Getting HPC into Regional University Curricula with Few Resources

Karl Frinkle - Mike Morris

Getting HPC into Regional University Curricula with Few Resources

(Doing big things with little $$$)

Karl Frinkle - Mike Morris

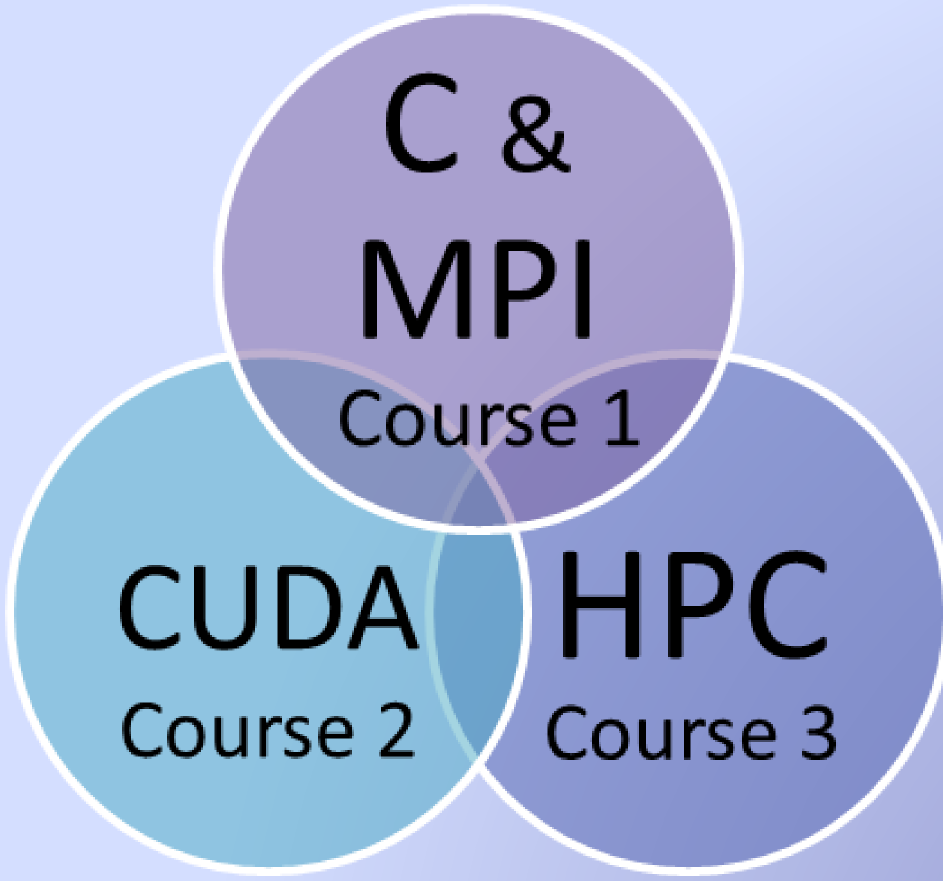
Some of our goals . . .

- Include parallel computing in our curriculum
- Follow ACM accreditation guidelines
- Allow early-CS and non-CS students to participate
- Develop permanent courses
- Exploit limited resources

Our Resources . . .

