An Introduction to Load Balancing CCSM3 Components

CCSM Workshop
June 23, 2005
Breckenridge, CO

The National Center for Atmospheric Research is funded by the National Science Foundation.
Overview

- CCSM3 Introduction
- Load Balance Introduction
- Examples
- Tools vs Log Files

- To learn this material, iteration is required
  - Read, try it, repeat
- This needs to be an interactive session
CCSM3 Introduction

- CCSM, the Community Climate System Model, is a coupled model for simulating the earth's climate system.
  - Developed at NCAR with significant collaborations with DOE, NASA and the university community
- Components in CCSM3 include
  - Atmospheric Model - CAM 3.0
    - T31: (48 x 96 x 26)  T42: (64 x 128 x 26)  T85: (128 x 256 x 26)
  - Ocean Model - modified version of POP 1.4.3
    - 3 degree: (100 x 116 x 25)  1 degree: (320 x 384 x 40)
  - Sea Ice Model - CSIM5 - grid matches ocean
  - Land Model - CLM3 - grid matches atmosphere
  - Coupler - CPL6
CCSM Hub and Spoke

atm

Ind

ocn

cpl

ice

CCSM Hub and Spoke
Performance Metrics

• Raw Performance: Simulated years per wall clock day
  - Capability: Optimize for single job maximum performance
• Performance Efficiency: Simulated years per wall clock day per cpu
  - Capacity: Optimize for system aggregate throughput
Two Kinds of “Load Balancing”

✓ CCSM load balancing: assigning right number of processors for each component

- Classic load balancing: moving processing around to even out execution times
The CCSM MPMD Balancing Act

• Each component has different scaling attributes in part based on different grid sizes

• System architecture/configuration constraints
  - Node size
  - Queue parameters
## Load Balancing Example

<table>
<thead>
<tr>
<th>T31x3</th>
<th>OCN</th>
<th>ATM</th>
<th>ICE</th>
<th>LND</th>
<th>CPL</th>
<th>Tot</th>
<th>Yrs/Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>4</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>40</td>
<td>20.76</td>
</tr>
<tr>
<td>Case 2</td>
<td>2</td>
<td>16</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>36</td>
<td>22.12</td>
</tr>
</tbody>
</table>

Case 2 used fewer processors and got better performance.
CCSM3 Process Flow

OCN

ATM

ICE

LND

CPL

CPL sending data to component (state 1) [receive]
CPL receiving data from component (state 3) [send]
Component processing data (state 2) [rec to send]
Component processing (state 4) [send to rec]
CCSM3 Process Flow

- **OCN**
- **ATM**
- **ICE**
- **LND**
- **CPL**

**CPL sending data to component (state 1) [receive]**

**Component processing data (state 2) [rec to send]**

**Component processing (state 4) [send to rec]**

**CPL receiving data from component (state 3) [send]**

**Once per day**

**Once per hour**
CPL Log File Timers

(shr_timer_print_all) print all timing info:
(shr_timer_print) timer  1:       0 calls,     0.000s, id: t01 - startup initialization
(shr_timer_print) timer  2:       1 calls,    819.242s, id: t02 - main integration
(shr_timer_print) timer  3:    240 calls,     0.321s, id: t03
(shr_timer_print) timer  4:    240 calls,     2.777s, id: t04
(shr_timer_print) timer  5:    240 calls,    4.692s, id: t05
(shr_timer_print) timer  6:    240 calls,     0.001s, id: t06
(shr_timer_print) timer  7:    240 calls,    15.205s, id: t07
(shr_timer_print) timer  8:    240 calls,     0.589s, id: t08
(shr_timer_print) timer  9:    240 calls,     4.596s, id: t09
(shr_timer_print) timer 10:    240 calls,    1.653s, id: t10
(shr_timer_print) timer 11:    240 calls,    4.229s, id: t11
(shr_timer_print) timer 12:    240 calls,    1.899s, id: t12
(shr_timer_print) timer 13:    240 calls,    5.137s, id: t13
(shr_timer_print) timer 14:    240 calls,    63.795s, id: t14
(shr_timer_print) timer 15:    240 calls,    3.649s, id: t15
(shr_timer_print) timer 16:    240 calls,   22.181s, id: t16
(shr_timer_print) timer 17:    240 calls,    8.407s, id: t17
(shr_timer_print) timer 18:    240 calls,    5.114s, id: t18
(shr_timer_print) timer 19:    240 calls,     0.001s, id: t19
(shr_timer_print) timer 20:    240 calls,   16.732s, id: t20
(shr_timer_print) timer 21:    240 calls,    7.187s, id: t21
(shr_timer_print) timer 22:    240 calls,   61.027s, id: t22
(shr_timer_print) timer 23:    240 calls,   16.389s, id: t23
(shr_timer_print) timer 24:    240 calls,     0.263s, id: t24
(shr_timer_print) timer 25:    240 calls,   570.794s, id: t25
(shr_timer_print) timer 26:    240 calls,    5.114s, id: t25
CPL Log File “avg dt”

