CSEG Coffee Talk

Load Balancing CESM

Jim Edwards and Sheri Mickelson

Friday, October 3, 2014

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Intra vs Inter component

Intra: within a single model component
Inter: between model components

Intracomponent Load Balancing

- CAM
  - phys_loadbalance
  - pcols (1-256+)
- POP
  - POP_NX_BLOCKS,POP_NY_BLOCKS
- CICE
  - CICE_BLCKX, CICE_BLCKY
- CLM
  - nsegspc

Intracomponent Load Balancing

```
***** GLOBAL STATISTICS (  2048 MPI TASKS) *****
'count' is cumulative. All other stats are max/min
name                          processes  threads            count     walltotal    wallmax (proc   thrd  )   wallmin (proc   thrd  )
DRIVER_INIT                        2048     2048 2.048000e+03   2.449308e+05   119.598 (      0         0)   119.595 (   209         0)
driver_init_comps                  2048     2048 2.048000e+03   2.321493e+05   113.354 (   960         0)   113.349 (     0         0)
phys_grid_init                     2048     2048 2.048000e+03   5.036139e+02        0.246 (   0         0)        0.246 (  1210       0)
driver_init_maps                   2048     2048 2.048000e+03   3.176462e+03        1.551 (   881       0)        1.550 (   0         0)
driver_init_atminit                2048     2048 1.228800e+04   3.109908e+01        0.016 (  1330       0)        0.015 (   1         0)
stepon_run1                        2048     2048 1.433600e+04   2.281145e+01        0.016 (   123       0)        0.007 (  1445       0)
d_p_coupling                       2048     2048 1.433600e+04   2.271088e+01        0.016 (   123       0)        0.007 (  1445       0)
UniquePoints                       2048     2048 1.433600e+04   2.867161e+00        0.002 (   123       0)        0.001 (  1445       0)
dpcopy                             2048     2048 1.433600e+04   2.452174e+00        0.002 (   123       0)        0.001 (  1445       0)
derived_phys                       2048     2048 1.433600e+04   1.679802e+01        0.012 (   123       0)        0.005 (  1445       0)
phys_run1                          2048     2048 1.433600e+04   1.229928e+04        6.754 (  1638       0)        4.578 (   240       0)
physpkg_st1                        2048     2048 2.867200e+04   9.207874e+03        4.527 (  1585       0)        3.411 (   0         0)
chk_en_gmean                       2048     2048 1.433600e+04   2.241349e+03        1.125 (  1995       0)        0.008 (   0         0)
gmean_fixed_repro                  2048     2048 1.433600e+04   2.241101e+03        1.125 (  1995       0)        0.007 (   0         0)
shr_reprosum_int                   2048     2048 1.433600e+04   2.240856e+03        1.124 (  1995       0)        0.007 (   0         0)
repro_sum_loopa                    2048     5400 3.780000e+04   1.352077e-01        0.000 (  1632       0)        0.000 (  1445       0)
repro_sum_allr_minmax              2048     2048 1.433600e+04   2.239212e+03        1.124 (  1995       0)        0.006 (   0         0)
repro_sum_loopb                    2048     5400 3.780000e+04   6.620559e-01        0.000 (   123       0)        0.000 (  1445       0)
repro_sum_allr_i4                  2048     2048 1.433600e+04   3.841594e-01        0.000 (  1445       0)        0.000 (   1         0)
advance_trcdata                    2048     2048 2.580480e+05   6.947327e+03        3.396 (   123       0)        3.388 (  1445       0)
read_next_trcdata                  2048     2048 3.686400e+04   6.905339e+03        3.372 (  1445       0)        3.371 (   123       0)
```

Strategic Parallel Optimization for Computing (SPOC)

CISL sponsored initiative to improve performance of NCAR codes including CESM on current and future hardware.

- Improve metrics and performance measurement.
- Identify and eliminate low level performance bottlenecks
- Prepare CESM for anticipated future systems.

