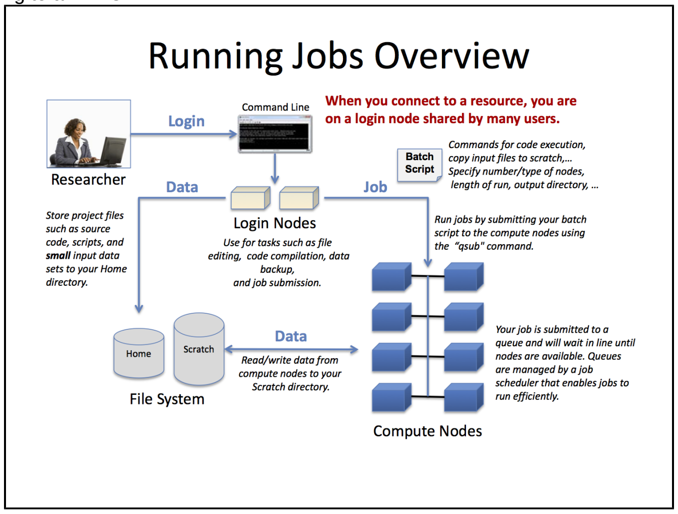
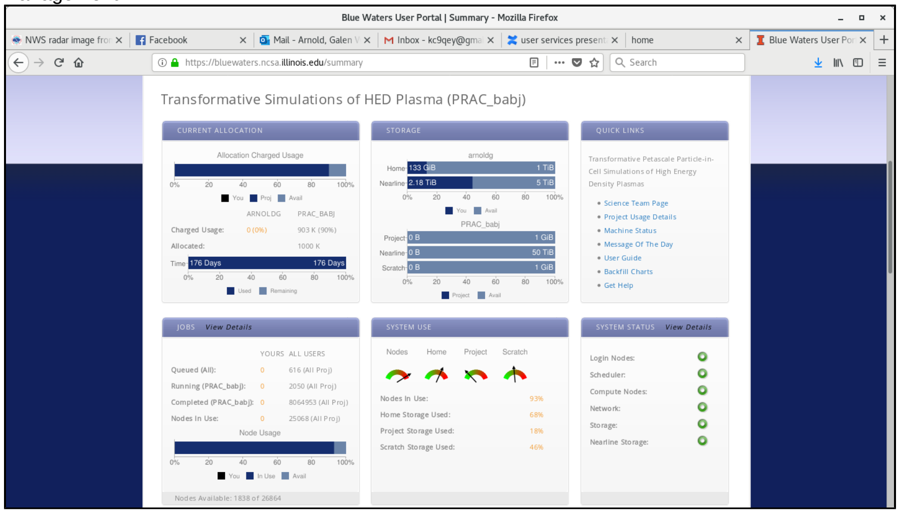
User Services Intro.

- Connecting to an HPC
  - 
Running Jobs Overview
Login
Command Line
When you connect to a resource, you are on a login node shared by many users.
Researcher
Batch Script
Commands for code execution, copy input files to scratch,… Specify number/type of nodes, length of run, output directory, …
Data
Job
Login Nodes
Use for tasks such as file editing, code compilation, data backup, and job submission.
Store project files such as source code, scripts, and small input data sets to your Home directory.
Run jobs by submitting your batch script to the compute nodes using the "qsub" command.
Home
Scratch
Data
Read/write data from compute nodes to your Scratch directory.
File System
Your job is submitted to a queue and will wait in line until nodes are available. Queues are managed by a job scheduler that enables jobs to run efficiently.
Compute Nodes
- Account management
  - 

Transformative Simulations of HED Plasma (PRAC_babj)

CURRENT ALLOCATION

Allocation Charged Usage

	ARNOLDG	PRAC_BABJ
Charged Usage:	0 (0%)	903 K (90%)
Allocated:		1000 K

Time 176 Days 176 Days

STORAGE

arnoldg

Home 133 GiB 1 TiB

Nearline 2.18 TiB 5 TiB

PRAC_babj

Project 0 B 1 GiB

Nearline 0 B 50 TiB

Scratch 0 B 1 GiB

QUICK LINKS

Transformative Petascale Particle-in-Cell Simulations of High Energy Density Plasmas

- Science Team Page
- Project Usage Details
- Machine Status
- Message Of The Day
- User Guide
- Backfill Charts
- Get Help

JOBS *View Details*

	YOURS	ALL USERS
Queued (All):	0	616 (All Proj)
Running (PRAC_babj):	0	2050 (All Proj)
Completed (PRAC_babj):	0	8064953 (All Proj)
Nodes In Use:	0	25068 (All Proj)

