

Visualization and analysis with Nautilus: From standard tools to unusual challenges

Amy F. Szczepański

Remote Data Analysis and Visualization Center, University of Tennessee

aszczepa@utk.edu

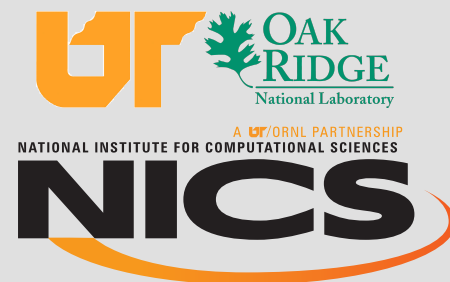
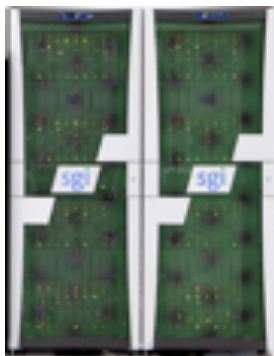


Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation



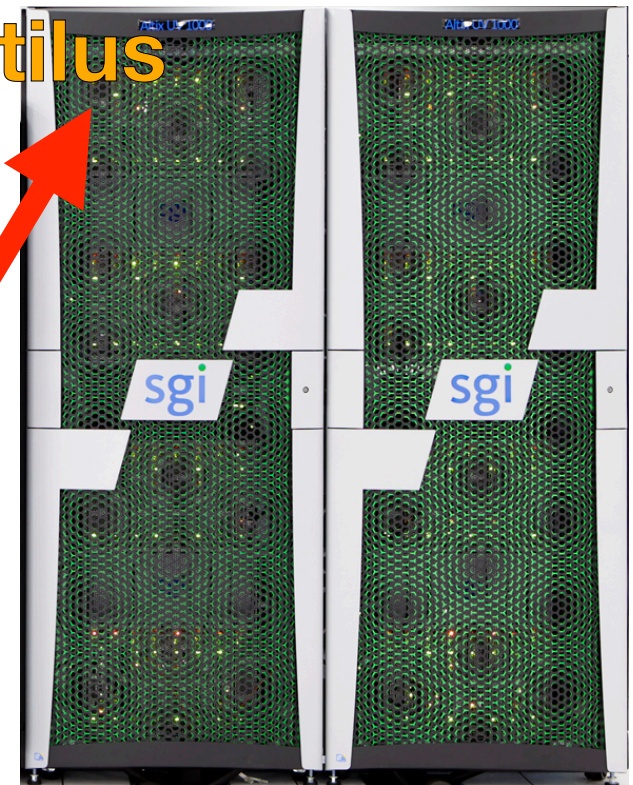
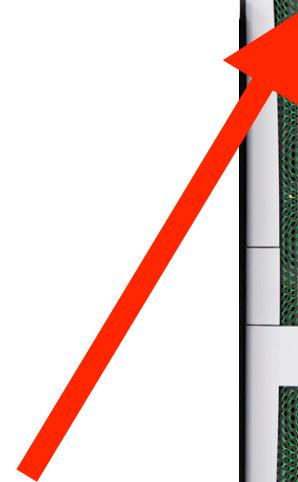