Plans to improve performance metrics in CESM

- Add runtime indication of load balance and/or performance alarms
- Improved output of performance metrics
- Develop a CESM tuning guide

CESM performance team

Jim Edwards (CSEG)
Stefan Muszala (CSEG)
Rory Kelly (CISL)

Intercomponent Load Balancing

Why Load Balance?

Bad ….

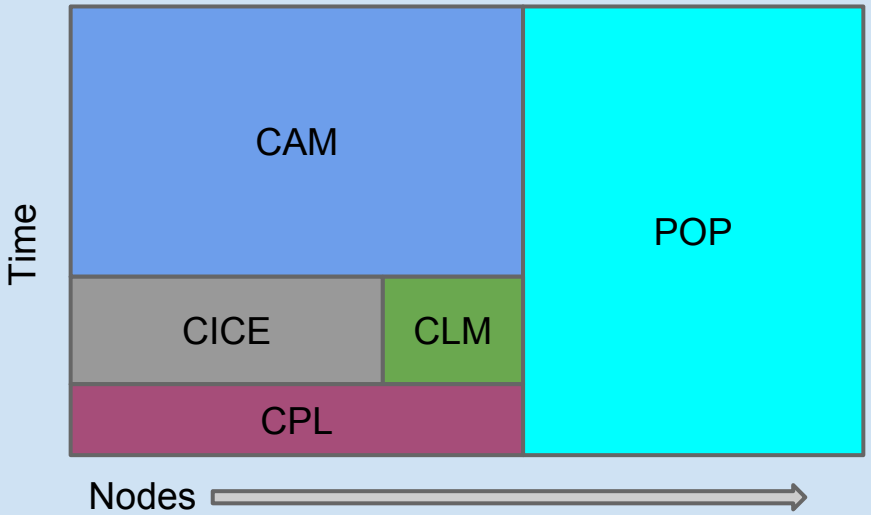
```
Model Throughput: 2.47 simulated_years/day
    TOT Run Time:95.983 seconds/mday
    LND Run Time: 0.663 seconds/mday
    ROF Run Time: 0.107 seconds/mday
    ICE Run Time: 4.274 seconds/mday
    ATM Run Time:37.790 seconds/mday
    OCN Run Time: 9.754 seconds/mday
    GLC Run Time: 0.000 seconds/mday
    WAV Run Time: 0.000 seconds/mday
    CPL Run Time: 1.624 seconds/mday
    CPL COMM Time:0.243 seconds/mday
```

Good ….

```
Model Throughput:  6.40   simulated_years/day
    TOT Run Time: 36.966 seconds/mday
    LND Run Time:  3.356 seconds/mday
    ROF Run Time:  0.122 seconds/mday
    ICE Run Time:  3.169 seconds/mday
    ATM Run Time: 29.270 seconds/mday
    OCN Run Time: 29.051 seconds/mday
    GLC Run Time:  0.000 seconds/mday
    WAV Run Time:  0.000 seconds/mday
    CPL Run Time:  4.434 seconds/mday
    CPL COMM Time:10.087 seconds/mday
```

More Science for the Same Cost

Basic Concept of Load Balancing a B Compset



Coupling Rules:

- CAM and POP run concurrently
- CICE and CLM run concurrently
- CAM runs sequentially with CICE and CLM

Solve for:

(CAM + (CICE == LND)) == POP

Why the throughput is better

Bad ….

```
Model Throughput: 2.47 simulated_years/day
    TOT Run Time:95.983 seconds/mday
    LND Run Time: 0.663 seconds/mday
    ROF Run Time: 0.107 seconds/mday
    ICE Run Time: 4.274 seconds/mday
    ATM Run Time:37.790 seconds/mday
    OCN Run Time: 9.754 seconds/mday
    GLC Run Time: 0.000 seconds/mday
    WAV Run Time: 0.000 seconds/mday
    CPL Run Time: 1.624 seconds/mday
    CPL COMM Time:0.243 seconds/mday
```