- Can “tail -f” to watch progress of running job
  - (tStamp_write) cpl model date 0532-04-30 00000s wall clock 2005-06-22 10:19:00 avg dt 54s dt 56s
  - (tStamp_write) cpl model date 0532-05-01 00000s wall clock 2005-06-22 10:19:59 avg dt 54s dt 60s
  - (tStamp_write) cpl model date 0532-05-02 00000s wall clock 2005-06-22 10:20:55 avg dt 54s dt 56s
  - (tStamp_write) cpl model date 0532-05-03 00000s wall clock 2005-06-22 10:21:50 avg dt 54s dt 54s
  - (tStamp_write) cpl model date 0532-05-04 00000s wall clock 2005-06-22 10:22:44 avg dt 54s dt 54s
  - (tStamp_write) cpl model date 0532-05-05 00000s wall clock 2005-06-22 10:23:39 avg dt 54s dt 55s
  - (tStamp_write) cpl model date 0532-05-06 00000s wall clock 2005-06-22 10:24:35 avg dt 54s dt 56s
  - (tStamp_write) cpl model date 0532-05-07 00000s wall clock 2005-06-22 10:25:34 avg dt 54s dt 59s
  - (tStamp_write) cpl model date 0532-05-08 00000s wall clock 2005-06-22 10:26:31 avg dt 54s dt 57s
  - (tStamp_write) cpl model date 0532-05-09 00000s wall clock 2005-06-22 10:27:26 avg dt 54s dt 55s

- Can see dramatic variation within run
  - Seasonal or longer changes
  - System issues
  - Min, max, mean, mode
  - Can see how fast it should run
How Bad Can It Be ("avg dt")?

• 5x impact shown in this case! Can be worse!

• Example: min 10, mode 12, mean 18, max 410
# CSIM Log File Timers

<table>
<thead>
<tr>
<th>Timer number</th>
<th>Description</th>
<th>Total</th>
<th>min/max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Total</td>
<td>1133.70 seconds</td>
<td>1133.70 1133.70</td>
</tr>
<tr>
<td>1</td>
<td>TimeLoop</td>
<td>827.27 seconds</td>
<td>827.27 827.27</td>
</tr>
<tr>
<td>2</td>
<td>Dynamics</td>
<td>215.45 seconds</td>
<td>215.45 215.45</td>
</tr>
<tr>
<td>3</td>
<td>Advecln</td>
<td>64.28 seconds</td>
<td>64.28 64.28</td>
</tr>
<tr>
<td>4</td>
<td>Column</td>
<td>77.08 seconds</td>
<td>77.08 77.08</td>
</tr>
<tr>
<td>5</td>
<td>Thermo</td>
<td>56.31 seconds</td>
<td>56.31 56.31</td>
</tr>
<tr>
<td>6</td>
<td>Ridging</td>
<td>3.96 seconds</td>
<td>3.96 3.96</td>
</tr>
<tr>
<td>7</td>
<td>Cat Conv</td>
<td>9.75 seconds</td>
<td>9.75 9.75</td>
</tr>
<tr>
<td>8</td>
<td>Coupling</td>
<td>449.50 seconds</td>
<td>449.50 449.50</td>
</tr>
<tr>
<td>9</td>
<td>ReadWrit</td>
<td>4.44 seconds</td>
<td>4.44 4.44</td>
</tr>
<tr>
<td>10</td>
<td>Bound</td>
<td>7.40 seconds</td>
<td>7.40 7.40</td>
</tr>
<tr>
<td>11</td>
<td>Pre-cpl</td>
<td>0.00 seconds</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>12</td>
<td>MPI-send</td>
<td>15.22 seconds</td>
<td>15.22 15.22</td>
</tr>
<tr>
<td>13</td>
<td>MPI-recv</td>
<td>434.16 seconds</td>
<td>434.16 434.16</td>
</tr>
<tr>
<td>14</td>
<td>Snd-&gt;Rcv</td>
<td>323.09 seconds</td>
<td>323.09 323.09</td>
</tr>
<tr>
<td>15</td>
<td>Rcv-&gt;Snd</td>
<td>54.79 seconds</td>
<td>54.79 54.79</td>
</tr>
<tr>
<td>16</td>
<td>Cpl-recv</td>
<td>428.28 seconds</td>
<td>428.28 428.28</td>
</tr>
<tr>
<td>17</td>
<td>CR-unpck</td>
<td>2.16 seconds</td>
<td>2.16 2.16</td>
</tr>
<tr>
<td>18</td>
<td>CS-pack</td>
<td>0.97 seconds</td>
<td>0.97 0.97</td>
</tr>
<tr>
<td>19</td>
<td>Cpl-send</td>
<td>12.44 seconds</td>
<td>12.44 12.44</td>
</tr>
<tr>
<td>20</td>
<td>=</td>
<td>0.00 seconds</td>
<td>0.00 0.00</td>
</tr>
</tbody>
</table>
## POP Log File Timers