Node Usage

Nodes Available: 1838 of 26864

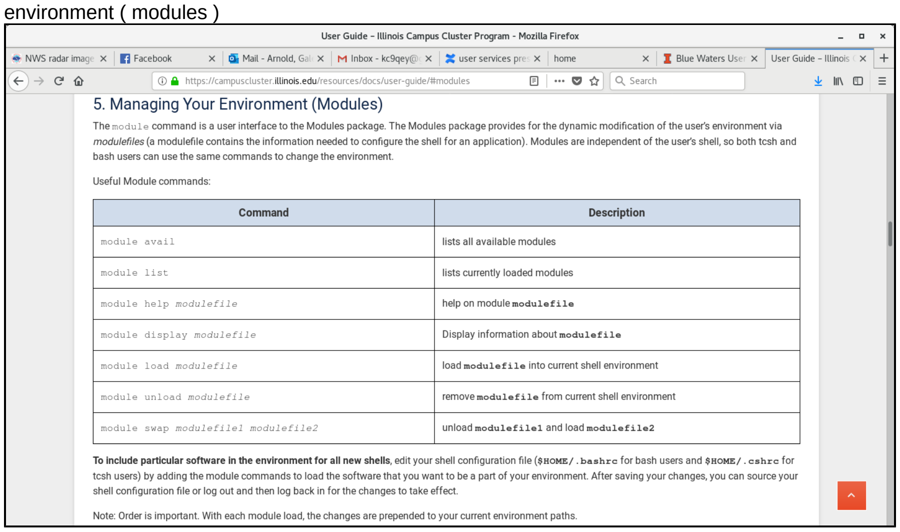
SYSTEM USE

Nodes Home Project Scratch

Nodes In Use:	93%
Home Storage Used:	68%
Project Storage Used:	18%
Scratch Storage Used:	46%

SYSTEM STATUS *View Details*

Login Nodes:	●
Scheduler:	●
Compute Nodes:	●
Network:	●
Storage:	●
Nearline Storage:	●

  - environment (modules)
  - 
User Guide – Illinois Campus Cluster Program - Mozilla Firefox
NWS radar image
Facebook
Mail - Arnold, Gal
Inbox - kc9qey@
user services pres
home
Blue Waters User
User Guide – Illinois
https://campuscluster.illinois.edu/resources/docs/user-guide/#modules
Search

5. Managing Your Environment (Modules)

The `module` command is a user interface to the Modules package. The Modules package provides for the dynamic modification of the user's environment via *modulefiles* (a modulefile contains the information needed to configure the shell for an application). Modules are independent of the user's shell, so both tcsh and bash users can use the same commands to change the environment.

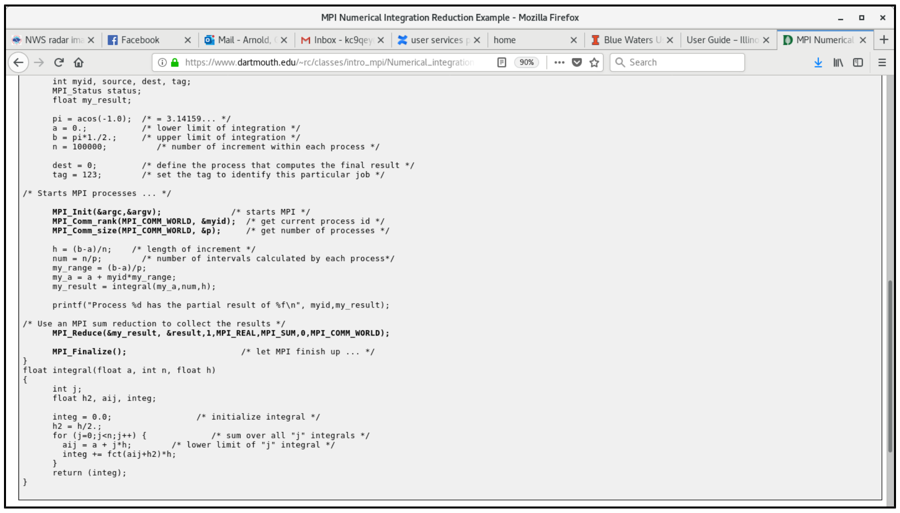
Useful Module commands:

Command	Description
`module avail`	lists all available modules
`module list`	lists currently loaded modules
`module help modulefile`	help on module **modulefile**
`module display modulefile`	Display information about **modulefile**
`module load modulefile`	load **modulefile** into current shell environment
`module unload modulefile`	remove **modulefile** from current shell environment
`module swap modulefile1 modulefile2`	unload **modulefile1** and load **modulefile2**

To include particular software in the environment for all new shells, edit your shell configuration file (**$HOME/.bashrc** for bash users and **$HOME/.cshrc** for tcsh users) by adding the module commands to load the software that you want to be a part of your environment. After saving your changes, you can source your shell configuration file or log out and then log back in for the changes to take effect.

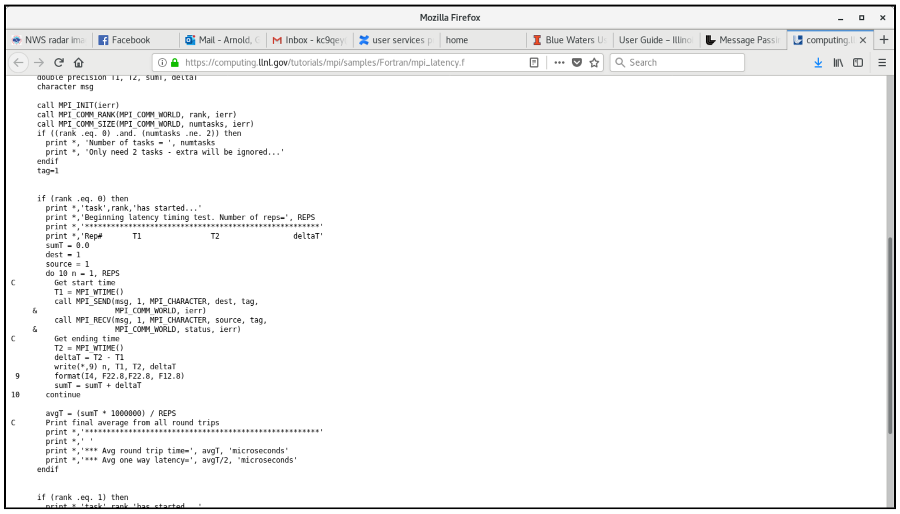
Note: Order is important. With each module load, the changes are prepended to your current environment paths.

