NCAR CCSM with Task Geometry Support in LSF

Mike Page
ScicomP 11 Conference
Edinburgh, Scotland
June 1, 2005
NCAR/CISL/SCD
Consulting Services Group
mpage@ucar.edu

Overview

- Description of CCSM
- Task Geometry
- Task Geometry Benchmarks
- Load Sharing Facility - LSF
- Throughput Benchmarks
- Conclusions

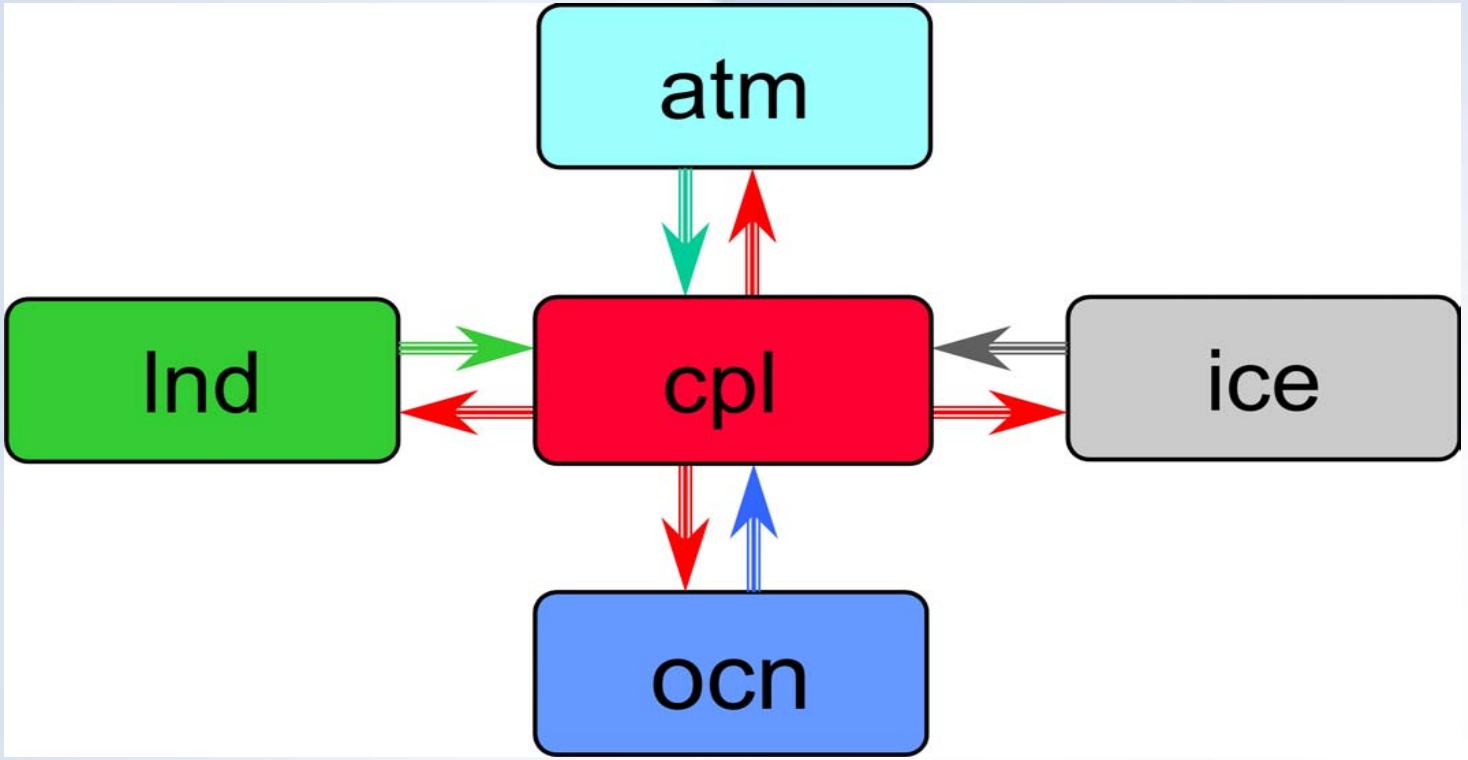
Description of CCSM

The Model

- Community Climate System Model
  - Version 3.0
- Models Earth's Climate
- Development supported by NSF and DoE

