Optimization of Climate/Weather Applications for Cray Architectures

John Levesque
Senior Technologist
Product Development
A Long and Proud History

Seymour Cray
Founded Cray Research 1972
The Father of Supercomputing

Cray-1 System (1976)
First Supercomputer
160 MFLOPS Processor
160 MFLOPS System Peak

First Multiprocessor Supercomputer
235 MFLOPS Processor
940 MFLOPS System Peak

Cray-C90 System (1991)
First Computer with 1 GFLOP Processor
1 GFLOPS Processor
16 GFLOPS System Peak

Cray T3E System (1996)
World’s Most Successful MPP
900-1350 MFLOPS Processor
2.8 TFLOPS System Peak

Cray X1 System (2002)
Vector MPP Architecture
12.8 GFLOPS Processor
52 TFLOPS System Peak

Strider (2004)
Largest X86 System
4.8 GFLOPS Processor
41 TFLOPS System Peak

Cray XD1 System (2004)
X86 Direct Connect Architecture
4.8 GFLOPS Processor
5 TFLOPS System Peak

(Cray Inc Founded 2000)
Cray’s Family of Supercomputers

Capacity Computing

XD1 & Red Storm

- 1 to 50+ TFLOPS
- 16 – 10,000+ processors
- Compute system for large-scale sustained performance

Purpose-Built High Performance Computers
Cray’s Family of Supercomputers

Capability Computing

Cray X1

- 1 to 50+ TFLOPS
- 4 – 4,069 processors
- Vector processor for uncompromised sustained performance

Purpose-Built High Performance Computers
Characteristics of Systems

- **Cray X1/X1E**
  - 12.8/18 GFLOPS Processors
  - Vector Processing
  - High Vector Memory Bandwidth
  - Low latency Network
  - Highest Bandwidth Network

- **Cray XD1/Red Storm**
  - .GT. 4 Gigaflop COTS processor
  - Superscalar Opertron
  - Highest Bandwidth micro-processor
  - Low latency Network
  - High Bandwidth Network

- **Cray X1/X1E**
  - Must Vectorize Code
  - Co-Array Fortran and UPC for optimizing communication

- **Cray XD1/Red Storm**
  - Cache Based system
  - SHMEM available for optimizing communication
Leadership Class Computing

- Cray-ORNL Selected by DOE for National Leadership Computing Facility (NLCF)
- Goal: Build the most powerful supercomputer in the world
- 250-teraflop capability by 2007
  - 50-100 TF sustained performance on challenging scientific applications
  - Cray X1/X1E and ‘Red Storm’ products
- Focused on capability computing
  - Available across government, academia, and industry
  - Including biology, climate, fusion, materials, nanotech, chemistry
  - Open scientific research
Cray X1 Systems

- **Widespread adoption**
  - Domestic and international; Government and commercial

- **In leading positions of the most powerful single computers**
  - (International Data Corporation Balanced Ratings – 2003)
  - 12.8 GF CPU with high memory bandwidth and sustained performance

- **Ten Cray X1 systems in TOP500 (November 2003)**
  - Three 256-CPU systems at positions #19, #20, and #21

- **Enabling New Science**
  - Improved weather forecast accuracy – 5km resolution model of entire U.S. in less than 2 hours
  - Parallel Ocean Program (POP) running 50% faster per CPU than the Earth Simulator
  - 1 TF Sustained performance on an unstructured finite element method-based fluid dynamics application
  - NASA CFD code run on single cabinet Cray X1 can accomplish work which used to take a week, in a single day.
The Cray XD1 Supercomputer

- **Built for price/performance**
  - Interconnect bandwidth/latency
  - System-wide process synchronization
  - Application Acceleration FPGAs

- **Standards-based**
  - 32/64-bit X86, Linux, MPI

- **High resiliency**
  - Self-configuring, self-monitoring, self-healing

- **Single system command & control**
  - Intuitive, tightly integrated management software

Entry/Mid Range System Optimized for Sustained Performance
Cray XD1

**Compute**
- 12 AMD Opteron processors 32/64 bit, x86 processors
- High Performance Linux

**RapidArray Interconnect**
- 12 communications processors
- 1 Tb/s switch fabric

**Active Management**
- Dedicated processor

**Application Acceleration**
- 6 FPGA co-processors

**Processors directly connected via integrated switch fabric**
Outline of Talk

- Investigation of POP Optimization
- Optimization of MM5 and WRF
- Optimization of CAM
Optimization of POP

- Over the past 2 years POP’s execution on the X1 has received considerable attention
  - 1 Degree Model
  - Using POP 1.4.3
Optimization observations

- POP out of the box did not run well
Result of High Bandwidth

Comparisons of Various Versions of POP

- ES
- Altix
- XD1
- Power 4

Simulated Years/CPU Days vs. Processors
Optimization observations

• POP out of the box did not run well
• Vectorized impvmix_t, impvmix_u
  – Inner K loop was recursive
  – Vectorized on I and streamed on J
• Vectorized hmix_aniso
  – Inner loops on quadrants and CASE statements were preventing vectorization
  – Vectorized on I and streamed on J
Optimization observations