  - account usage, project participants, quotas
- Programming environment
  - traditional HPC languages (scale well, high efficiency): c and fortran


MPI Numerical Integration Reduction Example - Mozilla Firefox
NWS radar ima
Facebook
Mail - Arnold,
Inbox - kc9qey
user services p
home
Blue Waters U
User Guide – Illino
MPI Numerical
https://www.dartmouth.edu/~rc/classes/intro_mpi/Numerical_integration
90%
Search

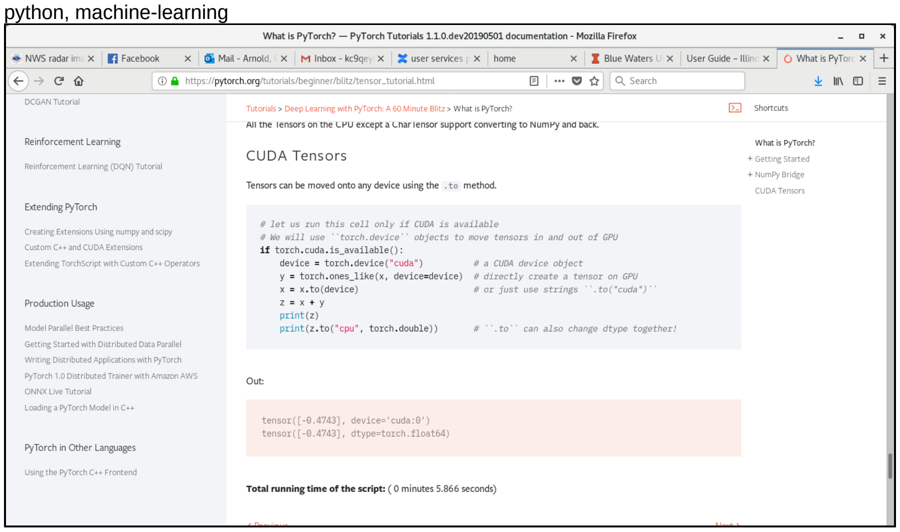
```c
      int myid, source, dest, tag;
      MPI_Status status;
      float my_result;

      pi = acos(-1.0);  /* = 3.14159... */
      a = 0.;           /* lower limit of integration */
      b = pi*1./2.;     /* upper limit of integration */
      n = 100000;          /* number of increment within each process */

      dest = 0;         /* define the process that computes the final result */
      tag = 123;        /* set the tag to identify this particular job */

/* Starts MPI processes ... */

      MPI_Init(&argc,&argv);              /* starts MPI */
      MPI_Comm_rank(MPI_COMM_WORLD, &myid);  /* get current process id */
      MPI_Comm_size(MPI_COMM_WORLD, &p);     /* get number of processes */

      h = (b-a)/n;    /* length of increment */
      num = n/p;        /* number of intervals calculated by each process*/
      my_range = (b-a)/p;
      my_a = a + myid*my_range;
      my_result = integral(my_a,num,h);

      printf("Process %d has the partial result of %f\n", myid,my_result);

/* Use an MPI sum reduction to collect the results */
      MPI_Reduce(&my_result, &result,1,MPI_REAL,MPI_SUM,0,MPI_COMM_WORLD);

      MPI_Finalize();                       /* let MPI finish up ... */
}
float integral(float a, int n, float h)
{
      int j;
      float h2, aij, integ;

      integ = 0.0;                 /* initialize integral */
      h2 = h/2.;
      for (j=0;j<n;j++) {              /* sum over all "j" integrals */
        aij = a + j*h;        /* lower limit of "j" integral */
        integ += fct(aij+h2)*h;
      }
      return (integ);
}
```

○



```fortran
      double precision T1, T2, sumT, deltaT
      character msg

      call MPI_INIT(ierr)
      call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierr)
      call MPI_COMM_SIZE(MPI_COMM_WORLD, numtasks, ierr)
      if ((rank .eq. 0) .and. (numtasks .ne. 2)) then
        print *, 'Number of tasks = ', numtasks
        print *, 'Only need 2 tasks - extra will be ignored...'
      endif
      tag=1


      if (rank .eq. 0) then
        print *,'task',rank,'has started...'
        print *,'Beginning latency timing test. Number of reps=', REPS
        print *,'*******************************************************'
        print *,'Rep#       T1                T2                 deltaT'
        sumT = 0.0
        dest = 1
        source = 1
        do 10 n = 1, REPS
C         Get start time
          T1 = MPI_WTIME()
          call MPI_SEND(msg, 1, MPI_CHARACTER, dest, tag,
     &                  MPI_COMM_WORLD, ierr)
          call MPI_RECV(msg, 1, MPI_CHARACTER, source, tag,
     &                  MPI_COMM_WORLD, status, ierr)
C         Get ending time
          T2 = MPI_WTIME()
          deltaT = T2 - T1
          write(*,9) n, T1, T2, deltaT
 9        format(I4, F22.8,F22.8, F12.8)
          sumT = sumT + deltaT
10      continue

        avgT = (sumT * 1000000) / REPS
C       Print final average from all round trips
        print *,'*******************************************************'
        print *,' '
        print *,'*** Avg round trip time=', avgT, 'microseconds'
        print *,'*** Avg one way latency=', avgT/2, 'microseconds'
      endif


      if (rank .eq. 1) then
        print *,'task',rank,'has started...'
```