### Timing information:

<table>
<thead>
<tr>
<th>Timer number</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51.92 s</td>
<td>EQUATION_OF_STATE</td>
</tr>
<tr>
<td>2</td>
<td>41.22 s</td>
<td>ANISO</td>
</tr>
<tr>
<td>3</td>
<td>41.26 s</td>
<td>HMIX_ANISO_MOMENTUM</td>
</tr>
<tr>
<td>4</td>
<td>79.87 s</td>
<td>HMIX_GM_TRACER</td>
</tr>
<tr>
<td>5</td>
<td>77.04 s</td>
<td>VMIX_COEFFICIENTS_KPP</td>
</tr>
<tr>
<td>6</td>
<td>6.25 s</td>
<td>VMIX_EXPLICIT_TRACER</td>
</tr>
<tr>
<td>7</td>
<td>0.00 s</td>
<td>VMIX_EXPLICIT_MOMENTUM</td>
</tr>
<tr>
<td>8</td>
<td>17.69 s</td>
<td>VMIX_IMPLICIT_TRACER</td>
</tr>
<tr>
<td>9</td>
<td>5.17 s</td>
<td>VMIX_IMPLICIT_MOMENTUM</td>
</tr>
<tr>
<td>10</td>
<td>290.63 s</td>
<td>SEND</td>
</tr>
<tr>
<td>11</td>
<td>154.38 s</td>
<td>RECV</td>
</tr>
<tr>
<td>12</td>
<td>381.75 s</td>
<td>RECV to SEND</td>
</tr>
<tr>
<td>13</td>
<td>0.00 s</td>
<td>SEND to RECV</td>
</tr>
<tr>
<td>14</td>
<td>47.31 s</td>
<td>ADVECTION_STANDARD_TRACER</td>
</tr>
<tr>
<td>15</td>
<td>9.56 s</td>
<td>ADVECTION_MOMENTUM</td>
</tr>
<tr>
<td>16</td>
<td>0.00 s</td>
<td>MOC</td>
</tr>
<tr>
<td>17</td>
<td>0.00 s</td>
<td>TRACER_TRANSPORTS</td>
</tr>
<tr>
<td>18</td>
<td>2.52 s</td>
<td>IO_WRITE_TAVG_DUMP_NCDF</td>
</tr>
<tr>
<td>19</td>
<td>826.75 s</td>
<td>TOTAL</td>
</tr>
<tr>
<td>20</td>
<td>821.47 s</td>
<td>STEP</td>
</tr>
<tr>
<td>21</td>
<td>322.24 s</td>
<td>BAROCLINIC</td>
</tr>
<tr>
<td>22</td>
<td>34.45 s</td>
<td>BAROTROPIC</td>
</tr>
</tbody>
</table>
# CAM Timer Files

Stats for thread 0:

<table>
<thead>
<tr>
<th>Name</th>
<th>Called</th>
<th>Wallclock Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>1</td>
<td>1134.599</td>
<td>1134.599</td>
</tr>
<tr>
<td>ccsm_initializa</td>
<td>1</td>
<td>299.906</td>
<td>299.906</td>
</tr>
<tr>
<td>ccsm_rcvtosnd</td>
<td>241</td>
<td>656.489</td>
<td>10.183</td>
</tr>
<tr>
<td>ccsm_runtotal</td>
<td>1</td>
<td>827.451</td>
<td>827.451</td>
</tr>
<tr>
<td>stepon</td>
<td>1</td>
<td>827.451</td>
<td>827.451</td>
</tr>
<tr>
<td>stepon_startup</td>
<td>1</td>
<td>0.015</td>
<td>0.015</td>
</tr>
<tr>
<td>radcswmx</td>
<td>8640</td>
<td>377.275</td>
<td>0.060</td>
</tr>
<tr>
<td>radclwmx</td>
<td>8640</td>
<td>105.468</td>
<td>0.138</td>
</tr>
<tr>
<td>ccsm_snd</td>
<td>240</td>
<td>1.991</td>
<td>0.045</td>
</tr>
<tr>
<td>ccsm_sndtorcv</td>
<td>240</td>
<td>133.871</td>
<td>1.984</td>
</tr>
<tr>
<td>ccsm_recv</td>
<td>240</td>
<td>35.091</td>
<td>2.526</td>
</tr>
<tr>
<td>ac_physics</td>
<td>481</td>
<td>15.561</td>
<td>0.042</td>
</tr>
</tbody>
</table>
## CLM Timer Files

Stats for thread 0:

<table>
<thead>
<tr>
<th>Name</th>
<th>Called</th>
<th>Wallclock Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>lnd_timeloop</td>
<td>1</td>
<td>827.755</td>
<td>827.755</td>
</tr>
<tr>
<td>clm_driver</td>
<td>482</td>
<td>825.743</td>
<td>8.654</td>
</tr>
<tr>
<td>lnd_recv</td>
<td>241</td>
<td>663.383</td>
<td>8.378</td>
</tr>
<tr>
<td>lnd_recvsend</td>
<td>240</td>
<td>124.379</td>
<td>1.345</td>
</tr>
<tr>
<td>loop1</td>
<td>481</td>
<td>106.095</td>
<td>0.353</td>
</tr>
<tr>
<td>drvinit</td>
<td>481</td>
<td>0.524</td>
<td>0.002</td>
</tr>
<tr>
<td>clm_driver_io</td>
<td>481</td>
<td>2.001</td>
<td>1.970</td>
</tr>
<tr>
<td>wrapup</td>
<td>481</td>
<td>0.019</td>
<td>0.000</td>
</tr>
<tr>
<td>surfalb</td>
<td>240</td>
<td>2.472</td>
<td>0.015</td>
</tr>
<tr>
<td>lnd_send</td>
<td>240</td>
<td>15.415</td>
<td>0.186</td>
</tr>
<tr>
<td>lnd_sendrecv</td>
<td>240</td>
<td>24.562</td>
<td>2.105</td>
</tr>
<tr>
<td>rtm_calc</td>
<td>80</td>
<td>2.991</td>
<td>0.046</td>
</tr>
<tr>
<td>rtm_update</td>
<td>80</td>
<td>0.379</td>
<td>0.006</td>
</tr>
<tr>
<td>rtm_global</td>
<td>80</td>
<td>1.440</td>
<td>0.024</td>
</tr>
</tbody>
</table>
The `getTiming.csh` script

- Does not work with all component options (ex. DATM). Will need to look at log files.
- Assumes LOGDIR set to "" (no log dir)
- Assumes short term archive turned off
- Assumes a fully qualified path is given to the `tdir` parameter (note that "." will not work)
- `cd ${CASEROOT};
  ${CCSMROOT}/scripts/ccsm_utils/Tools/timing/getTiming.csh -mach <machine name> -tdir `pwd`"
Note: for cpl, send=t3~t6, recv=t8~t13, s-r=t15~t16, r-s=t18~t20