- POP out of the box did not run well
- Vectorized impvmix_t, impvmix_u
  - Inner K loop was recursive
  - Vectorized on I and streamed on J
- Vectorized hmix_aniso
  - Inner loops on quadrants and CASE statements were preventing vectorization
  - Vectorized on I and streamed on J
- Rewrote global_sum and ninept_4 in CAF
- Tried different Synchronization techniques
Address Translation

- **Source** translation using 256 entry TLBs with multiple page sizes: virtual page # bits translated locally
  - Allows non-contiguous node jobs or single CPU job reference off-node memory
- **Remote** translation (RTT): virtual page # bits represent logical node #
  - Logical node bits + BaseNode \(\Rightarrow\) physical node, page offset translated remotely
  - TLB only needs to hold translations for one node \(\Rightarrow\) **scales with no TLB misses**

---

<table>
<thead>
<tr>
<th>VA:</th>
<th>MBZ</th>
<th>Virtual Page #</th>
<th>Page Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory region: useg, kseg, kphys
Possible page boundaries: 64 KB to 4 GB

<table>
<thead>
<tr>
<th>PA:</th>
<th>Node</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Physical address space:
Main memory, MMR, I/O
Max 64 TB physical memory

Memory region:
32 16
Page Offset Virtual Page #
31 15
Possible page boundaries:
MBZ 47 48 useg, kseg, kphys
47 46 45 36 35
45 036 Offset
44 43
Physical address space:
PA: Main memory, MMR, I/O
Max 64 TB physical memory

Source translation using 256 entry TLBs with multiple page sizes: virtual page # bits translated locally
- Allows non-contiguous node jobs or single CPU job reference off-node memory

Remote translation (RTT): virtual page # bits represent logical node #
- Logical node bits + BaseNode \(\Rightarrow\) physical node, page offset translated remotely
- TLB only needs to hold translations for one node \(\Rightarrow\) scales with no TLB misses
Using Global Address Capability

- Co-Array Fortran
  - Array must be symmetric – or
    - Use derived types with pointers
  - Actually use pointer directly, manipulating the address to get to other processors
- Advantages
  - No need to pack buffers, simply access boundary information from neighbor processors.
Comparisons of Various Versions of POP

As-is POP
Vectorized POP
Vec. POP w/CAF
Vec POP w/sync-free
ES
Altix
XD1
- POP out of the box did not run well
- Vectorized impvmix_t, impvmix_u
  - Inner K loop was recursive
  - Vectorized on I and streamed on J
- Vectorized hmix_aniso
  - Inner loops on quadrants and CASE statements were preventing vectorization
  - Vectorized on I and streamed on J
- Remote global_sum and ninept_4 in CAF
- Tried different Synchronization techniques
- OPERATING SYSTEM IMPROVEMENT
- MPI-LITE
Comparisons of Various Versions of POP

- As-is POP
- Vectorized POP
- Vec. POP w/CAF
- Vec POP w/sync-free
- Vec POP w/Best Barrier
- MPI-Lite
- ES
- Altix
- XD1

**OS Improvement**

**MPI-Lite**
Optimization observations

• POP out of the box did not run well
• Vectorized impvmix_t, impvmix_u
  – Inner K loop was recursive
  – Vectorized on I and streamed on J
• Vectorized hmix_aniso
  – Inner loops on quadrants and CASE statements were preventing vectorization
  – Vectorized on I and streamed on J
• Remote global_sum and ninept_4 in CAF
• Tried different Synchronization techniques
• OPERATING SYSTEM IMPROVEMENT
• AND MPI HAS IMPROVED
Comparisons of Various Versions of POP

Simulated Years/CPU Days

OS improvement

MPI Improvement

MSPs

As-is POP
Vectorized POP
Vec. POP w/CAF
Vec POP w/Best Barrier
Latest MPI
ES
Altix
XD1
Network Bandwidth Example

FPMD performance

FPMD performance - Mflop/s

- SP4 emeral
- X1 h2obig
- SP4 h2obig
- X1 emeral

npes
WRF on Cray X1

- WRF software design is well suited to Cray X1 architecture.
  - Designed with both parallel and vector hardware in mind.
  - Eulerian Mass dynamics core has ‘tiling’ feature which maps to OpenMP or Cray streaming.
  - Storage order (J,K,I) or (latitudes, vertical, longitudes) allows efficient use of X1 architecture:
    - Parallel over J with MPI and Cray streaming or OpenMP
    - Another parallel level possible over K using Cray Streaming
    - Long vector lengths over I dimension
- Cray is working with WRF consortium and NCAR to improve further.
WRF X1 Port