Good ….

```
Model Throughput:  6.40   simulated_years/day
    TOT Run Time: 36.966 seconds/mday
    LND Run Time:  3.356 seconds/mday
    ROF Run Time:  0.122 seconds/mday
    ICE Run Time:  3.169 seconds/mday
    ATM Run Time: 29.270 seconds/mday
    OCN Run Time: 29.051 seconds/mday
    GLC Run Time:  0.000 seconds/mday
    WAV Run Time:  0.000 seconds/mday
    CPL Run Time:  4.434 seconds/mday
    CPL COMM Time:10.087 seconds/mday
```

Why the throughput is better

Bad ….

```
Model Throughput: 2.47 simulated_years/day
    TOT Run Time:95.983 seconds/mday
    LND Run Time: 0.663 seconds/mday
    ROF Run Time: 0.107 seconds/mday
    ICE Run Time: 4.274 seconds/mday
    ATM Run Time:37.790 seconds/mday
    OCN Run Time: 9.754 seconds/mday
    GLC Run Time: 0.000 seconds/mday
    WAV Run Time: 0.000 seconds/mday
    CPL Run Time: 1.624 seconds/mday
    CPL COMM Time:0.243 seconds/mday
```

Good ….

```
Model Throughput:  6.40   simulated_years/day
    TOT Run Time: 36.966 seconds/mday
    LND Run Time:  3.356 seconds/mday
    ROF Run Time:  0.122 seconds/mday
    ICE Run Time:  3.169 seconds/mday
    ATM Run Time: 29.270 seconds/mday
    OCN Run Time: 29.051 seconds/mday
    GLC Run Time:  0.000 seconds/mday
    WAV Run Time:  0.000 seconds/mday
    CPL Run Time:  4.434 seconds/mday
    CPL COMM Time:10.087 seconds/mday
```

Walking through a load balancing example

Balancing a BC5.ne30_g16 case on mira

Automating the Process with a Load Balancing Tool

- Was introduced to the code base in cesm1_3_alpha13a
- Found within the tools directory in the CESM developer's repository
- Creates scale tests, builds, and runs them. Then it will solve the scaling curves to fit a target PE count and will give you a load balanced layout

First Step in Running the Load Balancing Tool

Edit global_variables.csh

```
################################################
# Set case variables
################################################
setenv ccsmsrc /$HOME/cesm1_3_beta12/
setenv res ne30_g16
setenv compset B1850C5
setenv mach mira_ibm
setenv casedir $SCRATCH/tests/cesm_timing_tests
setenv casestr  _B1850C5_ne30_g16__
setenv submit " "
setenv run_len 5

# Select either FV or SE below
#setenv DYCORE FV
setenv DYCORE SE

################################################
# Set the location of the load balancing results
################################################
setenv results_dir $HOME/cesm_load_balancing/results/
```

```
################################################
# Set the test layouts to produce the scaling curves
################################################
setenv NTHRDS_VAL 8

# Set the Task Counts
setenv TASK_ATM  "128,256,512,1024"
setenv TASK_LND  "128,256,512,1024"
setenv TASK_ROF  "128,256,512,1024"
setenv TASK_ICE  "128,256,512,1024"
setenv TASK_OCN  "128,256,512,1024"
setenv TASK_CPL  "128,256,512,1024"
setenv TASK_WAV  "128,256,512,1024"
setenv TASK_GLC  "1,1,1,1"

# Set Root Locations
setenv ROOT_ATM  "0,0,0,0,0"
setenv ROOT_LND  "0,0,0,0,0"
setenv ROOT_ROF  "0,0,0,0,0"
setenv ROOT_ICE  "0,0,0,0,0"
setenv ROOT_OCN  "0,0,0,0,0"
setenv ROOT_CPL  "0,0,0,0,0"
setenv ROOT_WAV  "0,0,0,0,0"
setenv ROOT_GLC  "0,0,0,0,0"

################################################
# Set the target task counts (ATM(LND+ICE) + OCN)
################################################
setenv TARGET_TASKS "2048"
```

Next steps in running the load balancing tool

After editing global_variables.csh

- Execute ./run_first.csh

Once scaling runs complete

- Execute ./run_second.csh

After you run the second script, your results directory will contain several files. The most important are ….