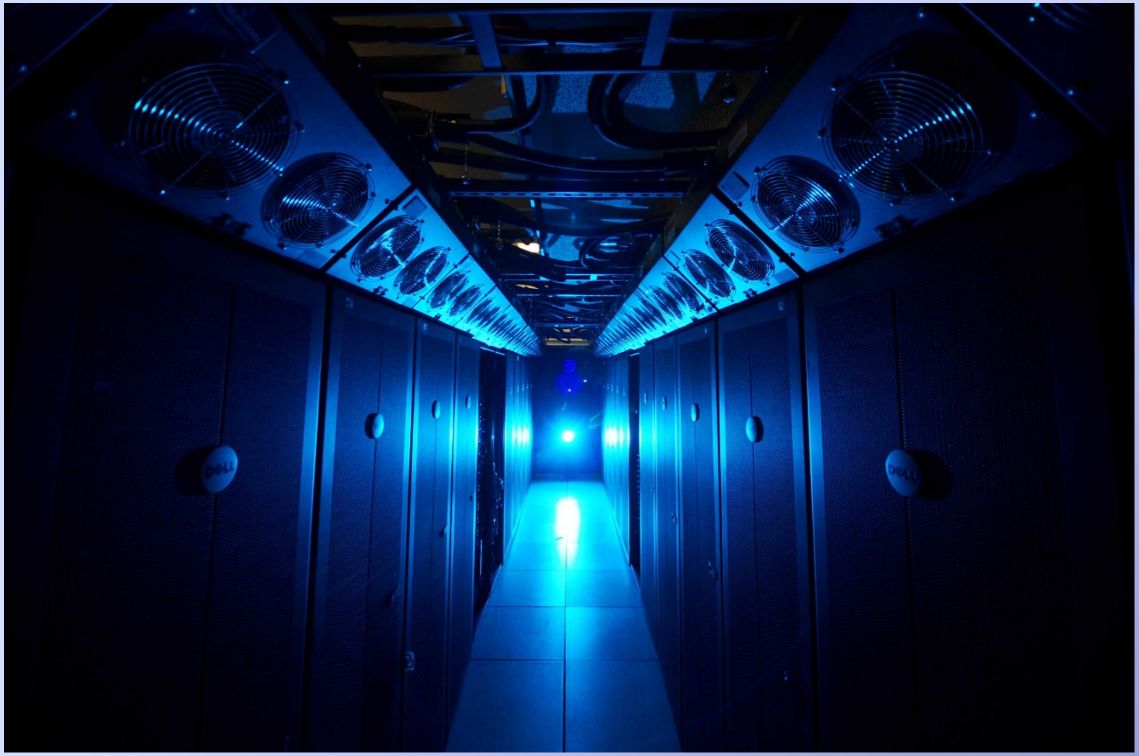
- Uhhhh . . .

3 courses seemed essential . . .



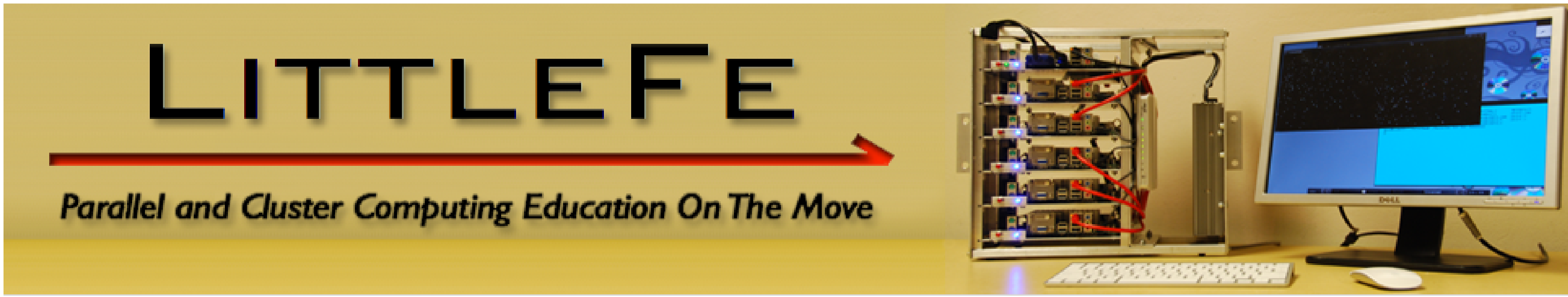
First we needed hardware.

Cluster



http://littlefe.net

Home | About | SC12 Buildout | History | Resources | Curriculum | News | Gallery | On The Road | Events



Welcome to LittleFe.net

SC12 HPC Educators Program Buildout application is available, apply now for a free LittleFe!

Many institutions and teaching environments do not have access to parallel platforms for parallel and distributed computing education. Teaching key concepts such as speedup, efficiency, and load balancing are much more effectively done on a parallel platform. LittleFe is a complete 6 node Beowulf style portable computational cluster which supports shared memory parallelism (OpenMP), distributed memory parallelism (MPI), and GPGPU parallelism (CUDA).

LittleFe weighs less than 50 pounds, easily and safely travels via checked baggage on the airlines, and sets-up in 10 minutes wherever there is a 110/220 VAC outlet and a wall to project an image on. By leveraging the Bootable Cluster CD project, and its associated curriculum modules, LittleFe makes it possible to have a powerful ready-to-run computational science and HPC educational platform for less than $3,000. The parts list and illustrated assembly instructions are available under the "Resources" tab above.