<table>
<thead>
<tr>
<th></th>
<th>atm</th>
<th>lnd</th>
<th>ice</th>
<th>ocn</th>
<th>cpl</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf</td>
<td>8*1</td>
<td>1*1</td>
<td>1*1</td>
<td>1*1</td>
<td>1*1</td>
</tr>
<tr>
<td>total</td>
<td>827.451</td>
<td>827.755</td>
<td>827.27</td>
<td>826.75</td>
<td>819.242</td>
</tr>
<tr>
<td>send</td>
<td>0.900</td>
<td>0.186</td>
<td>12.44</td>
<td>290.63</td>
<td>6.105</td>
</tr>
<tr>
<td>recv</td>
<td>27.981</td>
<td>8.378</td>
<td>428.28</td>
<td>154.38</td>
<td>18.103</td>
</tr>
<tr>
<td>s-r</td>
<td>510.633</td>
<td>2.105</td>
<td>323.09</td>
<td>0.00</td>
<td>25.83</td>
</tr>
<tr>
<td>r-s</td>
<td>656.489</td>
<td>1.345</td>
<td>54.79</td>
<td>381.75</td>
<td>21.847</td>
</tr>
</tbody>
</table>

STOP_N is 10. simulation years/day): 2.88

-----------------------------------------------

<table>
<thead>
<tr>
<th>s-r/r-s/(sum of s-r and r-s) ( for cpl send/recv/s-r/r-s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpus</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

-----------------------------------------------
#!/bin/csh

alias MATH 'set \!:1 = `echo "\!:3-$" | bc -l`'

tail -100 cpl > cpls

set val = `grep "id: t01" cpls | cut -b43-52`
echo "t01 = $val"

set val = `grep "id: t02" cpls | cut -b43-52`
echo "t02 = $val"

set val3 = `grep "id: t03" cpls | cut -b43-52`
set val4 = `grep "id: t04" cpls | cut -b43-52`
set val5 = `grep "id: t05" cpls | cut -b43-52`
set val6 = `grep "id: t06" cpls | cut -b43-52`

MATH val = $val3 + $val4 + $val5 + $val6

echo "t3-6 = $val = $val3 + $val4 + $val5 + $val6"

set val = `grep "id: t07" cpls | cut -b43-52`
echo "t07 = $val"

… etc. …
cplstats Output Example

```plaintext
>> cplstats
  t01 = 0.321
  t02 = 2.777
  t3-6 = 6.105 = 4.692 + 0.001 + 1.237 + 0.175
  t07 = 15.205
  t8-13 = 18.103 = 0.589 + 4.596 + 1.653 + 4.229 + 1.899 + 5.137
  t14 = 63.795
  t15-16 = 25.830 = 3.649 + 22.181
  t17 = 8.407
  t18-20 = 21.847 = 5.114 + 0.001 + 16.732
  t21 = 7.187
  t22-23 = 77.416 = 61.027 + 16.389
  t24 = 0.263
  t25 = 570.794
```
Walkabout

- CASEROOT
- EXEROOT
- CCSM web
- CSEG web
- CSEG web (internal)
- Bulletin Board
- Log file examples
- Hard copy: spreadsheet, charts
What Times Are Looked At?

• Timers do not add up
  - Different binaries measuring somewhat different things
  - Aggregation of timer issues

• When min? When max?
  - Transfer times use minimum
  - “Computational” times use maximum
  - Sanity checks to look at spread and variance

• Variation in timers
  - Load imbalance
  - Seasonal and longer variation
  - System events
What To Do? - Ground Rules

• Some hard limitations - cannot use completely arbitrary numbers of processors

• Start from a previously useful scenario

• Choose wisely (luck is ok)
  - Some problems identified at compile, some at runtime
  - Your exploration may lead you to options that may not be obvious ... try them
    • Ex. on IBM bluesky, using 20x4 = 80 CPUs
    • Ex. on IBM thunder, using 6x8 = 48 CPUs
Ground Rules (cont.)

- Keep records (paper, web, spreadsheets)
- Errors come out at various places (some at build time, some run time)
- 10 day run is only an estimate which may be impacted by
  - Seasonal variations
  - Annual variations
  - Longer term variations
  - Current timers do not make looking at these issues easy
Component Set Issues

• Unless otherwise stated all examples are fully coupled (i.e. Component set B with POP, CAM, CSIM, CLM, and CPL)

• General process applies to other choices
Data Decomposition Observations

• CAM
  - Must be factor of 2
  - May be factor of 3 or 5
  - Maximum MPI tasks based on resolution (T31 - 48, T42 - 64, T85 - 128)
  - Might be good to be an integral factor of max resolution size
  - Often good to fit into node reasonably
  - Might be able to use MPI and OpenMP
  - Minimum of 2 (all others have minimum of 1)
  - Number of processors does not change the numeric results (not true of all)