• Minimal X1 changes needed to build.
• No user configurable domain decomposition
  – Both MM5 and WRF run better on X1 with 1 dimensional decomposition (up to ~128 MSPs)
  – Increases vector lengths
  – WRF default is to make it 2D square
• Uses RSL (MPI) communications library from MM5 – X1 optimizations incorporated
  – Streaming and vector enhancements
• Reduce RSL stencil compilation overhead
  – Cut WRF init time in half
WRF v1.3 X1 Optimization

- Promote Streaming to ‘numtiles’ level with Cray csd directives, set numtiles=4
  - More evenly distributes work across MSPs
- Change order of loops in ‘Lin’ microphysics routine from J,I,K to J,K,I
  - Hand inlining
  - Promote local arrays to 2 dimensions (from K to K,I)
  - Get vectorization along full latitudes instead of shorter vertical column
  - Resulted in 2.1x performance gain
Cray streaming directives to distribute work across SSPs, num_tiles = 4

1246. 1 !csd$ parallel do private(ij)
1247. 1 !csd$& schedule(static,1)
1248. 1 M------< DO ij = 1 , grid%num_tiles
1249. 1 M
1250. 1 M I----< CALL wrf_debug ( 200 , ' call cumulus_driver' )
1251. 1 M
1252. 1 M I----< CALL cumulus_driver(itimestep,dt,DX,num_3d_m, &
1253. 1 M RTHCUTEN,RQVCUTEN,RQCCUTEN,RQRCUTEN, &
1254. 1 M RQICUTEN,RQSCUTEN,RAINC,RAINCV,NCA, &
1255. 1 M u_phy,v_phy,th_phy,t_phy,w_2,mise_2, &
1256. 1 M dz8w,p8w,p_phy,pi_phy,config_flags, &
1257. 1 M W0AVG, rho, STEPCU, &
1258. 1 M CLDEFI, LOWLYR, XLAND, CU_ACT_FLAG, warm_rain, &
1259. 1 M HTOP, HBOT, &
1260. 1 M ids, ide, jds, jde, kds, kde, &
1261. 1 M ims, ime, jms, jme, kms, kme, &
1262. 1 M grid%i_start(ij), min(grid%i_end(ij),ide-1), &
1263. 1 M grid%j_start(ij), min(grid%j_end(ij),jde-1), &
1264. 1 M k_start , min(k_end,kde-1) )
1265. 1 M
1266. 1 M------> ENDDO
1267. 1 !csd$ end parallel do
• Very vector friendly, I dimension = 600 for 1D decomposition

1222. 1------<   DO j = j_start, j_end
1223. 1
1224. 1 2-----<   DO k = kts+3, ktf-2
1225. 1 2 V----<   DO i = i_start, i_end
1226. 1 2 V       vel = 0.5 * (rom(i-1,k,j) + rom(i,k,j))
1227. 1 2 V       vflux(i,k) = vel * flux6( &
1228. 1 2 V       u(i,k-3,j), u(i,k-2,j), u(i,k-1,j), &
1229. 1 2 V       u(i,k+1,j), u(i,k+2,j), -vel)
1230. 1 2 V---->   ENDDO
1231. 1 2------>   ENDDO
1232. 1
1233. 1---------->   ENDDO
WRF 1D Decomposition

1x4 MPI decomposition, numtiles = 4

Vectorized (I dimension)

MPI and Streaming (J dimension)

MPI Halo Exchange

each MSP – 1 tile

each SSP – 1 tile

1 MPI rank
Early results for NCAR benchmark CONUS (continental US) forecast case

- 425x300, 12KM grid, 35 vertical levels
- Timestep = 72 seconds (2X typical MM5)
- Average of 72 timesteps, does not include first and last timesteps
- MSP mode
- 1D domain decomposition, numtiles=4
- 2D, 2x75, at 150 MSPs
- 75 MSPs ‘fits’ best, 300/75 = 4 latitudes per MSP, 1 latitude for each SSP
WRF v 1.3

Run in MSP mode; results plotted against # of SSPs
Spike at 300 SSPs due to benchmark having 300 latitudes

(WRF EM Core, 425x300x35, DX=12km, DT=72s)

(Feb 2004)
WRF NewConus Benchmark

WRF EM Core, 425x300x35, DX=12km, DT=72s

Diagram showing simulation speed vs. number of processors for different systems.
• Community Atmospheric Model from NCAR
• Used in CCSM
  – Usually is performance gating component
• Big vectorization effort for Cray X1 and NEC systems completed this year
• X1 performance matches Earth Simulator
CAM T42 (dev50) Performance

Simulated Years/Wall Day

Processors

Cray X1
CAM T85 (dev50) Performance

Simulated Years/Wall Day vs. Processors

- Cray X1
Summary

• Cray is executing successfully and achieving product development milestones.
• The Earth Sciences market is key for Cray’s unique range of science-driven technologies.
• Cray is actively participating and investing in this community.
• Large, experienced and dedicated Earth Sciences Team.
• Product roadmap is well positioned to meet the scientific needs of the community for many years.
Questions:

Ryan Joseph Levesque
Infant Technologist

After hearing this presentation