CONNECT WITH US

Like 82

Tweet 6

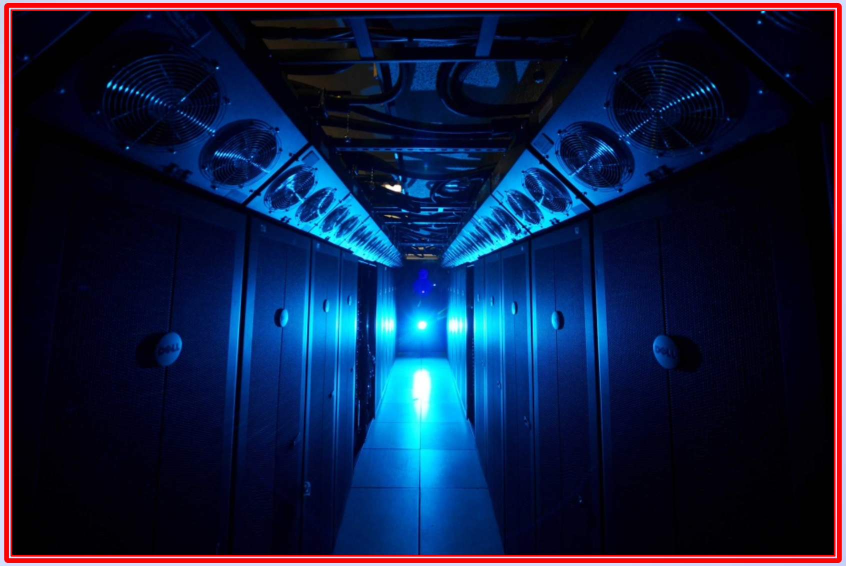
SITE SETTINGS

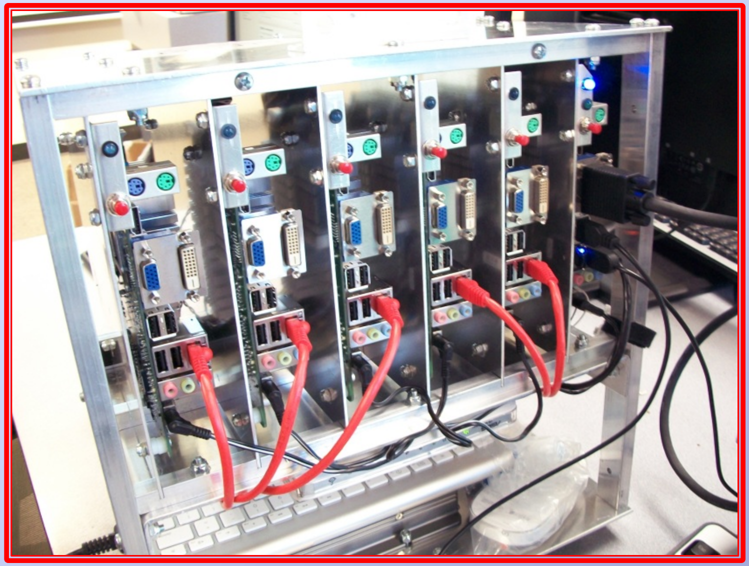
Select Language

© 2012 LittleFe.net - Powered By Drupal

(cc) BY-SA

What is a LittleFe?





BigFe **LittleFe**

Remember, Fe = Iron

Then we needed software.

Some sort of Linux.

We used BCCD,
(Bootable Cluster CD),
an incarnation of Debian.

Getting a course approved . . .

