Porting AVS/Express to the CRAY XT-4

e-Science AHM, 14 September 2010 Cardiff
George Leaver, Martin Turner
Research Computing Services
University of Manchester

Contents

- Data Driven: Fourth Paradigm
- HECToR
- AVS/Express
- Porting / development
- User Cases

Radiographs – Creating new “insights”

"Multiscale x-ray imaging facility for monitoring and modelling structural evolution in situ" project has created the multi-scale from macro to nano Henry Moseley X-Ray Imaging Facility, £2.5m, launched in June 2009.

Motivation

- In-place visualization on supercomputer
  - Dataset size, transfer, GPU limits (if present at all)
- Viz apps for
  - Materials Science
  - Life Sciences
- Port software
  - AVS/Express
**HECToR**
- Cray XT4 (Phase2a)
  - 5664 x quad-cores: 22,656 cores (AMD 2.3GHz Opteron)
  - 8Gb / quad-core socket: 45.3Tb
  - 24 dual-core 'login' nodes
  - UKRC funded, EPCC/NAG Ltd. operated
- Distributed memory
  - Message Passing Interface (MPI)
    - `MPI_Send()`, `MPI_Recv()`, `MPI_Bcast()`, ...
  - Launch n instances of MPI app
    - `aprun -n 512 mycode`
      - (2Gb per process)

**NAG dCSE**
- NAG Distributed CSE Projects
  - Funding to port software to HECToR
  - "Massive Remote Batch Visualizer"
    - Port a commercial viz code
  - 10 months DCSE
    - May 2009 – Feb 2010
    - MS1: Port / build components
    - MS2: Viz Networks
    - MS3: Parallel IO Module
    - MS4: Optimization
    - MS5: User base testing / results

**AVS/Express**
- Rapid visualization pipeline construction
  - Visual programming
  - Easy to build GUI system
- Extensive visualization tools
- User extensible (C/C++/fortran)
- Portability
  - V code description language

**AVS/Express Editions**
- Standard
  - Serial compute & rendering
  - Parallelize at user-module level
- Multipipe Edition: parallel rendering
  - MPI: express + mpnode render processes
- Parallel Edition: parallel modules
  - MPI: express + pstnode compute processes
- Distributed Data Renderer: parallel modules & rendering
  - MPI: express + pstnode + mpnode processes

AVS/Express Parallel Edition is productized using PST developed by JAEA
Distributed Data Renderer Edition

Currently all processes in MPI job

Express(GUI) on login node?
- HECToR: No MPI allowed

Express(GUI) on compute node?
- HECToR: No X11

Remove MPI?
- Two versions of code to maintain
- Rewrite all parallel modules

Replace Vendor MPI in Express
- Forward MPI calls from login to compute node

mpiexec -n 1 express : -n 4 pstnode : -n 4 mpunode
Forwarding MPI

- Replace Cray MPI in express
  - Fake MPI library (libxpmt) on login node
  - Proxy xnode on compute node calls Cray MPI (always rank 0)
    - Maintains mapping between XPMT and Cray MPI objects, types, requests etc.

 Mapping XPMT MPI to Cray MPI

- In express: all MPI types are now integers
  - `#include<xpmt_mpi.h>` gives `typedef int MPI_Datatype;
    typedef int MPI_Comm; ...

    `MPI_Comm comm = ...; MPI_Datatype datatype = ...;
    MPI_Send( buf, count, datatype, dest, tag, comm );`
    Packs all args and sends them with `MPI_SEND` token to proxy rank 0
    via socket

- In proxy: tables of Cray MPI types, indexed by above ints
  - Token receive / Function dispatch loop sees `MPI_SEND` token
    `xMPI_Send() {`
    `recv and unpack args (buf, count, comm etc)
    MPI_Comm craycomm = comm_table_lookup(comm);
    MPI_Send(buf, count, craycomm);` — Cray MPI function called
    `send flags/results back to express }

Optimize MPI communication in renderer

- Point-to-point -> Bcast
**Other Developments 2**

- Replace HP *Paracom* Image Compositor
  - No MPI communication in paracom
  - Requires dynamic linking (wasn't available on HECToR)
  - 2-3 Swap Compositing.
  - Yu, Wang, Ma: Massively parallel volume rendering using 2-3 swap image compositing. SC08.

**User Cases**

- Materials Science
  - Henry Moseley X-Ray Imaging Facility (EPSRC, NWDA, UoF)
  - CT scanning equipment (2k x 2k x 16bit x n slices)
  - Diamond JEEP Beamline I12 (~4k x 2k x n slices)
  - 50-100Gb volumes
  - Volume render (rgba=4xRaw data), isosurface (~2x10^9 triangles)
- 351Gb Palaeontological dataset (7150x7150x7369 bytes)
  - Volume render
  - 256 processes (memory requirements)
  - ~5 FPS
  - Request not to publish images at this time
Thanks to Drs Paul Mummery, Phil Manning, Bill Sellers

george.leaver@manchester.ac.uk
james.perrin@manchester.ac.uk
martin.turner@manchester.ac.uk
http://www.manchester.ac.uk/researchcomputing