• CPL
  - More flexible (can use odd prime numbers for example)
  - “Good” integral factors still seem to be better

• Others
  - Similar kinds of decomposition guidelines
Some Additional Items

- Data model only run on 1 CPU
- Group like processes to same node to fill nodes and reduce communication
  - Rearrange COMPONENTS in env_mach.<machid>
    - set COMPONENTS = ($COMP_CPL $COMP_ICE $COMP_LND $COMP_OCN $COMP_ATM)
- You can’t always “balance” the model
- I/O can be very important (including LOG files) including your neighbor’s use of it
- Your neighbor’s use of the network can be very important even if you can’t control it
- Where your nodes are on the network can be very important even if you can’t control it
- Reducing runtime of one component can improve another particularly when on same node
- Things will change
CCSM3 Process Flow

- **Targets**
  - $A \leq B + C$
  - $D < B$
  - $F < B$
  - $G < C$
  - $E < C$
  - $D < F$

- **Observations**
  - $B < C$
  - $D < E$
  - $F > G$
  - Scaling of $B$ different than $C$
  - CPL/ICE/LND will allows have idle time
OK ... How To Go About It?

• Start with CAM
  - Majority of CPUs assigned to CAM
  - Look at integral factors of resolution
  - Look at node size factors
  - Consider OpenMP option (where possible)

• Match POP to CAM processing time

• Pick smallest reasonable number of CPUs for other components such that CAM is not delayed
OK ... What Do I Really Do?

- Pick a configuration ... try it
- Look at CAM
  - Are there MPI wait times?
  - Which? Why?
- Compare POP to idealized CAM time
- Look at ICE and LND
  - Compare “compute” time phases to CAM
  - Examine MPI wait times
- Look at CPL times
  - “Compute” phases
  - Transfer phases
- Change a couple things and try it
<table>
<thead>
<tr>
<th>CCSM Version</th>
<th>Machine</th>
<th>Date</th>
<th>Resolution</th>
<th>Config #</th>
</tr>
</thead>
<tbody>
<tr>
<td>__________</td>
<td>__________</td>
<td>__________</td>
<td>__________</td>
<td>__________</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Years/day</th>
<th>Years/day/cpu</th>
<th>CPL main time</th>
<th>CPL avg dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>__________</td>
<td>__________</td>
<td>__________</td>
<td>__________</td>
</tr>
</tbody>
</table>

---

**Diagram**

- **OCN**
- **ATM**
- **ICE**
- **LND**
- **CPL**
- **Total**

- **t01**
- **t02**
- **t07**
- **t14**
- **t17**
- **t21**
- **t24**
- **t25**
<table>
<thead>
<tr>
<th>CCSM Version</th>
<th>Machine</th>
<th>Date</th>
<th>Resolution</th>
<th>Config #</th>
<th>Years/day</th>
<th>Years/day/cpu</th>
<th>CPL main time</th>
<th>CPL avg dt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Diagram**

- **OCN**
  - t01
  - t02
  - t07
  - t14
  - t17
  - t21

- **ATM**
  - t24
  - t25

- **ICE**
- **LND**
- **CPL**
- **Total**

- **NCAR**
Running CCSM: The Basic Steps

- cd ${CCSMROOT}/scripts
- ./create_newcase -case ~/test/T31x3 -mach calgary -res T31_gx3v5 -compset B
- cd ~/test/T31x3
- edit env_run to set run for 10 days. I also usually set INFO_DBUG to 0 and DIAG_OPTION to never (but that's not required).
- edit env_mach.calgary to set DOUT_S to FALSE
- configure -mach calgary
- ${CASE}.calgary.build [this builds and prestages data for the run]
- edit the ${CASE}.calgary.build if you need to set queues, time limits, or accounts for PBS
- qsub ${CASE}.calgary.run
- ${CCSMROOT}/scripts/ccsm_utils/Tools/timing/getTiming.csh -mach calgary -tdir `pwd`