C &
MPI
Course 1

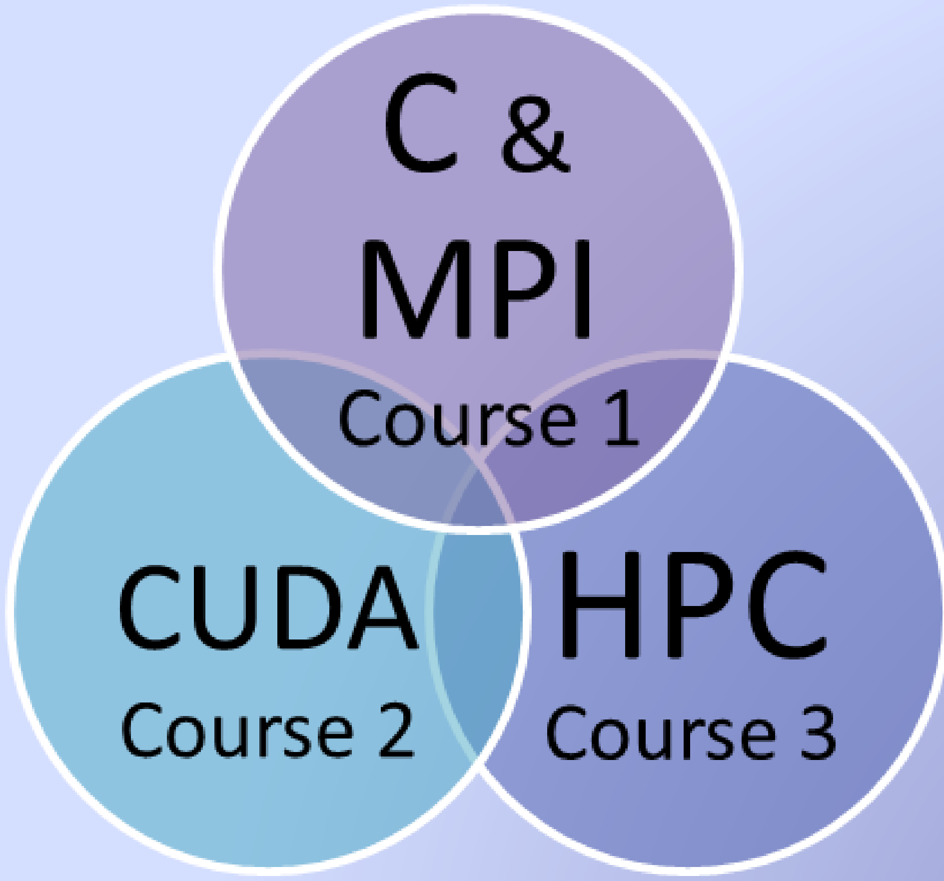
CUDA
Course 2

HPC
Course 3

Special Seminars fit our bill.

- CS4970 – Special Seminar
- CS4980 – Special Studies
- Immediate approval
- Upper level a concern
- . . . etc.

Back to the courses . . .



C with MPI

C &
MPI
Course 1

We threw caution to the winds and assumed a **“Hello World”** was the extent of students’ skills.

C with MPI

C & MPI
Course 1

We threw caution to the winds and assumed a **"Hello World"** was the extent of students' skills.

We delivered a crash course in C, emphasizing **array** structures, **memory** management, **binary** storage and other fundamentals.

C with MPI

C &
MPI
Course 1

We threw caution to the winds and assumed a **"Hello World"** was the extent of students' skills.

We did not emphasize things like **formatted i/o** and loop **syntax** – we gave them examples and let them run with these types of things.

C with MPI

C & MPI
Course 1

As an introduction to parallel programming, we took a parallel **“Hello World”** using **MPI** that Charlie Peck gave us and studied it extensively.

C with MPI

C & MPI
Course 1

The main parallel project topic of the first course was **matrix multiplication**.

C with MPI

C &
MPI
Course 1

The main parallel project topic of the first course was **matrix multiplication**.

Make no mistake – there is plenty of material in this seemingly simple problem for a **plethora** of **programs**.

matrix multiplication

$$\begin{bmatrix} 3 & 4 & 2 & 6 & 7 & 8 \\ 8 & 0 & 9 & 6 & 5 & 4 \\ 6 & 1 & 2 & 3 & 0 & 8 \\ 4 & 8 & 9 & 5 & 7 & 6 \\ 0 & 5 & 4 & 0 & 0 & 9 \\ 8 & 6 & 1 & 1 & 0 & 7 \end{bmatrix} \mathbf{x} \begin{bmatrix} 6 & 7 & 5 & 4 & 3 & 0 \\ 2 & 8 & 0 & 9 & 5 & 4 \\ 2 & 4 & 7 & 6 & 5 & 4 \\ 9 & 0 & 5 & 0 & 0 & 1 \\ 4 & 2 & 8 & 6 & 7 & 5 \\ 5 & 0 & 9 & 1 & 2 & 6 \end{bmatrix} = \begin{bmatrix} - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & * & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix}$$

Each cell of a product matrix is calculated independently of others,

so the problem screams, "Parallelize me!"

C with MPI

C &
MPI
Course 1

C fundamentals and the matrix problems were more than **enough** for a first course.

We were pleasantly surprised at how **naturally** students seemed to **accept** the parallel models.