RDIV



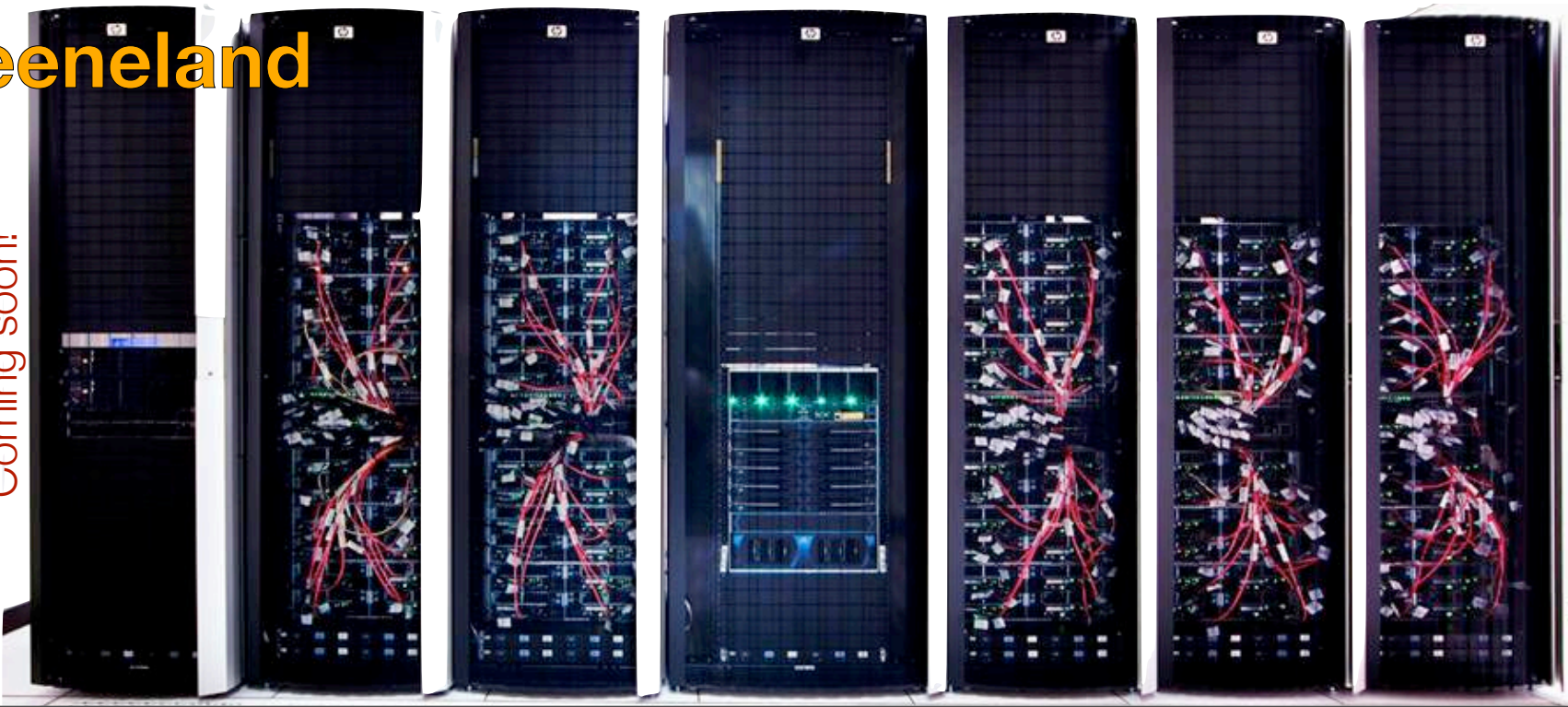


Nautilus



Keeneland

Coming soon!



