JLESC PROJECT STATUS
Developer tools for porting & tuning parallel applications

2019/04/15  I  BRIAN WYLIE
OVERVIEW

JLESC Collaborative Project (start May 2015)

• Personnel
  • JSC: Brian Wylie, Christian Feld – Scalasca/Score-P/CUBE
  • RIKEN R-CCS: Miwako Tsuji, Hitoshi Murai, Itaru Kitayama – XcalableMP
  • BSC: Judit Giménez, Germán Llort – Paraver/Extrae/Dimemas

• Goal
  • Integration and improvement of partners respective developer tools
    for porting and tuning parallel applications on large-scale computer systems
  • Joint training activities applying tools
TOOLS TRAINING

Previous period

• VI-HPS Tuning Workshops (featuring BSC & JSC tools and others)
  • #27 (23-27 Apr. 2018, LRZ, Garching, Germany) using CooLMUC-3 (KNL)
  • #29 (15-19 Oct. 2018, ROMEO, Reims, France) using ROMEO (SKX)
  • #30 (21-25 Jan. 2019, BSC, Barcelona, Spain) using MN-IV (SKX) & CTE (P9+V100)
  • #31 (9-12 Apr. 2019, UTK-ICL, Knoxville/TN, USA) using Stampede2 (KNL & SKX)

• POP training for E-CAM Centre of Excellence (8-10 Jan. 2019, TCD, Dublin, Ireland)
TOOLS TRAINING

Upcoming

• VI-HPS Tuning Workshops (featuring BSC & JSC and other tools)
  • #32 (24-26 Apr. 2019, Uni. Bristol, England)
    • using Isambard (Cray XC50 with Marvell ThunderX2 ARMv8-A cores)
  • #33 (24-28 June 2019, JSC, Jülich, Germany)
    • using JUWELS (SKX) and/or JURECA (HSW & KNL)
• Featuring individual toolsets
  • PTC Performance Analysis & Tools (20-21 May 2019, BSC, Barcelona, Spain)
  • ISC-HPC half-day tutorial (16 June 2019, Frankfurt/Main, Germany)
  • Int’l HPC Summer School (7-12 July 2019, RIKEN R-CCS, Kobe, Japan)
TOOLS DEVELOPMENTS

• BSC
  • Paraver 4.8.1, Extrae 3.6.1, Dimemas 5.4.1, etc.
• JSC
  • Score-P 5.0, OPARI2-2.0.4, OTF2-2.1.1, CUBE-4.4.x, Scalasca-2.5
• RIKEN R-CCS
  • Omni XMP
New features

- Uninstrumented (orphan) threads
  - e.g. threads created by POSIX-based runtimes (std::thread, Intel TBB, etc.)
- Cartesian topologies
  - MPI Cartesian
- BG/Q & K hardware
- Process x Thread
- User-defined
MODULAR SUPERCOMPUTING SUPPORT

Heterogeneous application execution on JURECA Cluster (HSW) and Booster (KNL)

• Combination of application heterogeneity with heterogeneous computer system
  • cf. multi-executable/MPMD, Intel symmetric and accelerator offload (CPU+GPU)
• Challenge
  • different processors/cores (2.5 vs 1.4 GHz cycle frequencies, IPC)
  • different configurations of processes/threads per compute node
    • e.g., 2x12x2 on Cluster, 4x17x4 on Booster
  • identification/distinction of each part and their interaction
PROFILE PRESENTATION

B8x64x2+C4x16x2

73% Computation
13% MPI, 13% IdleThreads

mainFunction with 200 steps mostly do_md
some gmx_pme_do

Booster ranks 0..511
on 8 nodes jrc5156..jrc5163
Cluster ranks 512..575
on 4 nodes jrc0372..jrc0375
(purple box)
PROFILE PRESENTATION

Complementary presentations of thread values

System tree presentation of 8x64x2 threads on Booster & 4x16x2 threads on Cluster

Statistics plot shows range and variation of metric values

Sunburst ring view
PROFILE PRESENTATION

Booster & Cluster computation parts
During 200 steps, blocking MPI_Recv LateSender by Cluster processes
Efficiency of do_md is very good (on Booster nodes)

Booster execution

Efficiency of do_md is very good (on Booster nodes)
Efficiency of gmx_pme_do is very poor (on Cluster nodes)
Full execution of 200 steps with marked initialization and finalization phases.

Booster ranks 0..511
Cluster ranks 512..575 (purple boxes)

MPI = red
Comp = green
Zoom into mid region of trace (with 15 steps)

Generally steps are very similar, however, every 12th step has some additional computation

Computation on Cluster nodes is comparatively short
Close-up of 3 steps

Both Cluster and Booster computations well balanced over MPI ranks

But Cluster computation is much shorter, and nodes mostly wait in MPI_Recv calls
EXECUTION TRACE VISUALIZATION (THREAD DETAILS)

Expanded view of OpenMP threads

Parallel regions also well balanced within Cluster and Booster

However, Cluster worker threads idle while master thread executes MPI_Send
SCALASCA/SCORE-P MODULAR SUPPORT

Measurement & analysis constraints

• MPI_COMM_WORLD encompassing all processes (no MPI_Comm_spawn/connect/etc)
• All executables need to be instrumented in the same manner (MPI or MPI+X)
  • routines in source files compiled from different locations/paths will be shown distinctly
• Measurement configurations need to be compatible
  • all summarization and/or tracing, identically-defined hardware counters
• ‘Manual’ launch of application executables (and trace analyzer)
  • no automated setting of environment for each part
• Global timer (e.g., gettimeofday) rather than TSC cycles
• Common file-system for shared measurement directory
• …
SUMMARY

- Modular supercomputing offers many configuration optimisation possibilities
  - which sections of application to execute on each part?
  - how many compute nodes of each type?
  - how many processes/threads per compute node?
- Complex executions and interactions need performance measurement and analysis tools
  - analyses and visualizations are challenging