○ python, machine-learning


What is PyTorch? — PyTorch Tutorials 1.1.0.dev20190501 documentation - Mozilla Firefox
NWS radar ima
Facebook
Mail - Arnold,
Inbox - kc9qey
user services p
home
Blue Waters U
User Guide – Illino
What is PyTorc
https://pytorch.org/tutorials/beginner/blitz/tensor_tutorial.html
Search
DCGAN Tutorial
Reinforcement Learning
Reinforcement Learning (DQN) Tutorial
Extending PyTorch
Creating Extensions Using numpy and scipy
Custom C++ and CUDA Extensions
Extending TorchScript with Custom C++ Operators
Production Usage
Model Parallel Best Practices
Getting Started with Distributed Data Parallel
Writing Distributed Applications with PyTorch
PyTorch 1.0 Distributed Trainer with Amazon AWS
ONNX Live Tutorial
Loading a PyTorch Model in C++
PyTorch in Other Languages
Using the PyTorch C++ Frontend
Tutorials > Deep Learning with PyTorch: A 60 Minute Blitz > What is PyTorch?
Shortcuts
What is PyTorch?
+ Getting Started
+ NumPy Bridge
CUDA Tensors
< Previous
Next >

All the Tensors on the CPU except a CharTensor support converting to NumPy and back.

CUDA Tensors

Tensors can be moved onto any device using the `.to` method.

```python
# let us run this cell only if CUDA is available
# We will use ``torch.device`` objects to move tensors in and out of GPU
if torch.cuda.is_available():
    device = torch.device("cuda")          # a CUDA device object
    y = torch.ones_like(x, device=device)  # directly create a tensor on GPU
    x = x.to(device)                       # or just use strings ``.to("cuda")``
    z = x + y
    print(z)
    print(z.to("cpu", torch.double))       # ``.to`` can also change dtype together!
```

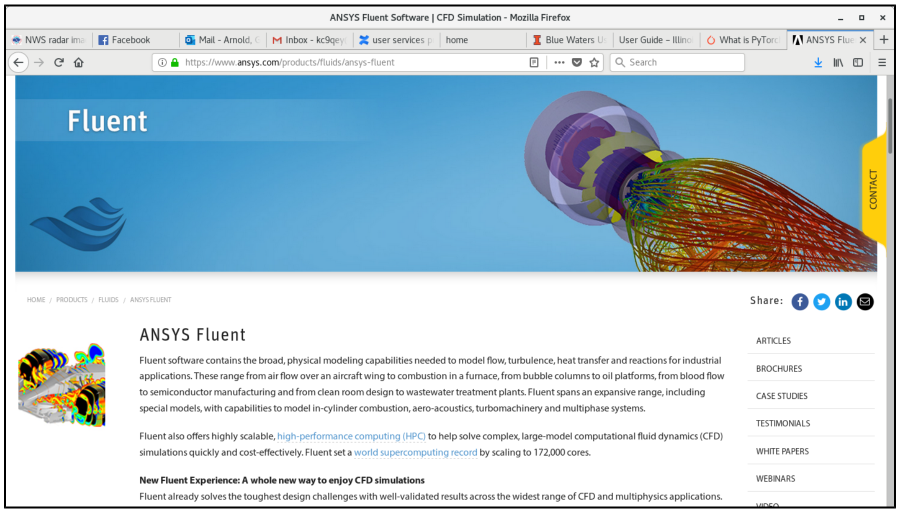
Out:

```
tensor([-0.4743], device='cuda:0')
tensor([-0.4743], dtype=torch.float64)
```

Total running time of the script: (0 minutes 5.866 seconds)

○

○ commercial software


ANSYS Fluent Software | CFD Simulation - Mozilla Firefox

https://www.ansys.com/products/fluids/ansys-fluent

Fluent

HOME / PRODUCTS / FLUIDS / ANSYS FLUENT

Share:

ANSYS Fluent

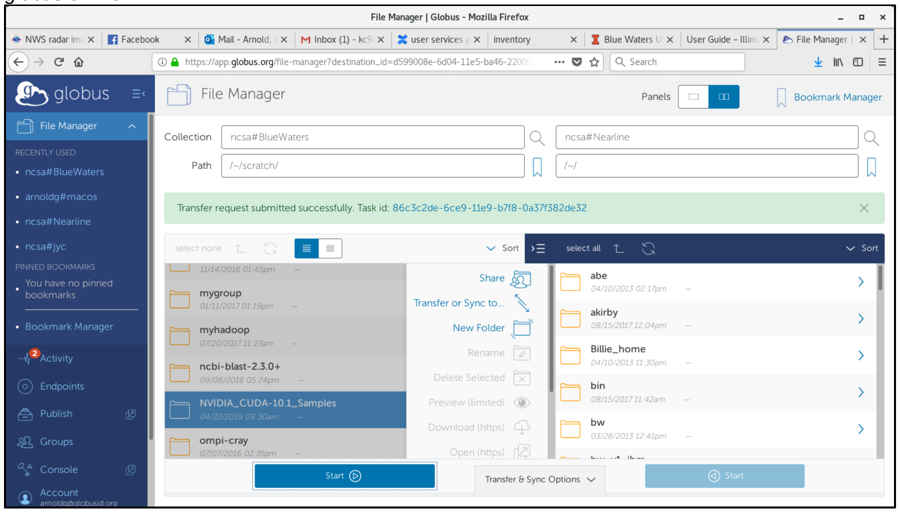
Fluent software contains the broad, physical modeling capabilities needed to model flow, turbulence, heat transfer and reactions for industrial applications. These range from air flow over an aircraft wing to combustion in a furnace, from bubble columns to oil platforms, from blood flow to semiconductor manufacturing and from clean room design to wastewater treatment plants. Fluent spans an expansive range, including special models, with capabilities to model in-cylinder combustion, aero-acoustics, turbomachinery and multiphase systems.

Fluent also offers highly scalable, high-performance computing (HPC) to help solve complex, large-model computational fluid dynamics (CFD) simulations quickly and cost-effectively. Fluent set a world supercomputing record by scaling to 172,000 cores.

New Fluent Experience: A whole new way to enjoy CFD simulations

Fluent already solves the toughest design challenges with well-validated results across the widest range of CFD and multiphysics applications.

- Data Transfer with an HPC
  - globus online


File Manager | Globus - Mozilla Firefox

https://app.globus.org/file-manager?destination_id=d599008e-6d04-11e5-ba46-2200

File Manager

Collection: ncsa#BlueWaters — ncsa#Nearline

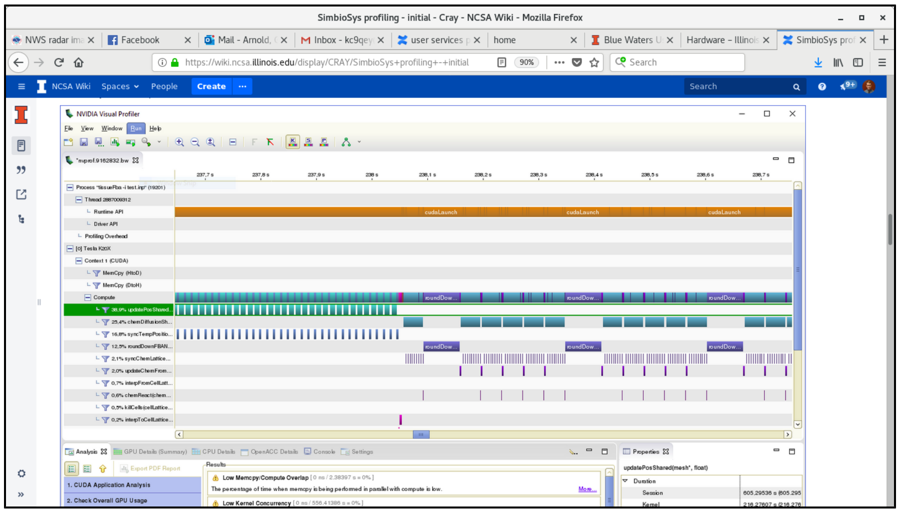
Path: /~/scratch/ — /~/

Transfer request submitted successfully. Task id: 86c3c2de-6ce9-11e9-b7f8-0a37f382de32

11/14/2016 01:45pm	
mygroup	01/11/2017 01:19pm
myhadoop	07/20/2017 11:27am
ncbi-blast-2.3.0+	09/06/2016 05:24pm
NVIDIA_CUDA-10.1_Samples	04/22/2019 09:30am
ompi-cray	07/07/2016 02:35pm

abe	04/10/2013 02:17pm
akirby	08/15/2017 12:04pm
Billie_home	04/10/2013 11:30pm
bin	08/15/2017 11:42am
bw	03/28/2013 12:41pm

  - ssh/scp/rdist
- Thinking about your workflow end to end
  - optimize performance
    - on the HPC
      - performance profiling