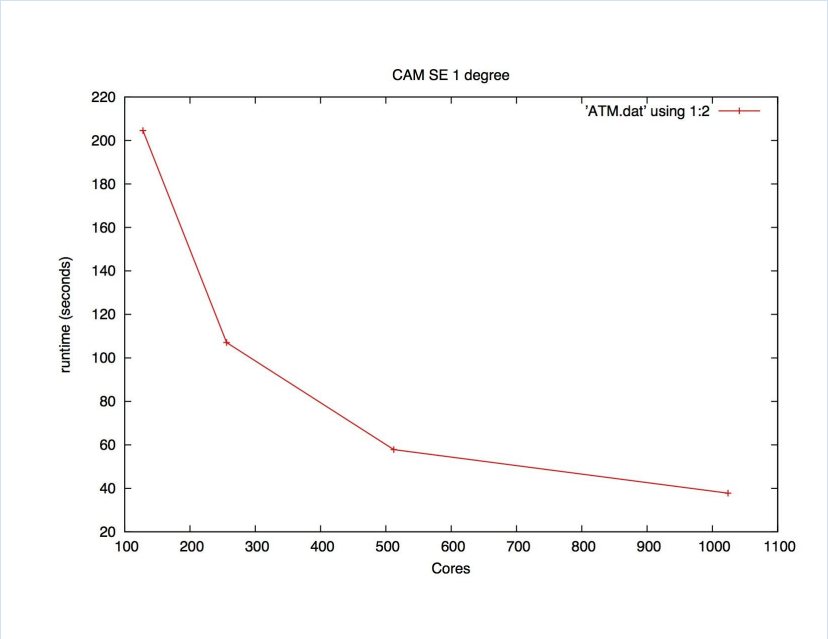
atm.gif
cost.gif
cpl.gif
ice.gif
lnd.gif
ocn.gif
rof.gif
tp.gif

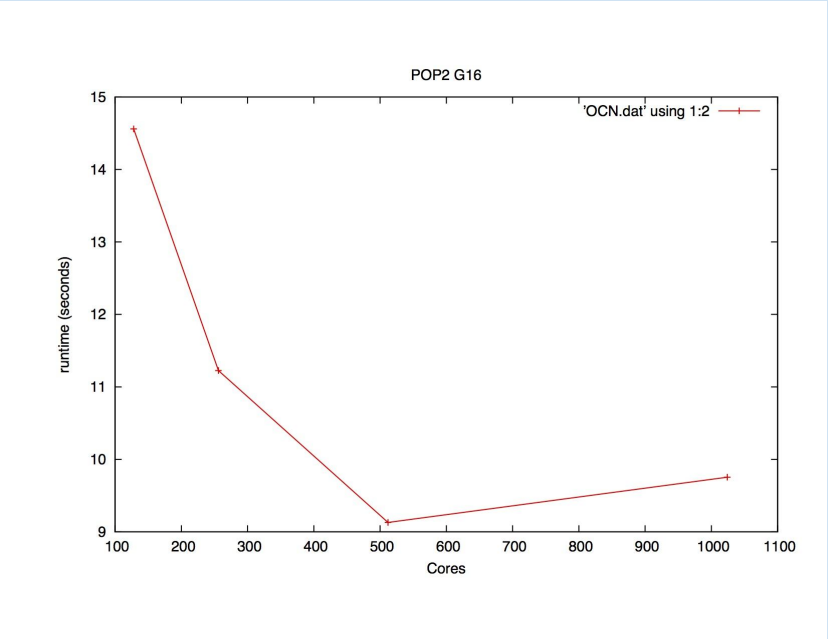
} Scaling Plots

minmax_times_*.txt (one file per target task count)
env_mach_pes_*.xml (one file per target task count)

