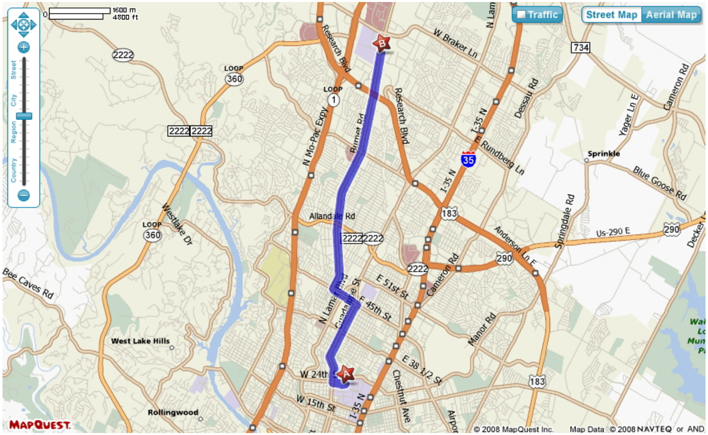
What is TACC and what can it mean for you?

Victor Eijkhout
research scientist

The what/where/why of TACC

Where is TACC?

On the J.J. Pickle campus of UT, 7.7 miles and 18 minutes north of here.



What is TACC?

- Founded in 1986 by UT System
- Part of UT Austin in 1990
- Current organization started in 2001, dozen people
- Currently 75+ and growing
- Lonestar: Dell cluster, Longhorn/Champion: IBM cluster
- Ranger, deployed in 2008, most powerful open science machine, 4th fastest in the world

You've probably heard about Ranger

Why is TACC?

The mission of the Texas Advanced Computing Center is to enable discoveries that advance science and society through the application of advanced computing technologies.

How is TACC?

- Resources & Services
  - Acquire & operate leading-edge advanced computing systems
  - Support world-class researchers with expert consulting, training, documentation
- Research & Development
  - Conduct R&D to produce new computational technologies
  - Collaborate with users to apply advanced computing techniques
- Education, Outreach, Public Relations
  - Educate the community to increase participation in advanced computing technology careers
  - Inform society about value of advanced computing technologies in improving knowledge and quality of life

Just thought I'd mention our new visualization lab

Area A: Stallion
Cosmic Renaissance
Area A: Stallion
Mantle Convection using Adaptive Mesh Refinement
Area A: Stallion
Hurricane Modeling

To business. Parallel computing

Computer organization

- A processor chip is called a ‘socket’.
- Each socket contains 2 (lonestar: dual-core) or 4 (ranger: quad-core) ‘cores’.
- The word ‘processor’ is confusing: is that the chip (socket) or processing unit (core)?
- A ‘node’ has 2 (lonestar) or 4 (ranger) cores, with shared memory.
- The nodes are connected through a network. Like ethernet, but much faster.

Sun
microsystems
Sun
Sun
Sun

Shared memory programming

If you do not need more tasks than cores on a node,
they can all access the shared memory on the node.

Pro: relative easy to program (loop-based parallelism)
Con: limited amount of parallelism

Access through OpenMP; sometimes the compiler can handle it
Some libraries can employ this

Distributed memory programming

If you need large numbers of processes,
let them all communicate through the network

Pro: lots of power at your disposal
Con: relatively hard to program (data parallel)

Hybrid approaches are possible

Distributed memory programming; the practice

- Each process has its own memory space
- The MPI (Message Passing Interface) library is used for communication
- Typical program structure: SPMD (Single Program Multiple Data)
- Ideally: user writes one code, parallel object accessed by 'handle', details taken care of by a library

‘Conveniently parallel’ programming

- Large number of completely independent sequential jobs.
- Requires powerful processors, not much of a network
- ‘Parameter sweep’

Lonestar

- 1460 nodes, 5840 cores
- 11.6Tb memory, 106+70Tb disk space
- 62Tflop peak
- UT system

Ranger

- 3936 nodes, 62,976 cores
- 123Tb memory, 1.7Pb disk space
- 579Tflop peak
- NSF (TeraGrid)

Stampede

- 217 nodes, 1736 cores (dual Intel quadcore)
- 1800Gb memory, approx 1Tb disk space
- 16 Tflop peak
- Gigabit ethernet
- UT Austin

How do clusters work?

- We have more users than cores, so
- Batch system: you submit a job, and it will get done; maybe very soon, maybe in a little while
- A bit cumbersome: no interactive input, testing may take a while

Standard parallel job file

```
#!/bin/csh
#BSUB -J sage
#BSUB -n 8
#BSUB -q development
#BSUB -o sage.o%J
#BSUB -W 0:20

# setup stuff
ibrun myprog
# post processing
```

Multiple serial job file

```
#!/bin/csh
#BSUB -J sage
#BSUB -n 8
#BSUB -q development
#BSUB -o sage.o%J
#BSUB -W 0:20

module load launcher
setenv EXECUTABLE     $TACC_LAUNCHER_DIR/launcher
setenv CONTROL_FILE   paramlist

pam -g 1 parametric_wrapper $EXECUTABLE $CONTROL_FILE
```

control file

```
myprog 1
myprog 2
myprog 3
myprog 5
myprog 8
myprog 13
myprog 21
myprog 34
```

Interactive runs

Running a serious program on the login nodes is frowned upon

Interactive batch queue:

```
bsub -I -n 1 -q development -W 0:05 ibrun myprog arg1 arg2
```

Use of sage in batch scripts:

```
sage -c cmd
```

More information

eijkhout@tacc.utexas.edu

https://portal.tacc.utexas.edu/

http://www.tacc.utexas.edu/services/userguides/