- 

SimbioSys profiling - initial - Cray - NCSA Wiki - Mozilla Firefox

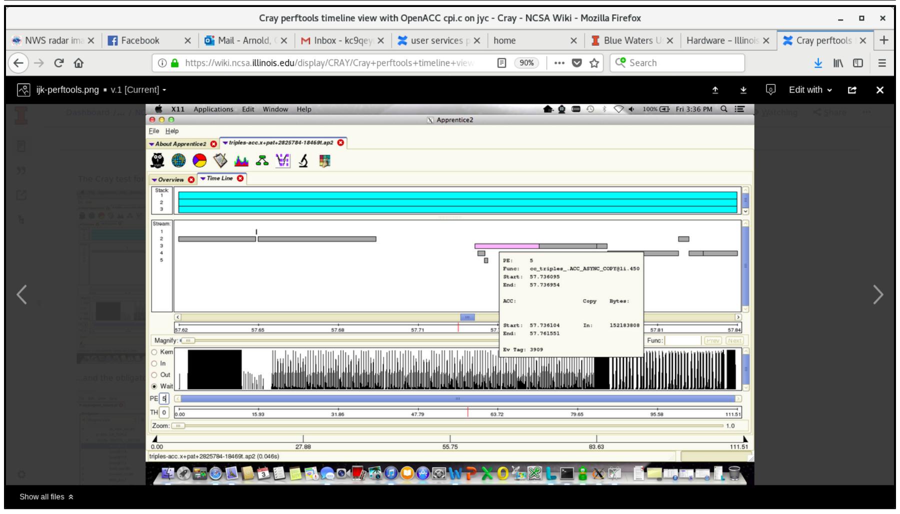
https://wiki.ncsa.illinois.edu/display/CRAY/SimbioSys+profiling+-+initial

NVIDIA Visual Profiler

Low Memcpy/Compute Overlap [0 ns / 2.38397 s = 0%]

The percentage of time when memcpy is being performed in parallel with compute is low.

Low Kernel Concurrency [0 ns / 556.41386 s = 0%]



Cray perftools timeline view with OpenACC cpi.c on jyc - Cray - NCSA Wiki - Mozilla Firefox

https://wiki.ncsa.illinois.edu/display/CRAY/Cray+perftools+timeline+view

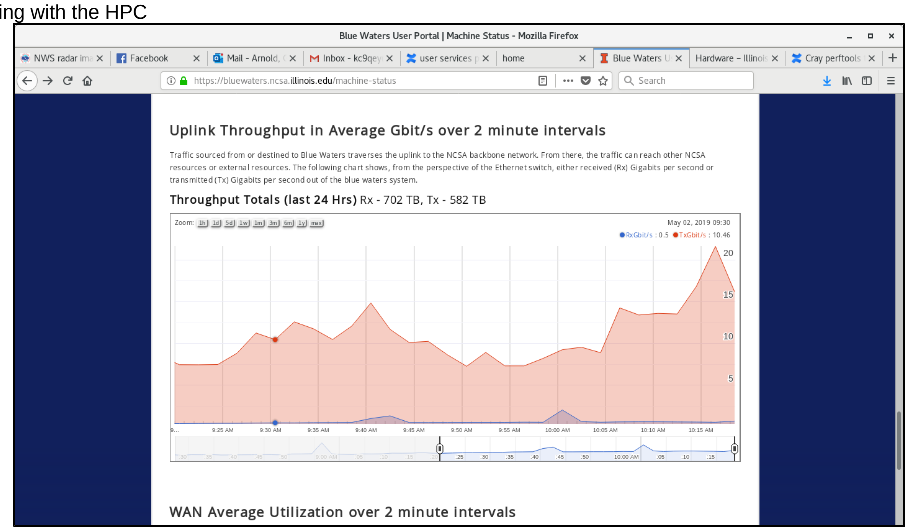
ijk-perftools.png • v.1 [Current]

Apprentice2

triples-acc.x+pat+2825784-18469t.ap2

```
PE:     5
Func:   cc_triples_.ACC_ASYNC_COPY@li.450
Start:  57.736095
End:    57.736954

ACC:            Copy    Bytes:

Start:  57.736104       In:     152183808
End:    57.761551

Ev Tag: 3909
```

  ■ interacting with the HPC



Blue Waters User Portal | Machine Status - Mozilla Firefox

https://bluewaters.ncsa.illinois.edu/machine-status

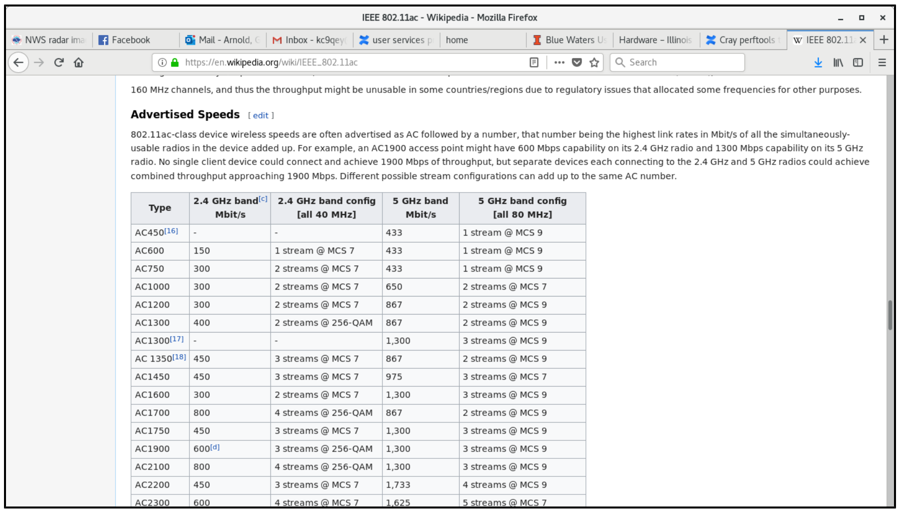
Uplink Throughput in Average Gbit/s over 2 minute intervals

Traffic sourced from or destined to Blue Waters traverses the uplink to the NCSA backbone network. From there, the traffic can reach other NCSA resources or external resources. The following chart shows, from the perspective of the Ethernet switch, either received (Rx) Gigabits per second or transmitted (Tx) Gigabits per second out of the blue waters system.