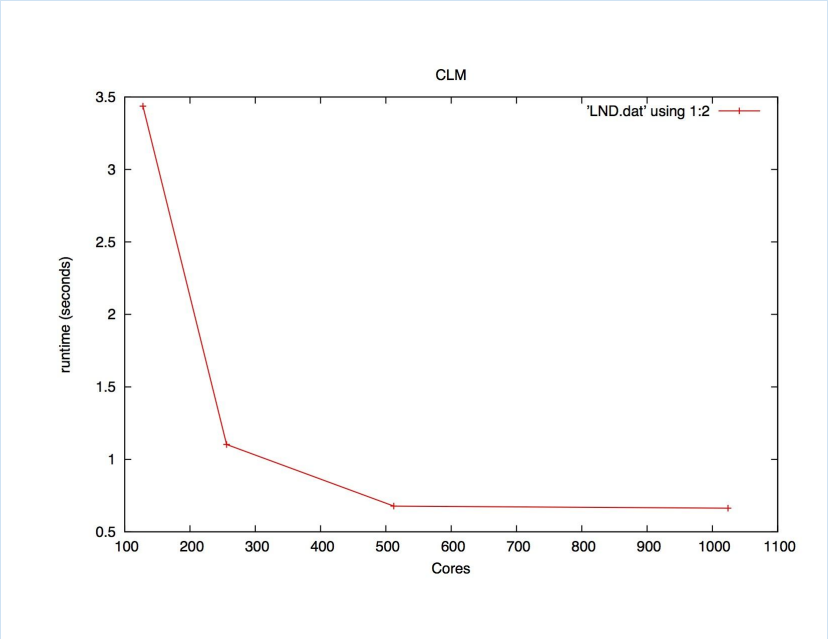
} Suggested layouts

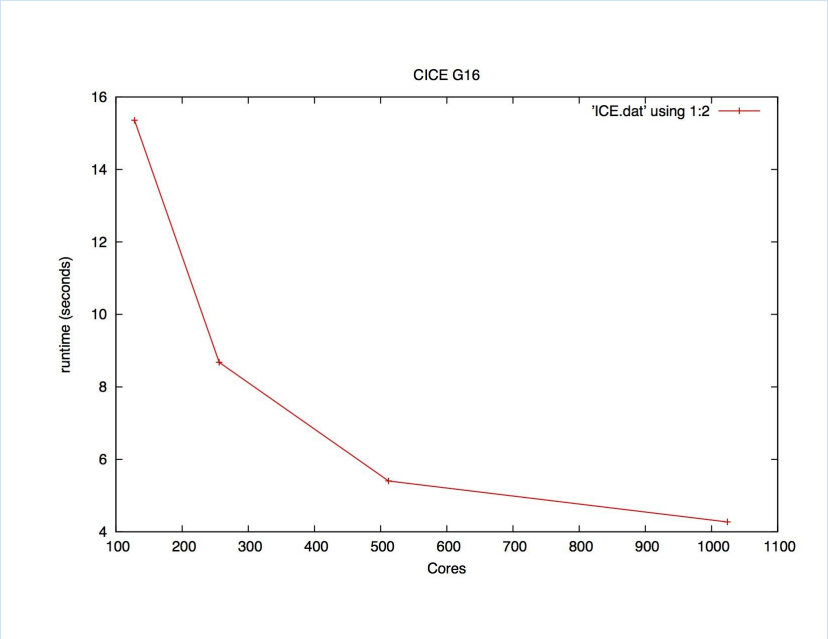
Look at the Scaling Plots to make sure the curves look okay





Look at the Scaling Plots to make sure the curves look okay





Suggested layouts are found in two places …

(both sets will be found in your results directory)

1.) **Suggested layouts and best guess at run times are found in minmax_times_*.txt**

example:

```
##Comp    NTASKS seconds/model-day
lnd         50       4.196
ice       1048       4.274
atm       1098      37.568
ocn        253      12.041
Total esimated total time: 41.841
```

2.) **And new env_mach_pes_*.xml files will be produced. There will be a new file for each target task count you requested in the global_variables.csh file**

Test the Suggested layout ….