CUDA
Course 2

CUDA

Compute
Unified
Device
Architecture

CUDA is the computing engine developed by Nvidia for high speed graphics rendering and physics calculations.

CUDA allows languages such as C/C++ to take advantage of high-speed GPUs in computationally intensive programs.

CUDA

CUDA
Course 2

We made the second course **project** based, and carefully chose a list of problems that would lend themselves to CUDA processing.

CUDA

CUDA
Course 2

We made the second course **project** based, and carefully chose a list of problems that would lend themselves to CUDA processing.

But like in the first course, we started with a **CUDA** version of **“Hello World”** which worked quite well.

CUDA

CUDA
Course 2

We made the second course **project** based, and carefully chose a list of problems that would lend themselves to CUDA processing.

For example:

- **More matrix multiplication**
- Large number multiplication
- Prime number checking
- Password cracking

more matrix multiplication

$$\begin{bmatrix} 3 & 4 & 2 & 6 & 7 & 8 \\ 8 & 0 & 9 & 6 & 5 & 4 \\ 6 & 1 & 2 & 3 & 0 & 8 \\ 4 & 8 & 9 & 5 & 7 & 6 \\ 0 & 5 & 4 & 0 & 0 & 9 \\ 8 & 6 & 1 & 1 & 0 & 7 \end{bmatrix} \times \begin{bmatrix} 6 & 7 & 5 & 4 & 3 & 0 \\ 2 & 8 & 0 & 9 & 5 & 4 \\ 2 & 4 & 7 & 6 & 5 & 4 \\ 9 & 0 & 5 & 0 & 0 & 1 \\ 4 & 2 & 8 & 6 & 7 & 5 \\ 5 & 0 & 9 & 1 & 2 & 6 \end{bmatrix} = \begin{bmatrix} - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & * & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix}$$

*** = 6 x 3 + 1 x 5 + 2 x 5 + 3 x 0 + 0 x 7 + 8 x 2**

Each cell's required data is sent to a processor with MPI code.

That processor sends out all multiplications to the GPUs by using CUDA code. [Blue font color represents CUDA calculations.]

CUDA

CUDA
Course 2

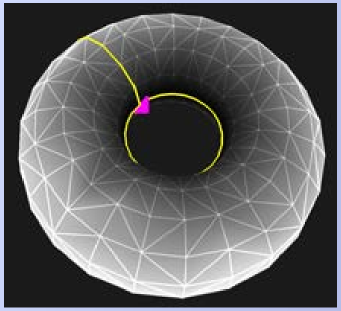
Basing all projects on a **shell** obtained from our **Hello World** was a great method of generating programs.

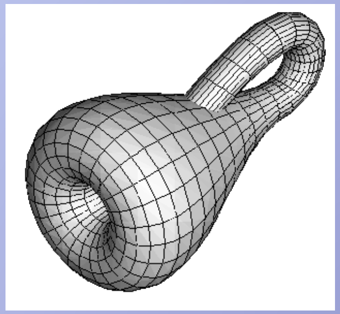
The projects were more than **enough** for a semester of work for the students.

High Performance Computing

HPC
Course 3

After getting a year of parallel programming under our belts, we decided to tackle a research-level math problem and make it the class project for an entire course.





High Performance Computing

HPC
Course 3

A colleague co-authored a mathematical paper studying the behavior of a particular class of functions as they traversed a path along a once-punctured torus.

All values were **complex** numbers, and paths were **integrated** in the complex field, so computation was intensive.

High Performance Computing

HPC
Course 3

The problem was based upon a construction of a **ternary tree** based upon certain convergence criteria, and computations once again begged for parallelization.

It worked!

High Performance Computing

HPC
Course 3

We got lots of help from the Oklahoma Supercomputer Center for Education and Research (**OSCER**).

After we "perfected" our programs, we ran them on OU's supercomputer, using up to 2,000 processors per run, racking up several **CPU-years** of run time.

Observations . . .

- We used “Special Seminars” so we didn’t have to go through Curriculum Committees, etc.

- Students adapted quickly to the parallel concept, even those of limited experience.