Nautilus

SGI Altix UV 1000

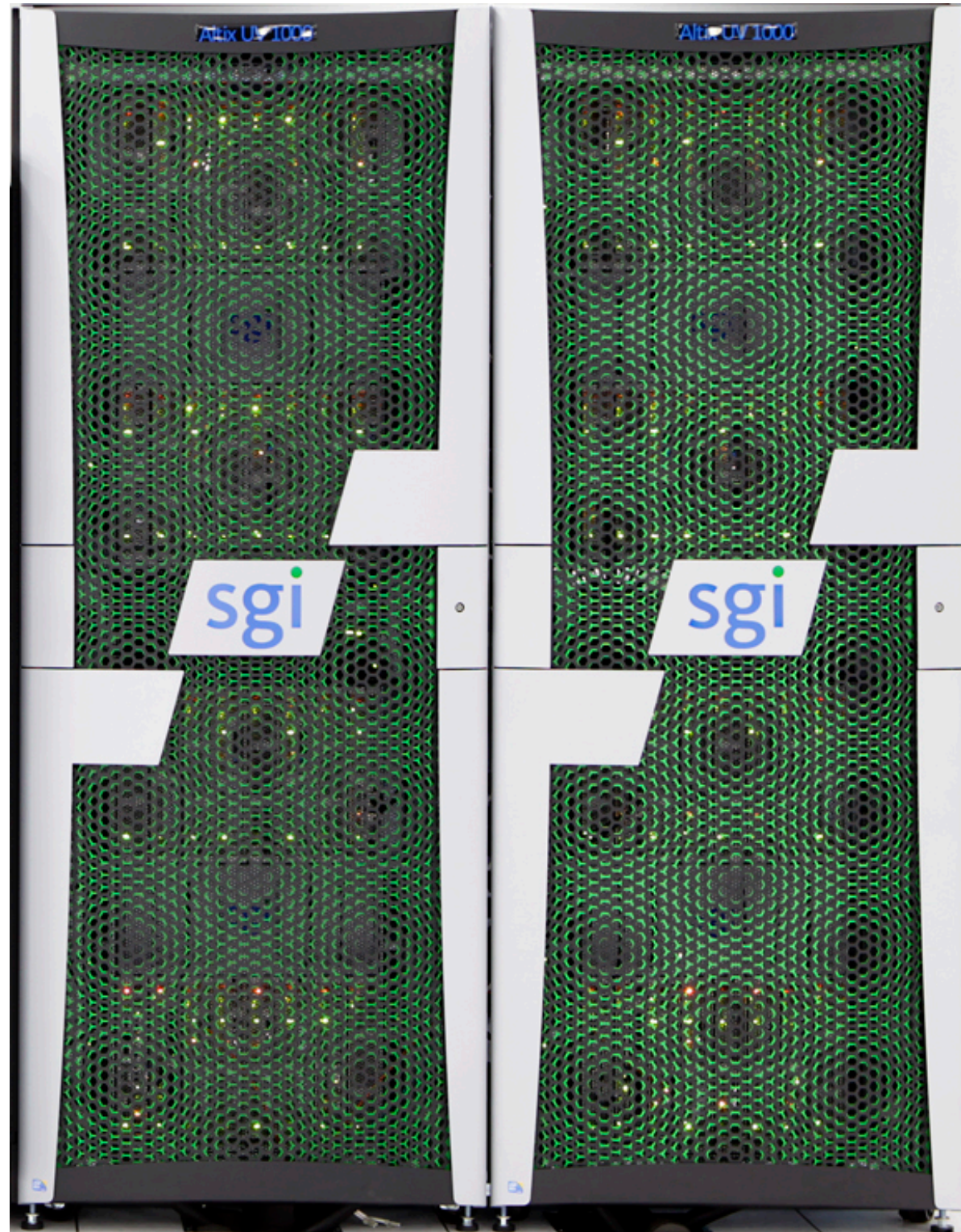
1024 cores

4 TB global
shared memory

Lustre file system

GPUs

Visualization & analysis,
simulation



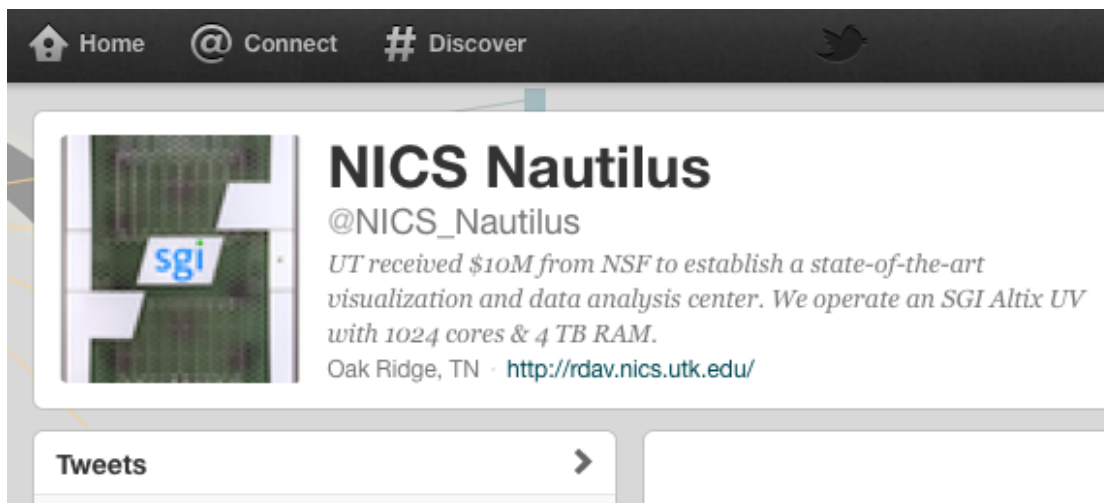
Keep up with RDAV and Nautilus



The screenshot shows the Facebook profile of the Remote Data Analysis and Visualization Center (RDAV). The profile picture is the RDAV logo. The cover photo shows a group of people in a meeting. The page is categorized as 'Computers/Technology' and is located in 'Oak Ridge, Tennessee'. The 'About' section states: 'We operate an SGI Altix UV1000 with 1024 cores and 4 TB shared memory to he...'. A post from the center is visible, titled 'Remote Data Analysis and Visualization Center', with the text: 'RDAV is hiring undergraduates for summer 2012! http://rdav.nics.tennessee.edu/interns2012'. Below this, a link is provided: 'Summer 2012 Internships for Undergraduates | rdav rdav.nics.tennessee.edu'. A red arrow points from the text 'We are hiring undergrads for summer 2012!' to this post.

Like us on Facebook!

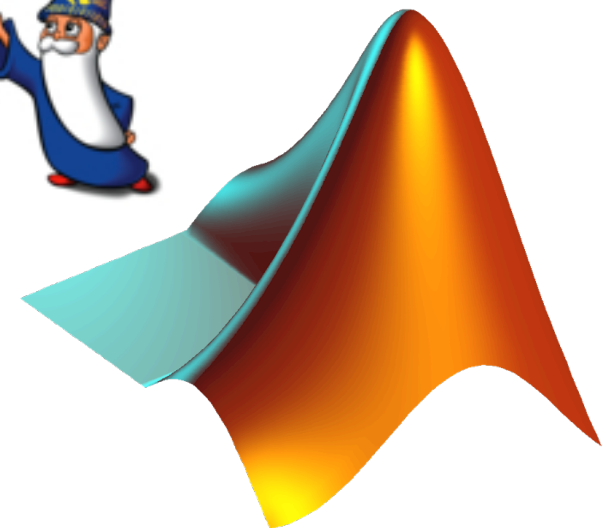
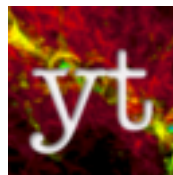
We are hiring
undergrads for
summer 2012!