Throughput Totals (last 24 Hrs) Rx - 702 TB, Tx - 582 TB

WAN Average Utilization over 2 minute intervals

-
- data transfer bottlenecks (last mile)
  - WiFi

- 

  IEEE 802.11ac - Wikipedia - Mozilla Firefox

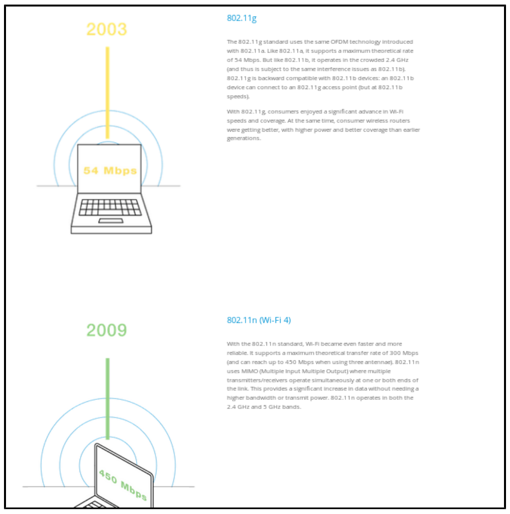
  https://en.wikipedia.org/wiki/IEEE_802.11ac

  160 MHz channels, and thus the throughput might be unusable in some countries/regions due to regulatory issues that allocated some frequencies for other purposes.

  Advertised Speeds [edit]

  802.11ac-class device wireless speeds are often advertised as AC followed by a number, that number being the highest link rates in Mbit/s of all the simultaneously-usable radios in the device added up. For example, an AC1900 access point might have 600 Mbps capability on its 2.4 GHz radio and 1300 Mbps capability on its 5 GHz radio. No single client device could connect and achieve 1900 Mbps of throughput, but separate devices each connecting to the 2.4 GHz and 5 GHz radios could achieve combined throughput approaching 1900 Mbps. Different possible stream configurations can add up to the same AC number.

Type	2.4 GHz band[c] Mbit/s	2.4 GHz band config [all 40 MHz]	5 GHz band Mbit/s	5 GHz band config [all 80 MHz]
AC450[16]	-	-	433	1 stream @ MCS 9
AC600	150	1 stream @ MCS 7	433	1 stream @ MCS 9
AC750	300	2 streams @ MCS 7	433	1 stream @ MCS 9
AC1000	300	2 streams @ MCS 7	650	2 streams @ MCS 7
AC1200	300	2 streams @ MCS 7	867	2 streams @ MCS 9
AC1300	400	2 streams @ 256-QAM	867	2 streams @ MCS 9
AC1300[17]	-	-	1,300	3 streams @ MCS 9
AC 1350[18]	450	3 streams @ MCS 7	867	2 streams @ MCS 9
AC1450	450	3 streams @ MCS 7	975	3 streams @ MCS 7
AC1600	300	2 streams @ MCS 7	1,300	3 streams @ MCS 9
AC1700	800	4 streams @ 256-QAM	867	2 streams @ MCS 9
AC1750	450	3 streams @ MCS 7	1,300	3 streams @ MCS 9
AC1900	600[d]	3 streams @ 256-QAM	1,300	3 streams @ MCS 9
AC2100	800	4 streams @ 256-QAM	1,300	3 streams @ MCS 9
AC2200	450	3 streams @ MCS 7	1,733	4 streams @ MCS 9
AC2300	600	4 streams @ MCS 7	1,625	5 streams @ MCS 7

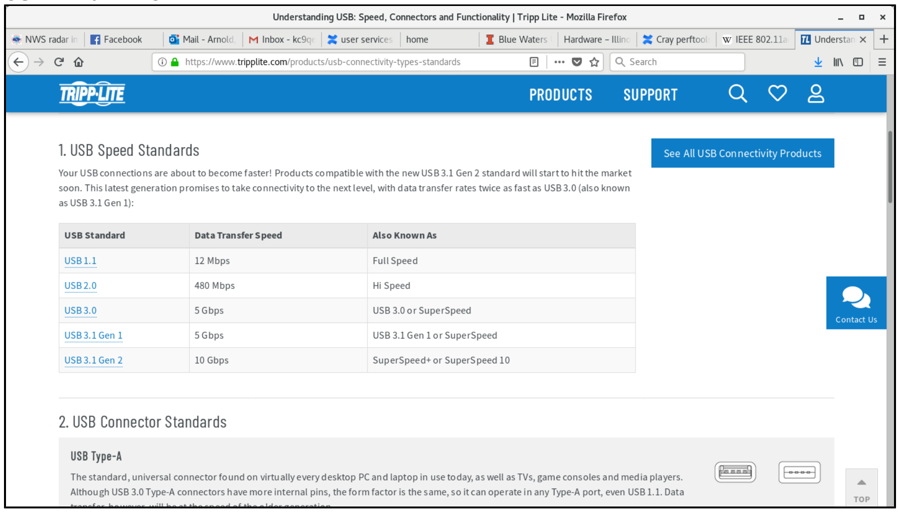
  

  802.11g

  The 802.11g standard uses the same OFDM technology introduced with 802.11a. Like 802.11a, it supports a maximum theoretical rate of 54 Mbps. But like 802.11b, it operates in the crowded 2.4 GHz (and thus is subject to the same interference issues as 802.11b). 802.11g is backward compatible with 802.11b devices: an 802.11b device can connect to an 802.11g access point (but at 802.11b speeds).