Note: See CCSM User's Guide and CCSM Scripts Tutorial
Cray X1

- ORNL’s Phoenix
  - Each node has 4 MSPs
  - Queuing in multiples of 4 MSPs
- T31x3 standard run
- Started with 6 nodes (24 MSPs)
- CAM: 12 MPI tasks (12 MSPs)
- Goal: find small configuration
  - Better efficiency
  - Better queue time
<table>
<thead>
<tr>
<th>CCSM Version</th>
<th>M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td>Phoenix</td>
</tr>
<tr>
<td>Date</td>
<td>10/16/04</td>
</tr>
<tr>
<td>Resolution</td>
<td>T31x3</td>
</tr>
<tr>
<td>Config #</td>
<td>12-1</td>
</tr>
<tr>
<td>OCN</td>
<td>20</td>
</tr>
<tr>
<td>ATM</td>
<td>25</td>
</tr>
<tr>
<td>ICE</td>
<td>29</td>
</tr>
<tr>
<td>LND</td>
<td>75</td>
</tr>
<tr>
<td>CPL</td>
<td>129</td>
</tr>
<tr>
<td>CPL avg dt</td>
<td>13 (12-14)</td>
</tr>
</tbody>
</table>

| Years/day | 18.4 |
| Years/day/cpu | 0.7667 |
| CPL main time | 129 |
| Total | 16
CCSM Version       __________
Machine            __________
Date               __________
Resolution         __________
Config #           __________

OCN            20
ATM            25
ICE            29
LND            75
CPL            84

Years/day          18.4
Years/day/cpu      0.7667
CPL main time       129
CPL avg dt         13 (12-14)

Try 1
Try 1
Try 1
Try 4

Total
<table>
<thead>
<tr>
<th>CCSM Version</th>
<th>Machine</th>
<th>Date</th>
<th>Resolution</th>
<th>Config #</th>
</tr>
</thead>
<tbody>
<tr>
<td>____________</td>
<td>Phoenix</td>
<td>12/9/04</td>
<td>T31x3</td>
<td>12-1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OCN</th>
<th>ATM</th>
<th>ICE</th>
<th>LND</th>
<th>CPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>12</td>
<td>2</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>Try 1</td>
<td>Try 1</td>
<td>Try 2</td>
<td>Try 4</td>
<td>Total</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Years/day</th>
<th>Years/day/cpu</th>
<th>CPL main time</th>
<th>CPL avg dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.4</td>
<td>0.7667</td>
<td>129</td>
<td>13 (12-14)</td>
</tr>
<tr>
<td>Time</td>
<td>OCN</td>
<td>ATM</td>
<td>ICE</td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>t01</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t02</td>
<td></td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>t07</td>
<td>37</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>t14</td>
<td></td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>t17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CCSM Version**: M3

**Machine**: Phoenix

**Date**: 12/9/04

**Resolution**: T31x3

**Config #**: 12-2

**Years/day**

**Years/day/cpu**

**CPL main time**

**CPL avg dt**
<table>
<thead>
<tr>
<th>CCSM Version</th>
<th>Machine</th>
<th>Date</th>
<th>Resolution</th>
<th>Config #</th>
<th>Years/day</th>
<th>Years/day/cpu</th>
<th>CPL main time</th>
<th>CPL avg dt</th>
</tr>
</thead>
<tbody>
<tr>
<td>____________</td>
<td>Phoenix</td>
<td>12/9/04</td>
<td>T31x3</td>
<td>12-2</td>
<td>21.81</td>
<td>1.0905</td>
<td>108</td>
<td>11</td>
</tr>
</tbody>
</table>

**Diagram:****

- **OCN**
  - 1
  - 20
  - 21

- **ATM**
  - 12
  - 25
  - 76

- **ICE**
  - 1
  - 37
  - 10
  - 14
  - 62

- **LND**
  - 2
  - 71
  - 20
  - 5
  - 13

- **CPL**
  - 4
  - t01
  - t02
  - t07
  - t14
  - t17
  - t21
  - t24
  - t25

- **Total**
  - 20
  - 0
  - 1
  - 3
  - 4
  - 10
  - 2
  - 5
  - 0
  - 1
  - 47
<table>
<thead>
<tr>
<th>CCSM Version</th>
<th>M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td>Phoenix</td>
</tr>
<tr>
<td>Date</td>
<td>12/9/04</td>
</tr>
<tr>
<td>Resolution</td>
<td>T31x3</td>
</tr>
<tr>
<td>Config #</td>
<td>12-2</td>
</tr>
</tbody>
</table>

| Years/day | 21.81 |
| Years/day/cpu | 1.0905 |
| CPL main time | 108 |
| CPL avg   | 11 |