The screenshot shows the Twitter profile of NICS Nautilus (@NICS_Nautilus). The profile picture is a logo with the letters 'sgi'. The bio states: 'UT received \$10M from NSF to establish a state-of-the-art visualization and data analysis center. We operate an SGI Altix UV with 1024 cores & 4 TB RAM. Oak Ridge, TN · http://rdav.nics.utk.edu/'. The 'Tweets' section is visible at the bottom.

Follow us on Twitter!
`@NICS_Nautilus`

[http://www.nics.tennessee.edu/
computing-resources/nautilus/software](http://www.nics.tennessee.edu/computing-resources/nautilus/software)



```
#!/bin/bash
```

From an actual ticket

```
#PBS -A TG-ZZZ12000
```

```
#PBS -l size=12
```

```
cd $PBS_O_WORKDIR
```

```
aprun -n 1 -d 12 -cc none -a xt ./run.sh
```

```
/lustre/scratch/username/run_single.sh 0&
```

```
/lustre/scratch/username/run_single.sh 1&
```

```
/lustre/scratch/username/run_single.sh 2&
```

```
/lustre/scratch/username/run_single.sh 3&
```

```
/lustre/scratch/username/run_single.sh 4&
```

```
/lustre/scratch/username/run_single.sh 5&
```

```
/lustre/scratch/username/run_single.sh 6&
```

```
/lustre/scratch/username/run_single.sh 7&
```

```
/lustre/scratch/username/run_single.sh 8&
```

```
/lustre/scratch/username/run_single.sh 9&
```

```
/lustre/scratch/username/run_single.sh 10&
```

```
/lustre/scratch/username/run_single.sh 11&
```

```
wait
```

Dear user support,
I would really appreciate your help with the following issue. I have to perform post-processing data extraction that is not parallelized. I can subdivide it into 192 single-processor programs that consist of 192 copies of the same serial program (with different input parameters)