Timing results from the ccsm_timing.* file:

```
Model Throughput:         4.98   simulated_years/day
TOT Run Time:     237.443 seconds       47.489 seconds/mday         4.98 myears/wday
LND Run Time:      28.553 seconds        5.711 seconds/mday        41.45 myears/wday
ROF Run Time:       1.407 seconds        0.281 seconds/mday       841.20 myears/wday
ICE Run Time:      25.277 seconds        5.055 seconds/mday        46.82 myears/wday
ATM Run Time:     182.851 seconds       36.570 seconds/mday         6.47 myears/wday
OCN Run Time:      56.924 seconds       11.385 seconds/mday        20.79 myears/wday
GLC Run Time:       0.000 seconds        0.000 seconds/mday         0.00 myears/wday
WAV Run Time:       0.000 seconds        0.000 seconds/mday         0.00 myears/wday
CPL Run Time:      30.601 seconds        6.120 seconds/mday        38.68 myears/wday
CPL COMM Time:     62.764 seconds       12.553 seconds/mday        18.86 myears/wday
```

After You Run the Load Balancing Tool ….

Timing results from the ccsm_timing.* file:

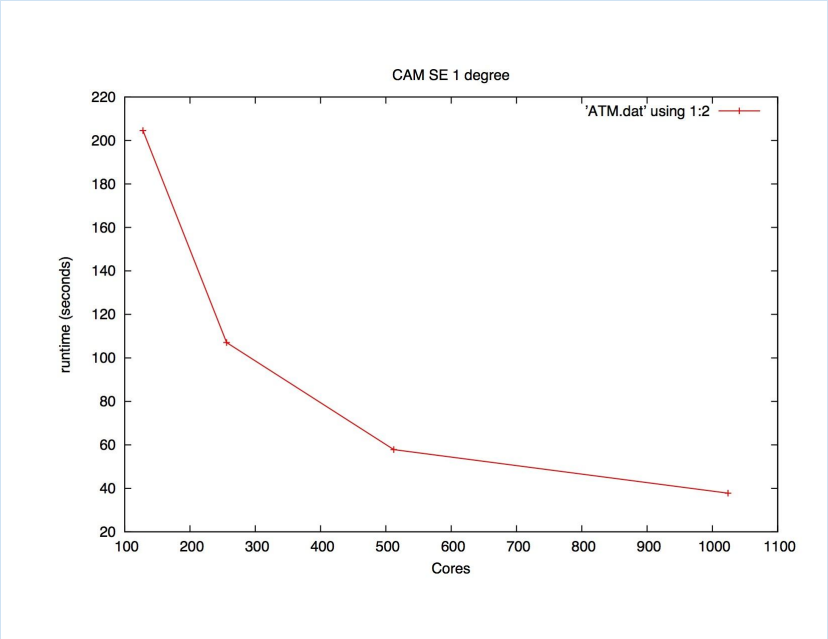
```
Model Throughput:         4.98   simulated_years/day
TOT Run Time:      237.443 seconds       47.489 seconds/mday         4.98 myears/wday
LND Run Time:       28.553 seconds        5.711 seconds/mday        41.45 myears/wday
ROF Run Time:        1.407 seconds        0.281 seconds/mday       841.20 myears/wday
ICE Run Time:       25.277 seconds        5.055 seconds/mday        46.82 myears/wday
ATM Run Time:      182.851 seconds       36.570 seconds/mday         6.47 myears/wday
OCN Run Time:       56.924 seconds       11.385 seconds/mday        20.79 myears/wday
GLC Run Time:        0.000 seconds        0.000 seconds/mday         0.00 myears/wday
WAV Run Time:        0.000 seconds        0.000 seconds/mday         0.00 myears/wday
CPL Run Time:       30.601 seconds        6.120 seconds/mday        38.68 myears/wday
CPL COMM Time:      62.764 seconds       12.553 seconds/mday        18.86 myears/wday
```

