CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Performance analysis with CrayPat

CSCS

Instrumenting your code

-module load perftools
-make
-pat_build main

This will produce an executable with extension ***+pat**

To do:

- **Fix the grid size and run on different number of cores**
- Plot timing, and % of MPI communication

- **Pick a core count , increase the grid size**

- **Increase them in tandem**

Running the instrumented executable

- **As before!**

```
aprun –n 2  main+pat 128 128  100 0.01
```

This will produce a file with extension *.xf
→ **pat_report <file.xf>**

Now you will also have a file with extension .ap2

→ **app2 *.ap2**

Speedup and efficiency

- **We define the speedup to be how much faster a code is on N processors compared to one processor**
- speedup is a measure of reduced time-to-solution
- We define the efficiency to be the speedup on N processors divided by the number of processors
- efficiency is a measure of resource utilisation
- efficiency is often expressed as a percentage
- **If T_1 is the time taken to run on 1 processor and T_N is the time taken to run on N processors then we have**

$$Speedup = \frac{T_1}{T_N}$$

$$Efficiency = \frac{Speedup}{N}$$

Strong vs Weak Scaling

- If you keep the problem size the same as you change the number of tasks then we call this strong scaling
- If you change the problem size in proportion to the number of tasks then we call this weak scaling
- Strong scaling is typically harder to achieve than weak scaling

Non- cray performance tools

- **Score-p for instrumentation (**http://www.vi-hps.org/projects/score-p/**)**
- **Scalasca (**http://www.scalasca.org/**)**
- **Vampir (**http://www.vampir.eu/**)**
- **Tau (**http://www.cs.uoregon.edu/research/tau/home.php**)**

Thank you for your attention.