Discover Life in America: All Taxa Biodiversity Inventory

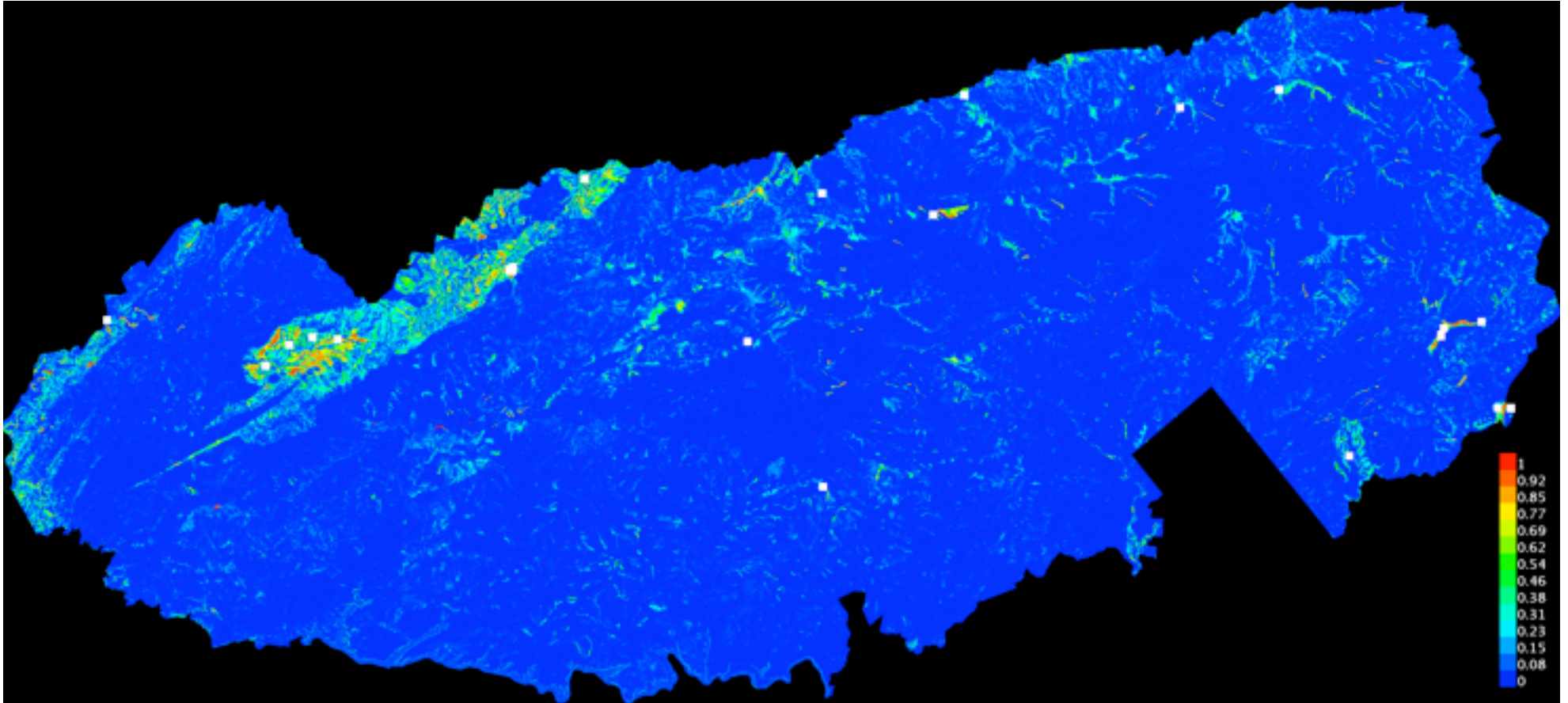


Image and data courtesy: William Godsoe (NIMBioS) and Keith Langdon USDA/National Park Service

Bombus impatiens in GSMNP

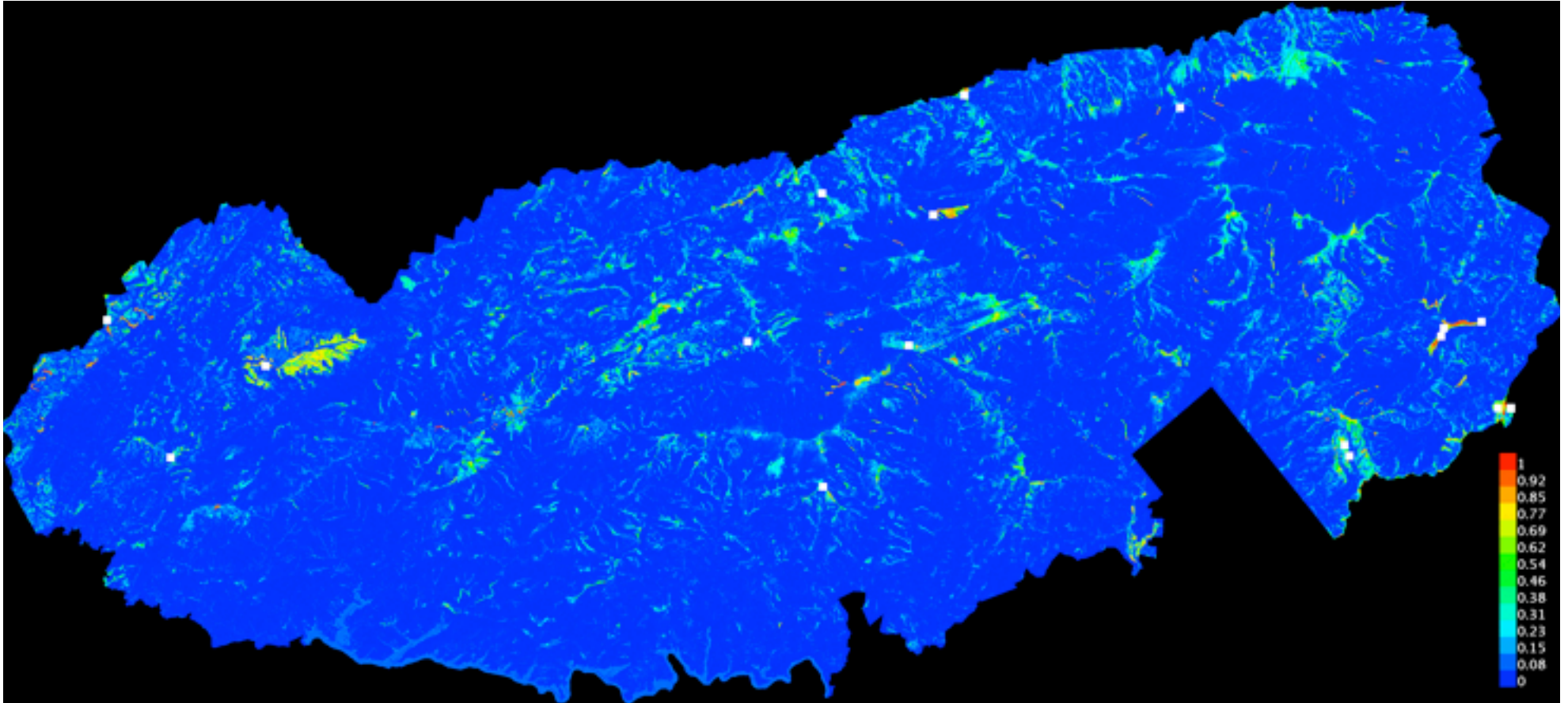
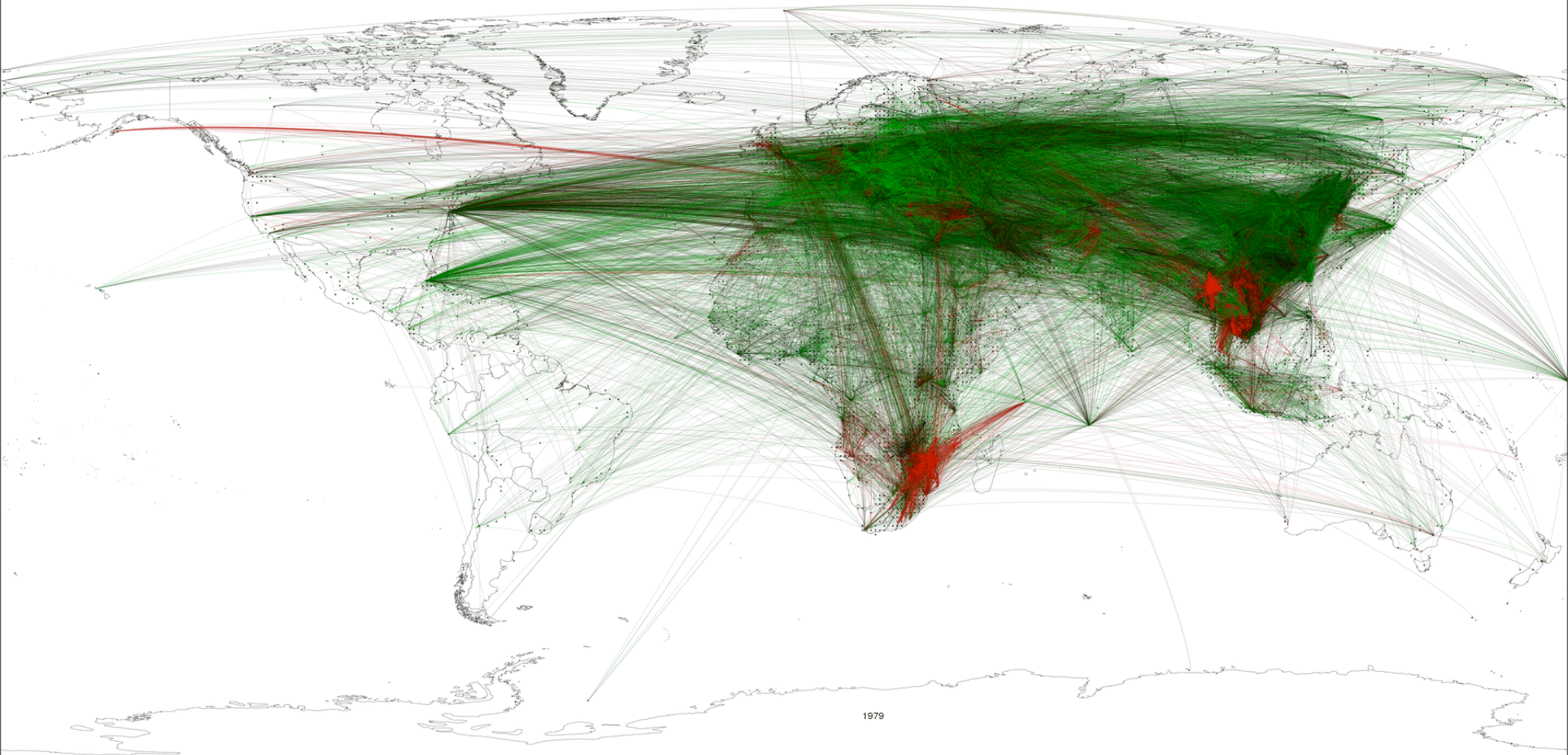


Image and data courtesy: William Godsoe (NIMBioS) and Keith Langdon USDA/National Park Service

Bombus vagans in GSMNP

Kalev Leetaru: The GlobalNet Project



“I’m doing some runs where I need to run a particular **serial code in 16,000 different configurations**. It is launched by handing the configuration options via command line options. Right now I have a PBS job that requests 16 CPUs and then I have a little PERL script that runs at startup...Is there a quick-and-easy toolkit anywhere that would allow me to just hand it a file containing the 16,000 exec commands for my code and it will **automatically load-balance** them across the CPUs in my job and ensure that all CPUs are always working? That would be a massive, massive help.”