  With 802.11g, consumers enjoyed a significant advance in Wi-Fi speeds and coverage. At the same time, consumer wireless routers were getting better, with higher power and better coverage than earlier generations.

  802.11n (Wi-Fi 4)

  With the 802.11n standard, Wi-Fi became even faster and more reliable. It supports a maximum theoretical transfer rate of 300 Mbps (and can reach up to 450 Mbps when using three antennae). 802.11n uses MIMO (Multiple Input Multiple Output) where multiple transmitters/receivers operate simultaneously at one or both ends of the link. This provides a significant increase in data without needing a higher bandwidth or transmit power. 802.11n operates in both the 2.4 GHz and 5 GHz bands.
- USB-N with N < 3 ?
- 

  Understanding USB: Speed, Connectors and Functionality | Tripp Lite - Mozilla Firefox

  https://www.tripplite.com/products/usb-connectivity-types-standards

  1. USB Speed Standards

  Your USB connections are about to become faster! Products compatible with the new USB 3.1 Gen 2 standard will start to hit the market soon. This latest generation promises to take connectivity to the next level, with data transfer rates twice as fast as USB 3.0 (also known as USB 3.1 Gen 1):

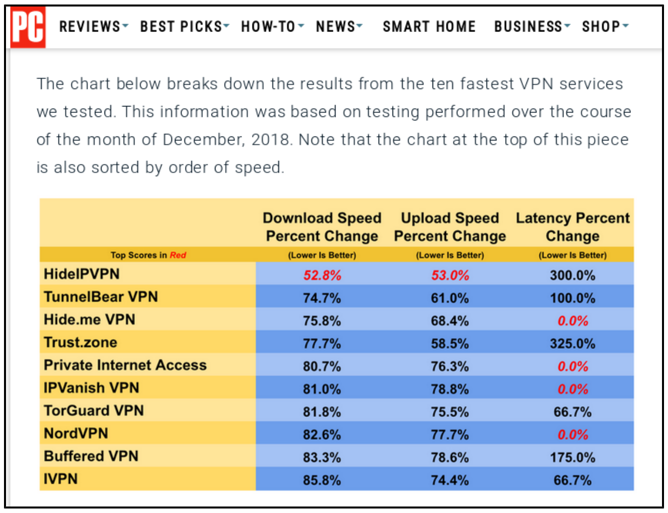
USB Standard	Data Transfer Speed	Also Known As
USB 1.1	12 Mbps	Full Speed
USB 2.0	480 Mbps	Hi Speed
USB 3.0	5 Gbps	USB 3.0 or SuperSpeed
USB 3.1 Gen 1	5 Gbps	USB 3.1 Gen 1 or SuperSpeed
USB 3.1 Gen 2	10 Gbps	SuperSpeed+ or SuperSpeed 10

  2. USB Connector Standards

  USB Type-A

  The standard, universal connector found on virtually every desktop PC and laptop in use today, as well as TVs, game consoles and media players. Although USB 3.0 Type-A connectors have more internal pins, the form factor is the same, so it can operate in any Type-A port, even USB 1.1. Data

- on a campus lan
  - gigabit ethernet (show bandwidth and latency from a test)
- on a "foreign" lan (home, coffee shop, another continent...)
  - vpn ?

- 
PC
REVIEWS
BEST PICKS
HOW-TO
NEWS
SMART HOME
BUSINESS
SHOP

The chart below breaks down the results from the ten fastest VPN services we tested. This information was based on testing performed over the course of the month of December, 2018. Note that the chart at the top of this piece is also sorted by order of speed.

	Download Speed Percent Change	Upload Speed Percent Change	Latency Percent Change
Top Scores in *Red*	(Lower Is Better)	(Lower Is Better)	(Lower Is Better)
HideIPVPN	*52.8%*	*53.0%*	300.0%
TunnelBear VPN	74.7%	61.0%	100.0%
Hide.me VPN	75.8%	68.4%	*0.0%*
Trust.zone	77.7%	58.5%	325.0%
Private Internet Access	80.7%	76.3%	*0.0%*
IPVanish VPN	81.0%	78.8%	*0.0%*
TorGuard VPN	81.8%	75.5%	66.7%
NordVPN	82.6%	77.7%	*0.0%*
Buffered VPN	83.3%	78.6%	175.0%
IVPN	85.8%	74.4%	66.7%

- performance vs security

- debugging codes and issues that arise
  - your code ?
  - system problem ?
  - Debugging on Blue Waters

References:

https://bluewaters.ncsa.illinois.edu/user-guide

https://campuscluster.illinois.edu/resources/docs/user-guide/