Description of CCSM

Hub and Spoke Model



Description of CCSM

Model Types

Model Type	Name	Use
Active	atm lnd ice ocn	Actively models physics and dynamics of respective geophysical system
Data	datm dlnd dice docn	Reads existing datasets that were previously written by the dynamical models and passes the resulting data to the coupler
Dead	xatm xlnd xice xocn	Generate unrealistic forcing data internally, require no input data and can be run on multiple processors to simulate the software behavior of the fully active system

Description of CCSM

Usage Modes

Component Name	Version	Type	MPI Parallel	OpenMP Parallel	Hybrid* Parallel
cam	cam3	active	YES	YES	YES
datm	datm6	data	NO	NO	NO
latm	latm6	data	NO	NO	NO
xatm	dead	dead	YES	NO	NO
clm	clm3	active	YES	YES	YES
dlnd	dlnd6	data	NO	NO	NO
xlnd	dead	dead	YES	NO	NO
pop	ccsm_pop_1_4	active	YES	NO	NO
docn	docn6	data	NO	NO	NO
xocn	dead	dead	YES	NO	NO
csim	csim5	active	YES	NO	NO
dice	dice6	data	NO	NO	NO
xice	dead	dead	YES	NO	NO
cpl	cpl6	active	YES	NO	NO

Task Geometry

History of Blackforest

Year	Node Type	Node Count	Processors per Node	Peak GFlops
1999	Winterhawk I	148	2	237
2000	Winterhawk II	151	4	906
2001	Winterhawk II Nighthawk II	315 3	4 16	1962

- LL task geometry developed 2000
- LSF 'paper' evaluation, 2003-2004
- LSF installed on Bluedawn and Thunder
- LSF task geometry developed in 2004
- Blackforest decommissioned in 2005

Task Geometry

LoadLeveler syntax

#@ task_geometry={(*task id*,...) ... **}** # LoadLeveler directive

NOTE: In LoadLeveler, task id refers to an MPI task
- LoadLeveler is "thread unaware"

Example (bluedawn, 4 4-way nodes):
#@ task_geometry={(0)(1)(2,3,4,5)(6)}

Can be used to locate 1 cam MPI task on each of nodes 0 and 1
(each runs 4 threads),
2 clm MPI tasks on node 2,
1 serial dice on node 2,
1 serial docn task on node 2 and
1 serial cpl task on node 3

Don't try this at home

Threadedness specified by content of file named by MP_CMDFILE

Task Geometry Benchmarks

CCSM Setup on Thunder
Four 16-way nodes

Component	Number of Processors
CPL	2 MPI Tasks
ICE	2 MPI Tasks
LND	8 MPI Tasks
POP	4 MPI Tasks
CAM	Test 1 - 48 MPI Tasks Test 2 - 12 MPI Tasks, 4 OpenMP Threads Test 3 - 6 MPI Tasks, 8 OpenMP Threads

Task Geometry Benchmarks

CCSM on Thunder

Configuration	Simulated Years per Day
48 MPI Tasks, no OpenMP Threads	18.87
12 MPI Tasks, 4 OpenMP Threads	20.34
6 MPI Tasks, 8 OpenMP Threads	21.94

Load Sharing Facility

LSF Task Geometry syntax

setenv LSB_PJL_TASK_GEOMETRY "{(task id, …) …}”

NOTE: In LSF, task id refers to either an MPI task or an OpenMP thread
- LSF is “thread aware”

Example (bluedawn):
setenv LSB_PJL_TASK_GEOMETRY "{(0)(1)(2,3,4,5)(6)}”

Can be used to locate 1 cam MPI task on both nodes 0 and 1
(each runs 4 threads),
2 clm MPI tasks on node 2,
1 serial dice on node 2,
1 serial docn task on node 2 and
1 serial cpl task on node 3

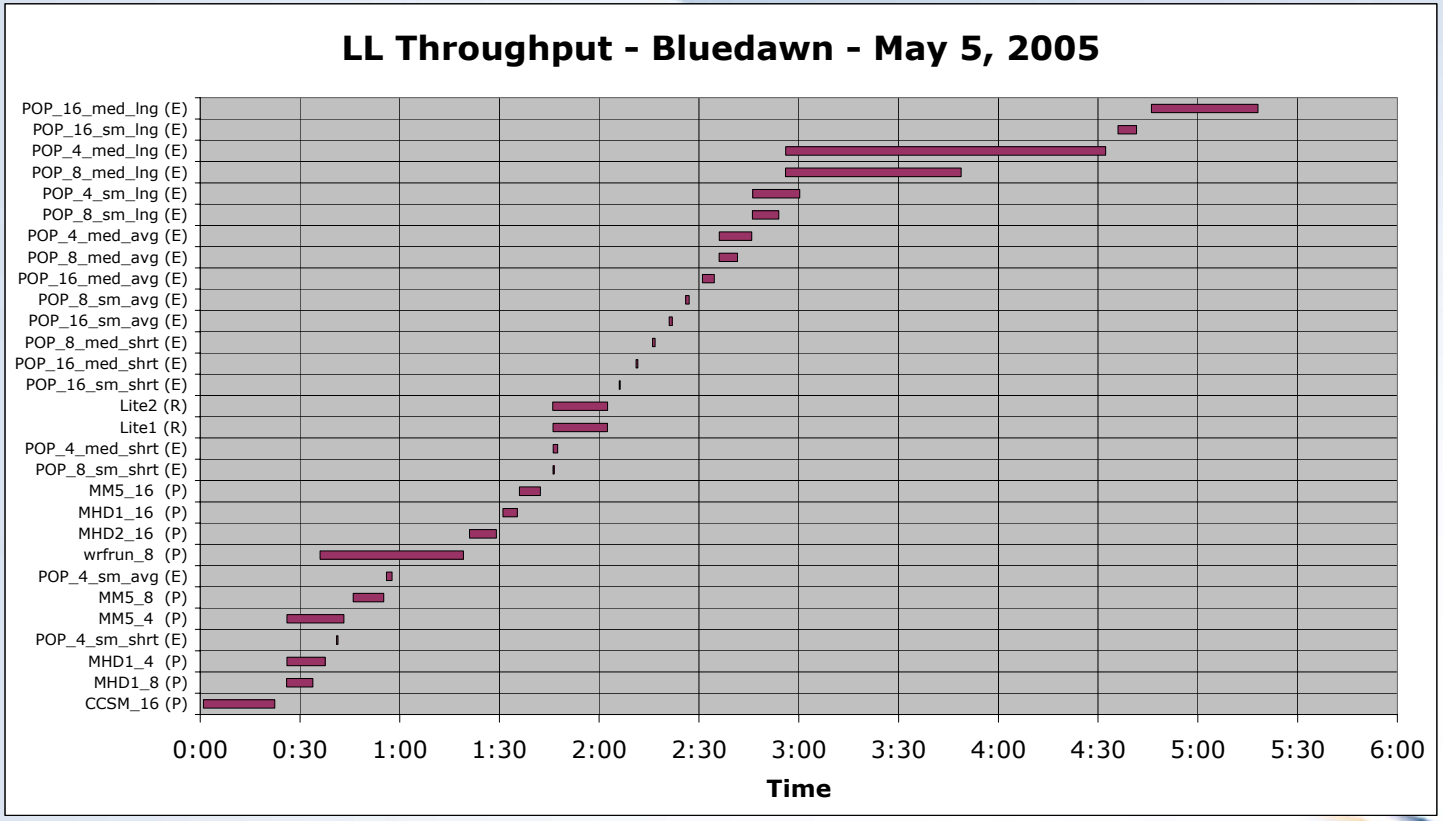
Load Sharing Facility

LSF is “thread aware”

```
#BSUB -a 'poe'                  # select the POE elim
#BSUB -n 16                     # number of tasks - LSF’s “thread-awareness”
#BSUB -R "span[ptile=4]"        # the largest number of tasks asked for in task geom
#BSUB -x                        # exclusive us of node (not_shared)
.
.
.
# Setup LSF Task Geometry
setenv LSB_PJL_TASK_GEOMETRY "{(0)(1)(2,3,4,5)(6)}"
.
.
.
mpirun.lsf -cmdfile cmdfile     #cmdfile  serves same purpose as MP_CMDFILE
```

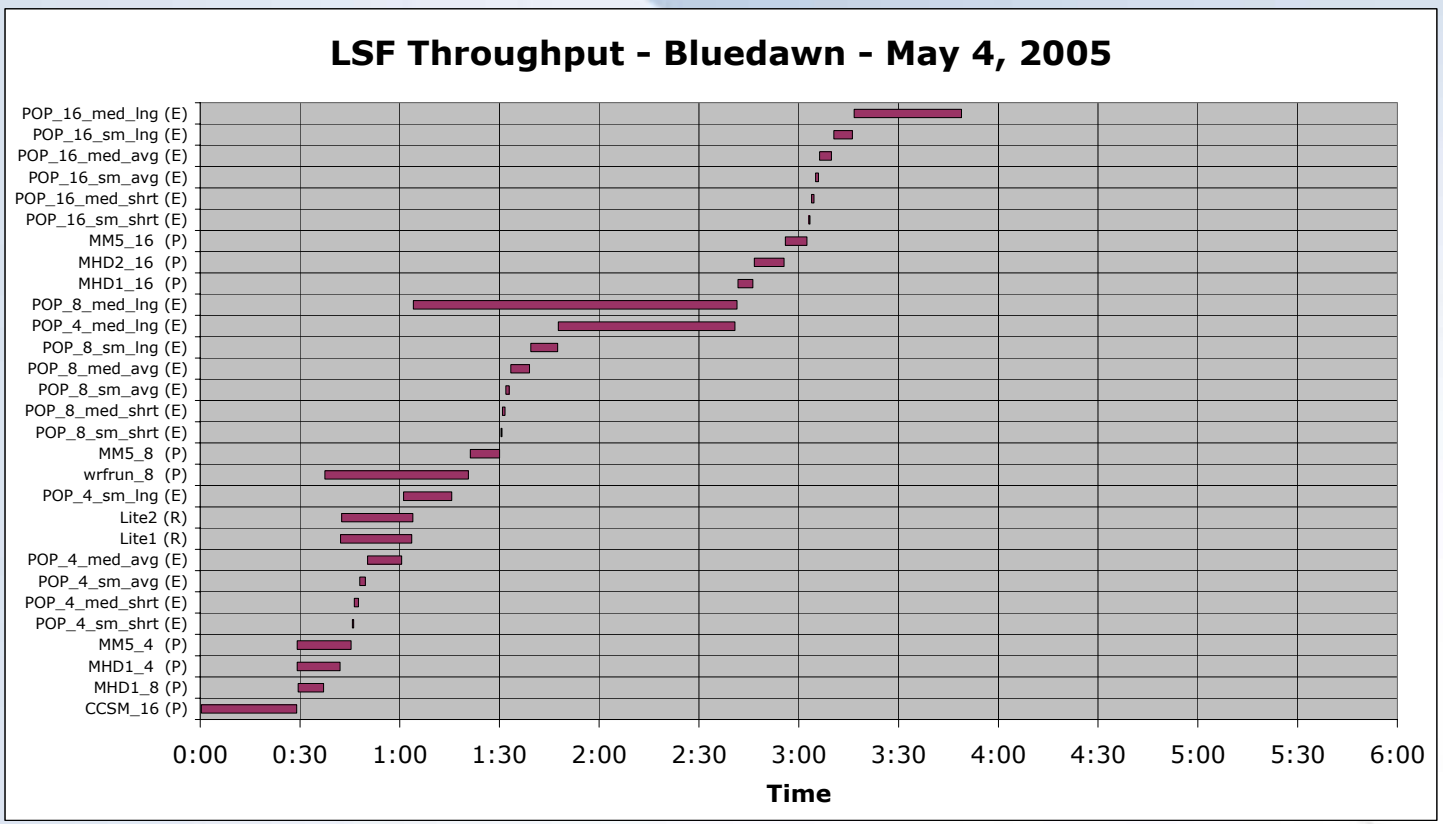
Throughput Benchmarks

Sample job load on Bluedawn/LoadLeveler



Throughput Benchmarks

Sample job load on Bluedawn/LSF



Conclusions

- 1+ year of experience with LSF
  - Bluedawn
  - Thunder
  - Lightning
- Task Geometry functional under LSF
- User acceptance of LSF is good
- Future plans
  - Test CCSM performance on Lightning
  - Test other LSF features
    - Inter-platform operability
    - Grid capability

The End

Questions?