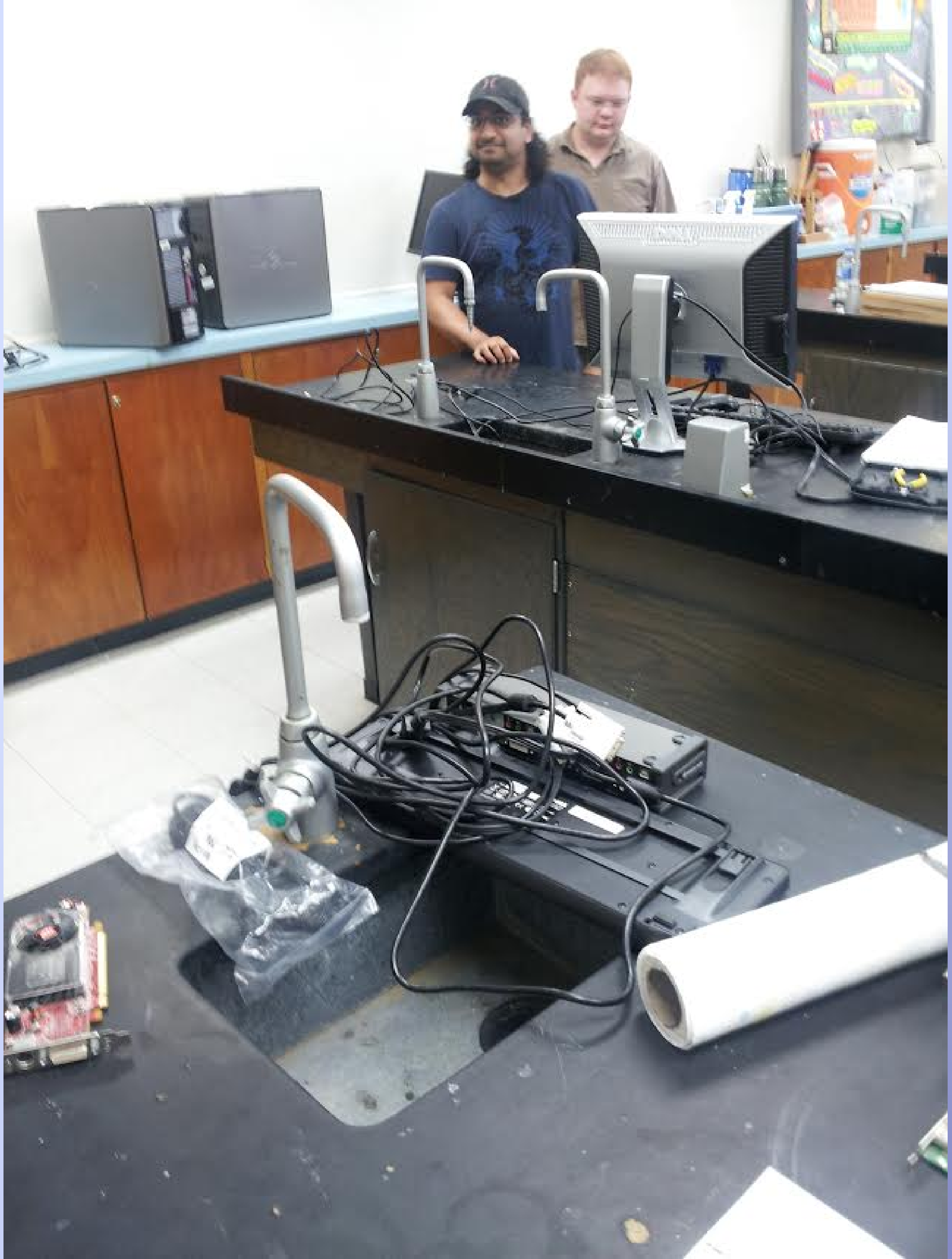
- The association with Oklahoma University’s supercomputer people was invaluable.

Newest Course . . .

- We are injecting a hardware component into the first course.
- We have procured about 40 retired office computers and the students will build several clusters.
- We’re not looking for speed, but we are demonstrating HPC techniques.

Up-to-the-minute activities . . .

We commandeered an old science lab!



Up-to-the-minute activities . . .