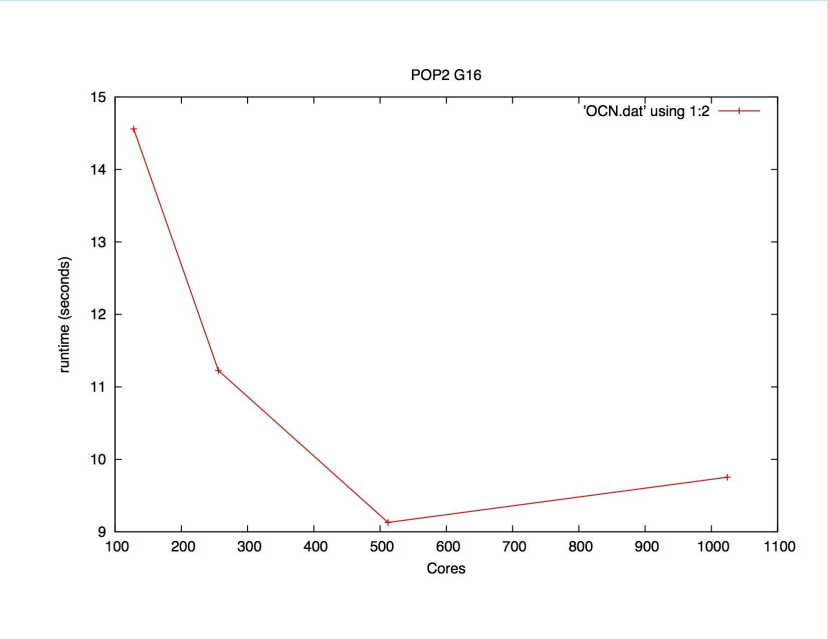
After You Run the Load Balancing Tool ….

Timing results from the ccsm_timing.* file:

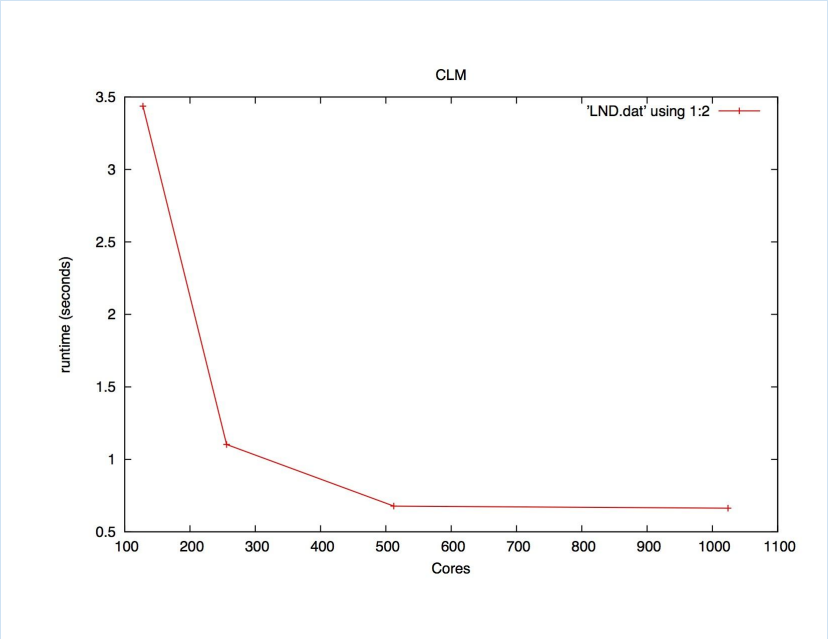
```
Model Throughput:         4.98   simulated_years/day

TOT Run Time:      237.443 seconds       47.489 seconds/mday         4.98 myears/wday
LND Run Time:       28.553 seconds        5.711 seconds/mday        41.45 myears/wday
ROF Run Time:        1.407 seconds        0.281 seconds/mday       841.20 myears/wday
ICE Run Time:       25.277 seconds        5.055 seconds/mday        46.82 myears/wday
ATM Run Time:      182.851 seconds       36.570 seconds/mday         6.47 myears/wday
OCN Run Time:       56.924 seconds       11.385 seconds/mday        20.79 myears/wday
GLC Run Time:        0.000 seconds        0.000 seconds/mday         0.00 myears/wday
WAV Run Time:        0.000 seconds        0.000 seconds/mday         0.00 myears/wday
CPL Run Time:       30.601 seconds        6.120 seconds/mday        38.68 myears/wday
CPL COMM Time:      62.764 seconds       12.553 seconds/mday        18.86 myears/wday
```