—Kalev Leetaru (Nautilus user)

```
#PBS -A TG-STA110018S
```

```
#PBS -l ncpus=8,walltime=01:00:00
```

```
./analyze_data file0 low &
```

```
./analyze_data file0 high &
```

```
./analyze_data file1 low &
```

```
./analyze_data file1 high &
```

```
./analyze_data file2 low &
```

```
./analyze_data file2 high &
```

```
./analyze_data file3 low &
```

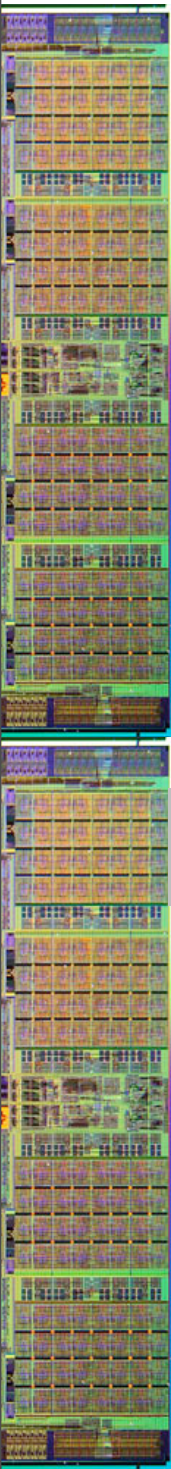
```
./analyze_data file3 high &
```

```
wait
```


0:00

0:30

1:00



file0 low

file0 high

file1 low

file1 high

file2 low

file2 high

file3 low

file3 high

Eden: by Scott Simmerman (RDAV)

<http://rdav.nics.tennessee.edu/eden>

[http://sourceforge.net/projects/
rdaveden/](http://sourceforge.net/projects/rdaveden/)



A terminal window titled "szczepanski — bash — 80x24" with standard macOS window controls. The prompt is "amymac:~ szczepanski\$". The command "ssh szczepa@login.nautilus.nics.utk.edu" is entered and the cursor is at the end of the line. A mouse cursor is visible on the right side of the terminal area.

```
amymac:~ szczepanski$ ssh szczepa@login.nautilus.nics.utk.edu
```


ECS.o74331

Job Id: 74331.nemo.nics.utk.edu

Job_Name = ECS

resources_used.cput = 00:00:00

resources_used.mem = 41428kb

resources_used.vmem = 483308kb

resources_used.walltime = 00:00:39

```
szczepa@arronax:/lustre/medusa/szczepa/edendemo/my_runs> cd outfiles
```

```
szczepa@arronax:/lustre/medusa/szczepa/edendemo/my_runs/outfiles> ls
```

| | | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 00.err | 09.err | 18.err | 27.err | 36.err | 45.err | 54.err | 63.err | 72.err |
| 00.out | 09.out | 18.out | 27.out | 36.out | 45.out | 54.out | 63.out | 72.out |
| 00.time | 09.time | 18.time | 27.time | 36.time | 45.time | 54.time | 63.time | 72.time |
| 01.err | 10.err | 19.err | 28.err | 37.err | 46.err | 55.err | 64.err | 73.err |
| 01.out | 10.out | 19.out | 28.out | 37.out | 46.out | 55.out | 64.out | 73.out |
| 01.time | 10.time | 19.time | 28.time | 37.time | 46.time | 55.time | 64.time | 73.time |
| 02.err | 11.err | 20.err | 29.err | 38.err | 47.err | 56.err | 65.err | 74.err |
| 02.out | 11.out | 20.out | 29.out | 38.out | 47.out | 56.out | 65.out | 74.out |
| 02.time | 11.time | 20.time | 29.time | 38.time | 47.time | 56.time | 65.time | 74.time |
| 03.err | 12.err | 21.err | 30.err | 39.err | 48.err | 57.err | 66.err | 75.err |
| 03.out | 12.out | 21.out | 30.out | 39.out | 48.out | 57.out | 66.out | 75.out |
| 03.time | 12.time | 21.time | 30.time | 39.time | 48.time | 57.time | 66.time | 75.time |
| 04.err | 13.err | 22.err | 31.err | 40.err | 49.err | 58.err | 67.err | 76.err |
| 04.out | 13.out | 22.out | 31.out | 40.out | 49.out | 58.out | 67.out | 76.out |
| 04.time | 13.time | 22.time | 31.time | 40.time | 49.time | 58.time | 67.time | 76.time |
| 05.err | 14.err | 23.err | 32.err | 41.err | 50.err | 59.err | 68.err | 77.err |
| 05.out | 14.out | 23.out | 32.out | 41.out | 50.out | 59.out | 68.out | 77.out |
| 05.time | 14.time | 23.time | 32.time | 41.time | 50.time | 59.time | 68.time | 77.time |
| 06.err | 15.err | 24.err | 33.err | 42.err | 51.err | 60.err | 69.err | 78.err |
| 06.out | 15.out | 24.out | 33.out | 42.out | 51.out | 60.out | 69.out | 78.out |
| 06.time | 15.time | 24.time | 33.time | 42.time | 51.time | 60.time | 69.time | 78.time |
| 07.err | 16.err | 25.err | 34.err | 43.err | 52.err | 61.err | 70.err | 79.err |
| 07.out | 16.out | 25.out | 34.out | 43.out | 52.out | 61.out | 70.out | 79.out |
| 07.time | 16.time | 25.time | 34.time | 43.time | 52.time | 61.time | 70.time | 79.time |
| 08.err | 17.err | 26.err | 35.err | 44.err | 53.err | 62.err | 71.err | |
| 08.out | 17.out | 26.out | 35.out | 44.out | 53.out | 62.out | 71.out | |
| 08.time | 17.time | 26.time | 35.time | 44.time | 53.time | 62.time | 71.time | |