Changes might reduce efficiency

Changes might reduce efficiency

Try 4

Try 6

Should drop to 14

Should drop further

Should drop further
Cray X1

• ORNL’s Phoenix
  - Each node has 4 MSPs
  - Queuing in multiples of 4 MSPs
• T85x1 standard run
• Started with 34 nodes (136 MSPs)
• CAM: 64 MPI tasks (64 MSPs)
• Goal: Looking for compromise of years per day and efficiency
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CCSM Version</td>
<td>M3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine</td>
<td>Phoenix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>10/14/04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>T85x1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Config #</td>
<td>64-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OCN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ICE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Years/day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9.73</td>
<td></td>
</tr>
<tr>
<td>Years/day/cpu</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0737</td>
<td></td>
</tr>
<tr>
<td>CPL main time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>CPL avg dt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24 (22 - 31)</td>
<td></td>
</tr>
</tbody>
</table>

- OCN: 35, 174, 29
- ATM: 25, 39, 179, 2
- ICE: 180, 40, 11, 14
- LND: 137, 13, 87, 135
- CPL: 180, 40, 11, 14

Days: 0, 2, 14, 6, 2, 1, 23, 6, 154

Notes:
- Was 8.24
- Was 0.0665
- Was 287
- Was 29
- Was 79
- Was 8, 25, 14, 5
- Was 8, 25, 14, 5
- Was 0, 2, 31, 10, 3, 1
**From Previous Tests we know that 48 LNDs only reduces from 40 to 35. Need to speed up LND and CPL components.**
IBM 8 and 32 Way Example

- NCAR’S bluesky
  - Each node has 8 processors
    - More network connections
  - Each node has 32 processors
    - Fast messaging for 32 processors on node
  - Colony switch
  - Job queuing in whole node multiples

- T85x1 standard run

- 24 8way nodes or 6 32way nodes (192 CPUs) (common IPCC job size)

- CAM: 32 MPI tasks, 4 threads per MPI (128 CPUs)
CCSM Version: Rel04
Machine: Bluesky8
Date: 10/13/04
Resolution: T85x1
Config #: 128-1

Years/day: 4.30
Years/day/cpu: 0.0224
CPL main time: 550
CPL avg dt: 55

24 OCN
32x4 ATM
8 ICE
24 LND
8 CPL
192 Total

NCAR
CCSM Version
Rel04
Machine
Bluesky32
Date
10/13/04
Resolution
T85x1
Config #
128-1

Years/day
4.30
Years/day/cpu
0.0224
CPL main time
550
CPL avg dt
55

Was 474
OCN
59
Was 63
ATM
Was 84
456
Was 407
ICE
Was 2
36
459
Was 2
LND
Was 278
237
Was 10
CPL
Was 27
52
20
Was 0
24
8
24
8
192
Total
1
1
7
17
49
1
3
13
17
431

NCAR
8 way vs 32 way - What Happened?

- Allocation of processes to processors
  
  - set COMPONENTS = ($COMP_CPL $COMP_ICE $COMP_LND $COMP_OCN $COMP_ATM)
  
  32way: (8c8i16l),(8l,24o),4x(32a)
  8way: (8c),(8i),3x(8l),3x(8o),16x(8a)

- Anything wrong? Anything better?
  
  - Land split across two nodes

- Might make better use of 32 way
  
  32way: (8c24l),(8i,24o),4x(32a)
  32way: (8c24o),(8i,24l),4x(32a)
CCSM Hybrid Example on IBM

- Thunder is IBM system with four 16 way nodes and Federation switches
- We typically run with 4 CAM threads on IBM systems
- Tried 0, 4, and 8 threads keeping total number of CAM processors constant 48

<table>
<thead>
<tr>
<th>Num CAM MPI tasks</th>
<th>Num CAM Threads</th>
<th>Coupled Years per Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>0</td>
<td>18.87</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>20.34</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>21.94</td>
</tr>
</tbody>
</table>
For Further Information

- **CCSM web pages**
  - http://www.ccsm.ucar.edu/ccsm3
  - http://www.ccsm.ucar.edu/support_model
    - See CCSM User’s Guide
    - See Scripts Tutorial
  - http://www.ccsm.ucar.edu/support_model/mach_support.html

- **CCSM Bulletin Board**
  - http://bb.cgd.ucar.edu

- gcarr@ucar.edu