26/12**	**Tue 12/31/13**
46	2.1 Development of Input Power Profile	89 days?	Tue 5/1/12	Fri 8/31/12
47	2.2 Evaluate POD System Power Performance from the Measured Data and Derive Correlations	218 days?	Mon 7/2/12	Wed 5/1/13
48	**TASK 3 – Development of a System Level Model Using Experimental Data for Design Optimization**	**129 days?**	**Thu 11/1/12**	**Tue 4/30/13**
49	3.1 Development of a System Level Model Using Experimental Data for Design Optimization	129 days?	Thu 11/1/12	Tue 4/30/13
50	**TASK 4 – Migration of Datacenter Workloads into and out of a POD in a controlled environment**	**504 days?**	**Thu 1/26/12**	**Tue 12/31/13**
51	4.1 Develop a Model for Predicting the Time, Bandwidth and Power Required to Migrate Datacenter Workload	409 days?	Thu 1/26/12	Tue 8/20/13
52	4.2 Experiment with Virtual Machine (VM) Migration to/from a POD	347 days?	Mon 9/3/12	Tue 12/31/13
53	**TASK 5 – Development of Prototype System and Field Demonstration Specification**	**86 days?**	**Mon 9/3/12**	**Mon 12/31/12**
54	5.1 Development of Prototype System and Field Demonstration Specificatio	86 days?	Mon 9/3/12	Mon 12/31/12
55	**TASK 6 – Field Demonstration**	**304 days?**	**Thu 11/1/12**	**Tue 12/31/13**
56	6.1 Design of Field Demonstratio	129 days?	Thu 11/1/12	Tue 4/30/13
57	6.2 Field Demonstration of Prototype System	218 days?	Fri 3/1/13	Tue 12/31/13
58	**TASK 7 – Development of Design Guidelines for Subsequent Demonstrations**	**43 days?**	**Fri 11/1/13**	**Tue 12/31/13**
59	7.1 Development of Design Guidelines for Subsequent Demonstrations	43 days?	Fri 11/1/13	Tue 12/31/13

e. Dissemination and Meetings

P. Marzocca, K. Janoyan, and A. Achuthan attended the CU CAMP Technical meeting held during 5/16-18 in Albany NY. This trip was supported by CAMP. A project progress status was presented to the meeting participants and to Joe Borowiec (NYSERDA).

P. Marzocca was invited to talk about the GDC project at the IEEE EnergyTech 2012 Conference, May 29-31, 2012, at Case Western Reserve University, Cleveland, OH (USA). He participated to the technical panel "Adopting DC Power for Buildings," http://energytech2012.org. Panelist included: Panelists: Ken Gettman, National Electrical Manufacturers Association (NEMA), DC; Brian Fortenbery, Electric Power Research Institute (EPRI), Palo Alto, CA; Chris Marnay, Lawrence Berkeley National Labs (LBNL), Berkeley, CA; Mike Rowand, Duke Energy, Durham, NC; Keiichi Hirose, NTT Facilities Inc., Japan; Pier Marzocca, Clarkson University, Postdam, NY; John Jahshan, Nextek Power Systems, Detroit, MI; Emmett Romine, DTE Energy, Detroit, MI.

P. Marzocca also presented the GDC concept to C.L. Bloebaum, Director of the Engineering Design and Innovation (EDI) Program at NSF, during the 2012 NSF Engineering Research and Innovation Conference, sponsored by the National Science Foundation's Division of Civil, Mechanical and Manufacturing Innovation (CMMI) and hosted by Northeastern University, on July 9–12 in Boston, Massachusetts. http://www.cmmigranteeconference.org/. There are potential opportunities for the GDC

project with NSF including the System Science program and the Grant Opportunities for Academic Liaison with Industry (GOALI) project.

J. Matthews participated in a "Future of Cloud Computing Think Tank" in San Francisco sponsored by Dell and VMware.

S. Bird was an invited keynote speaker (along with Frank Murray of NYSERDA) at the first annual North Country Clean Energy Conference, June 21-22 in Lake Placid. In his keynote presentation, he discussed the POD concept in terms of its potential synergies with the North Country's energy profile.

S. Bird has begun work on a concept paper to be submitted for publication in early fall. A first draft will be available in late August. NYSERDA will be consulted prior to any publication.

f. Analysis of actual cost incurred in relation to budget

Total cost incurred up to July 15th: $66,267.65 (NYSERDA portion)
Student Salary – $25,411.80 (budget allocation - $89,320.00)
Travel expense – $1,307.80 (budget allocation - $10,000.00)
Equipment - 25,343.91 (budget allocation - $40,000.00)
Indirect cost – $14,204.14 (budget allocation - $88,946.00)
Cost-share – **$45,024.00** (Clarkson)

Student Salary	$11,000.00
Student Tuition	$25,524.00
Summer Students	$5,000.00 (McNair Scholar)
Travel	$3,500.00

Cost-share – **$35,000.00** (Partners)

AMD Corporation	$20,000
HP Corporation	$10,000
WindE Systems	$5,000

Appendix A

Mathematical programming formulation:

$$\min\left\{\sum_{t=1}^{NT}\left(\sum_{m=1}^{NG}\left(Co_m I_{mt} + \sum_{n=1}^{Nseg} IC_{nm} P_{nmt}\right)\right)\right\}$$

Subject to:

$$\sum_{m=1}^{NG} P_{mt} = P_t$$

$$P_{mt} = P_m^{min} I_{mt} + \sum_{n=1}^{Nseg} P_{nmt}$$

$$P_m^{min} I_{mt} \leq P_{mt} \leq P_m^{max} I_{mt}$$

$$|P_{mt}^c - P_{mt}| = 0$$

$$P_m^{min} I_{mt} \leq P_{mt}^c \leq P_m^{max} I_{mt}$$

$$SR_{mt} + P_{mt} \leq P_m^{max} I_{mt}$$

$$SR_t \leq \sum_{m=1}^{NG} SR_{mt}$$

$$0 \leq I_{mt} \leq 1$$

$$0 \leq P_{mt}, P_{mt}^c, SR_{mt} \leq \infty$$

$$0 \leq P_{nmt} \leq \delta_m$$

NT	Number of hours to be scheduled
NG	Number of Generators
$Nseg$	Number of segments
Co_m	Minimum operating cost of unit *m*
IC_{nm}	Incremental Cost of segment *n* of unit *m*
P_{nmt}	Power output of segment n of unit *m at time t*
I_{mt}	Status ON/OFF Indicator of unit *m at time t*
P^{min}_m	Minimum power output of unit *m*
P^{max}_m	Maximum power output of unit *m*
P_{mt}	Total power output of unit *m at time t*
P^c_{mt}	Total power output of unit *m at time t in a contingency case*
P_t	Load Demand at time t
SR_{mt}	Spinning Reserve of unit *m at time t*
SR_t	Requested Spinning Reserve *at time t*
δ_m	Segment interval of power in the power-cost curve linearization of unit m

Table 7 – Variables and Their Designation