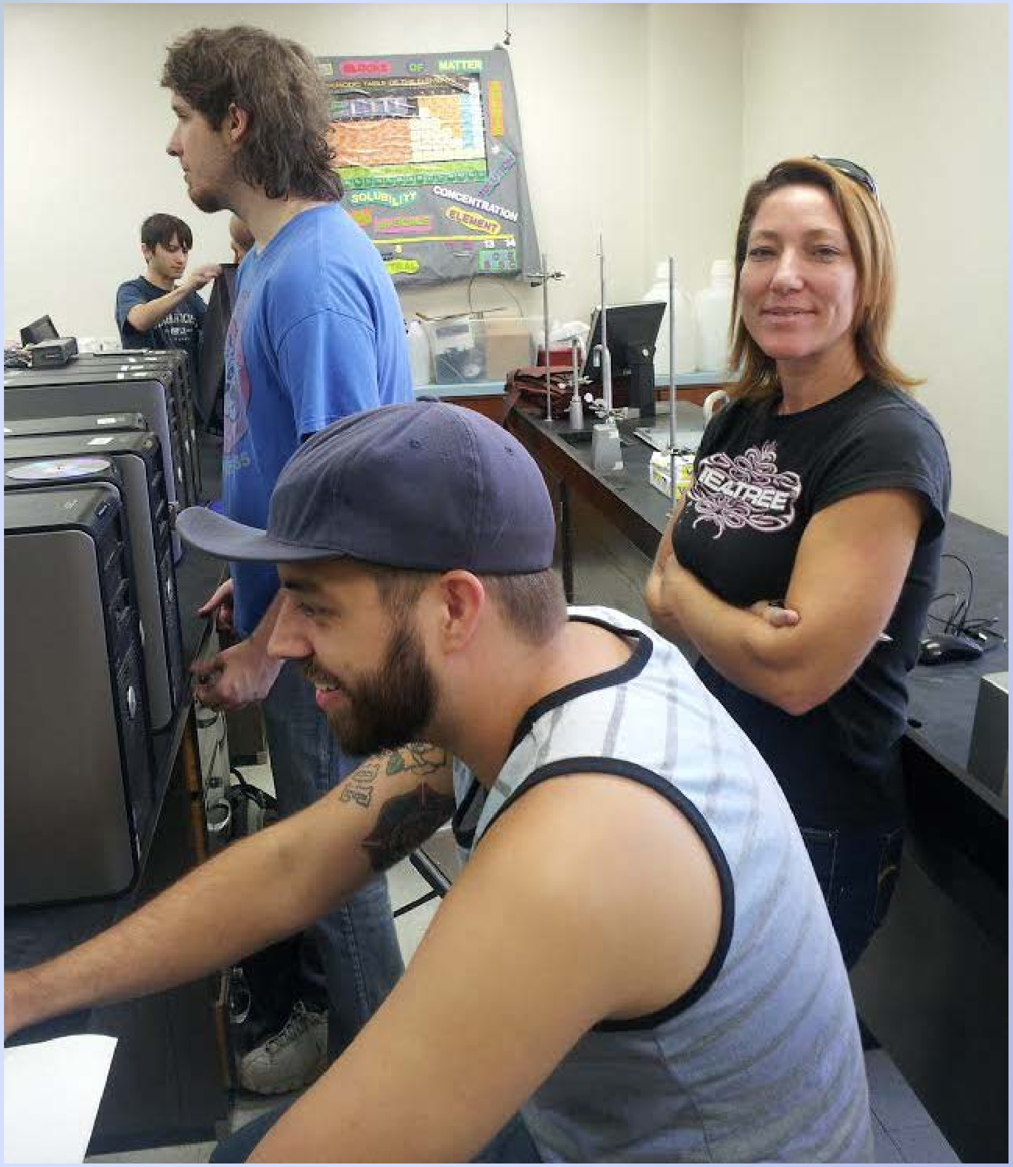
Everyone has a sink!

We catalogued and “fixed” the cast off computers.

Up-to-the-minute activities . . .

We broke into 3 teams, each to build their own super computer!

Up-to-the-minute activities . . .



Students love the format –

per the latest pedagogy, we have a high entertainment value in the class.

Goals Accomplished

- We’ve spent almost nothing.
- We’ve delivered 4 separate HPC courses.
- We’ve generated interest in permanent courses.
- We’ve reached across disciplines.
- We’ve stayed in accreditation guidelines.
- We’re on the cutting edge of CS.

Thank You!

We especially thank Henry Neeman, Charlie Peck, Tom Murphy, Bob Panoff and everyone else associated with OU IT, the LittleFe project, the SOSU IT guys and all of our colleagues and friends in the educational community involved with HPC, for all the help we have received.

Karl Frinkle
Mike Morris