AMath 483/583 — Lecture 15

Outline:

- Cloud computing on Amazon Web Services
- Timing Fortran codes

Reading:

- class notes: AWS section
- class notes: Timing code section

Notes:

Cloud Computing

- Computing resources as a “utility”.
- Rent computer time by the hour as needed.
- Avoid buying computers that will sit idle most of the time.
- Provide a computing platform with necessary software pre-installed.

Notes:

Amazon Web Services (AWS)

- Elastic Cloud Computing (ECC)
- Scalable Storage (S3)
- Many other services: aws.amazon.com

Several instance types are available.

- Free usage tier: Can run one “micro-instance” free for a year. (1 EC2 compute unit, 613 MiB memory)
- C1, High CPU medium instance: 2 cores with 5 EC2 units, 1.7 GiB memory.
- See the Price list

Notes:

Amazon Machine Images (AMIs)

Choice of virtual machines to use when launching an instance.

See the List of basic AMIs

For this class, and AMI is available with much of the software needed.

https://console.aws.amazon.com/ec2/home?region=us-west-2#launchAmi=ami-b47feb84

Notes:

AWS demo

See the instructions in the class notes: AWS section

Note:

- You will need to create an account
- and create a key-pair
- and a security group
- On a Mac, for X-window forwarding you need to install Xcode
- On Windows, you need an ssh client such as putty
  For X-window forwarding you also need xming

Notes:

AMath 483/583 — Lecture 15

Outline:

- Timing Fortran codes

Reading:

- class notes: Timing code section
- `$UWHPSC/codes/fortran/timings.f90`
- `$UWHPSC/codes/openmp/timings.f90`

Notes:

Determining CPU and execution time

Unix time command, e.g.

```
$ time ./a.out
<output from code>

real    0m5.279s
user    0m1.915s
sys     0m0.006s
```

Means the elapsed (wall clock) time was 5.279 seconds,

CPU time dedicated to your code was $\approx$ 1.915 seconds.

System time $\approx$ 0.006 seconds.

Doesn't allow examining parts of code, not always very accurate.

Note that timing small codes can be deceptive

Notes:

Fortran timing utilities

`system_clock`: elapsed time between 2 calls.

`cpu_time`: CPU time used between 2 calls.

See class notes: Timing code

Notes: