Guardians of the ACME
and
Only YOU Can Prevent Performance Fires

Phil Jones
On behalf of the Performance Team

ACME All-Hands
June 2017
Guardians of the ACME
I am groot

- I am groot
- I am groot

- I am groot
  - I am groot
  - I am groot
Performance Group Roles

• Hero role: v1 support
  – POCs, SEs, Coupled Group
  – Fixing bugs
  – PE Layouts, PIO config

• Performance optimization
  – Evaluating and enhancing performance
  – Focus on current sims, current machines

• Preparing for Transition
  – Preventing future disasters
  – Early access
  – Identifying changes needed
  – Prototyping new ideas

• Performance Team
  – Az, Noel, Jayesh, Pat, Phil, Ben, Min
  – ECP/CAAR: Matt N., Sarat, Erich
Hero Contributions

• Machine Specific (POCs)
  – Changes in environment
    • Modules, compilers, etc.
    • Batch systems
  – Sporadic slowdowns Edison

• PE Layouts (next slide)

• Bugs
  – I/O Stack: pio, pnetcdf, netcdf
  – CIME
  – Thread irreproducibility
  – Misc.

• Infrastructure review

• Improved testing coverage
  – Integrators, developers, machines
Biggest weapons/knob

- PE layouts
- See: Benchmark Results and Optimal Layout in Perf Confluence

Caveats:
- Out-of-box results, YMMV
- Little/no I/O, YMMV
- Sometimes outdated

Default PE layouts
If you see something, say something
- Try some tweaks, add your results
Internal Changes

- Nested threading in atm
- Threading in sea-ice
- CLUBB optimizations
- Atmospheric physics load balancing
- Improvements to initialization
  - Communication algorithms
Preparing for Next Generation

• Nested threading, improved threading, affinity
• GPU acceleration
  – superparameterization
• Vectorization
• Integrated projects
  – ECP
  – CAAR, NESAP
  – CMDV
  – SciDAC
• Significant code refactoring
• All above likely to be 2x-3x range.
• Need algorithm changes, new ideas

KNL: 68 cores, 272 threads, Vector units, high-bandwidth memory

GPU: SIMD cores, separate memory

ARM?????
Only YOU can prevent performance fires

Seriously. We’ve been defunded. It’s just you now.
• Extreme
  – Preventing high-priority sims
  – Type 1 Incident Response, multi-agency, aircraft, etc.

• Very High
  – Blocking of other sims (esp. high resolution), Substantial reduction in high-priority sim
  – Type 2 Incident Response

• High
  – Significant performance hit
  – Hot Shot team

• Moderate
  – Known issue
  – Workaround (containment)

• Low
  – No immediate threat, let burn
At a minimum

- Look at timing info:

<table>
<thead>
<tr>
<th>Component</th>
<th>Time (seconds)</th>
<th>Time (seconds/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init Time</td>
<td>36.677</td>
<td></td>
</tr>
<tr>
<td>Run Time</td>
<td>1117.133</td>
<td>37.238</td>
</tr>
<tr>
<td>TOT Run Time</td>
<td>1117.133</td>
<td>37.238</td>
</tr>
<tr>
<td>LND Run Time</td>
<td>31.850</td>
<td>1.062</td>
</tr>
<tr>
<td>ROF Run Time</td>
<td>1.150</td>
<td>0.038</td>
</tr>
<tr>
<td>ICE Run Time</td>
<td>66.940</td>
<td>2.231</td>
</tr>
<tr>
<td>ATM Run Time</td>
<td>945.307</td>
<td>31.510</td>
</tr>
<tr>
<td>OCN Run Time</td>
<td>170.009</td>
<td>5.667</td>
</tr>
<tr>
<td>GLC Run Time</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>WAV Run Time</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>CPL Run Time</td>
<td>482.530</td>
<td>16.084</td>
</tr>
<tr>
<td>CPL COMM Time</td>
<td>1068.084</td>
<td>35.603</td>
</tr>
</tbody>
</table>

6.36 myears/wday
222.96 myears/wday
6175.10 myears/wday
106.09 myears/wday
7.51 myears/wday
41.77 myears/wday
0.00 myears/wday
0.00 myears/wday
14.72 myears/wday
6.65 myears/wday
At a minimum

- Adjust PE layout
- Rules of thumb – see Confluence:
  - How to Create PE Layout in Atm How-to
- But…always exceptions, Edisonian

---

**ATM**
- 675 tasks, 2 threads per task,
- 945 seconds, 7.5 SYPD

**OCN**
- 128 tasks, 2 threads per task,
- 170.0 seconds, 42 SYPD

**ICE:**
- 512 tasks, 2 threads per task,
- 67 seconds, 107 SYPD

**CPL**
- 512 tasks, 2 threads per task,
- 483 seconds

**LND**
- 168 tasks,
- 2 threads per task,
- 32 seconds, 223 SYPD

**processor cores**
- 0 1024 1360 1615

---

*Acme Accelerated Climate Model for Energy*
Climate Exceptionalism

- Climate Lags Others
  - Failing readiness
  - Can’t use GPU or KNL, but at least we scale poorly
Climate Exceptionalism

- Climate Lags Others
  - Failing readiness
  - Can’t use GPU or KNL, but at least we scale poorly

- Climate Modelers are Computational Scientists
  - Science
  - Algorithms
  - Computer Science
Climate Exceptionalism

• Climate Lags Others
  – Failing readiness
  – Can’t use GPU or KNL, but at least we scale poorly

• Climate Modelers are Computational Scientists
  – Science
  – Algorithms
  – Computer Science

• Back in the Day…
  – Where performance comes from
  – Pros and Cons of abstractions
Climate Exceptionalism

- Climate Lags Others
  - Failing readiness
  - Can’t use GPU or KNL, but at least we scale poorly
- Climate Modelers are Computational Scientists
  - Science
  - Algorithms
  - Computer Science
- Back in the Day…
  - Where performance comes from
  - Pros and Cons of abstractions
- Everyone must think
  - Performance is MY responsibility
We are SPARTA! (SuperParameterized ACME Refactored for Tomorrow's Architectures)