| | A | B | C | D | E | F | G |
|----|----------------|-----------|-----------|----------|-----------------|-----------------|------------------|
| 1 | #command_index | real_time | user_time | sys_time | stderr_filesize | stdout_filesize | command_line |
| 2 | 0 | 0.07 | 0 | 0.02 | 0 | 2 | ./program 0 0 |
| 3 | 1 | 0.06 | 0 | 0.03 | 0 | 5 | ./program 0 1000 |
| 4 | 2 | 0.07 | 0 | 0.03 | 0 | 5 | ./program 0 2000 |
| 5 | 3 | 0.06 | 0 | 0.03 | 0 | 5 | ./program 0 3000 |
| 6 | 4 | 0.06 | 0 | 0.03 | 0 | 5 | ./program 0 4000 |
| 7 | 5 | 0.05 | 0.01 | 0.02 | 0 | 5 | ./program 0 5000 |
| 8 | 6 | 0.07 | 0.01 | 0.02 | 0 | 5 | ./program 0 6000 |
| 9 | 7 | 0.07 | 0 | 0.02 | 0 | 5 | ./program 0 7000 |
| 10 | 8 | 1.06 | 0 | 0.02 | 0 | 2 | ./program 1 0 |
| 11 | 9 | 1.08 | 0 | 0.02 | 0 | 5 | ./program 1 1000 |
| 12 | 10 | 1.06 | 0 | 0.02 | 0 | 5 | ./program 1 2000 |
| 13 | 11 | 1.06 | 0 | 0.02 | 0 | 5 | ./program 1 3000 |
| 14 | 12 | 1.06 | 0 | 0.03 | 0 | 5 | ./program 1 4000 |
| 15 | 13 | 1.07 | 0 | 0.02 | 0 | 5 | ./program 1 5000 |
| 16 | 14 | 1.05 | 0.01 | 0 | 0 | 5 | ./program 1 6000 |
| 17 | 15 | 1.05 | 0 | 0.01 | 0 | 5 | ./program 1 7000 |
| 18 | 16 | 2.11 | 0 | 0.02 | 0 | 2 | ./program 2 0 |
| 19 | 17 | 2.11 | 0 | 0.02 | 0 | 5 | ./program 2 1000 |
| 20 | 18 | 2.11 | 0 | 0.02 | 0 | 5 | ./program 2 2000 |
| 21 | 19 | 2.11 | 0 | 0.02 | 0 | 5 | ./program 2 3000 |
| 22 | 20 | 2.11 | 0 | 0.02 | 0 | 5 | ./program 2 4000 |
| 23 | 21 | 2.1 | 0 | 0.02 | 0 | 5 | ./program 2 5000 |
| 24 | 22 | 2.11 | 0.01 | 0.01 | 0 | 5 | ./program 2 6000 |
| 25 | 23 | 2.1 | 0 | 0.02 | 0 | 5 | ./program 2 7000 |
| 26 | 24 | 3.06 | 0 | 0.02 | 0 | 2 | ./program 3 0 |
| 27 | 25 | 3.04 | 0.01 | 0.01 | 0 | 5 | ./program 3 1000 |
| 28 | 26 | 3.05 | 0 | 0.02 | 0 | 5 | ./program 3 2000 |
| 29 | 27 | 3.04 | 0.01 | 0 | 0 | 5 | ./program 3 3000 |
| 30 | 28 | 3.04 | 0.01 | 0 | 0 | 5 | ./program 3 4000 |
| 31 | 29 | 3.05 | 0.01 | 0 | 0 | 5 | ./program 3 5000 |
| 32 | 30 | 3.05 | 0 | 0.01 | 0 | 5 | ./program 3 6000 |
| 33 | 31 | 3.05 | 0 | 0.01 | 0 | 5 | ./program 3 7000 |

Advanced Features

- Restart an interrupted run
- Automatically generate commands
- More than one core per process

“I will say that Eden is a great, great tool; it is proving *tremendously* useful and really helping to speed things up for me...”

—Kalev Leetaru (Nautilus user)