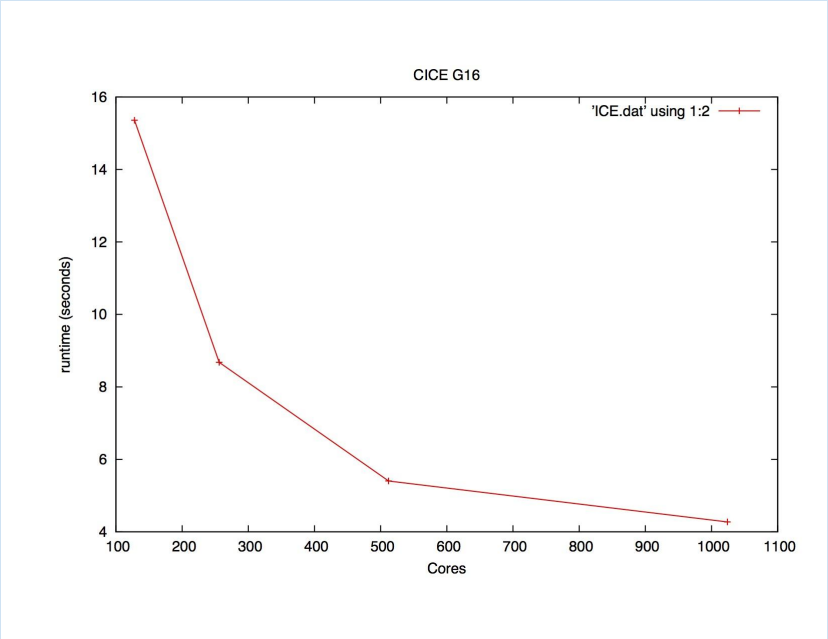
Look at the scaling plots again to get ideas for improvement





Look at the scaling plots again to get ideas for improvement





Hand Tuning ...

Suggested layout from the tool ….

```
##Comp     NTASKS seconds/model-day
lnd         50       4.196
ice       1048       4.274
atm       1098      37.568
ocn        253      12.041
Total esimated total time:      41.841
```

Try this new layout ….

```
##Comp     NTASKS
lnd         144
ice        2456
atm        2600
ocn         192
```

Results from the tuned layout

Results from the suggested layout

Model Throughput:4.98 simulated_years/day

```
TOT Run Time: 47.489 seconds/mday
LND Run Time:  5.711 seconds/mday
ROF Run Time:  0.281 seconds/mday
ICE Run Time:  5.055 seconds/mday
ATM Run Time: 36.570 seconds/mday
OCN Run Time: 11.385 seconds/mday
GLC Run Time:  0.000 seconds/mday
WAV Run Time:  0.000 seconds/mday
CPL Run Time:  6.120 seconds/mday
CPL COMM Time:12.553 seconds/mday
```

Results from hand tuning

Model Throughput:6.13 simulated_years/day

```
TOT Run Time: 38.631 seconds/mday
LND Run Time:  3.044 seconds/mday
ROF Run Time:  0.146 seconds/mday
ICE Run Time:  3.141 seconds/mday
ATM Run Time: 31.994 seconds/mday
OCN Run Time: 11.892 seconds/mday
GLC Run Time:  0.000 seconds/mday
WAV Run Time:  0.000 seconds/mday
CPL Run Time:  4.240 seconds/mday
CPL COMM Time:12.132 seconds/mday
```

How to Set Up Layouts for Single Component Compsets

- Run the Load Balancing Tool
- Look at the Scaling Curves to get performance characteristics
- Ignore load balancing results and set all components to use same task counts

We would like to thank

- ALCF for the time used on mira and cetus
- Yuri Alexeev for his help in developing the load balancing tool