

10h-

Center



F.B. 28TH

Martin Čuma Center for High Performance Computing University of Utah m.cuma@utah.edu

# OF UTAH<sup>TT</sup> SW development tools



- Development environments
- Compilers
- Version control
- Debuggers
- Profilers
- Runtime monitoring
- Benchmarking





# **PROGRAMMING TOOLS**

EEB. 28TH

8/5/2016



## **Program editing**



• Text editors

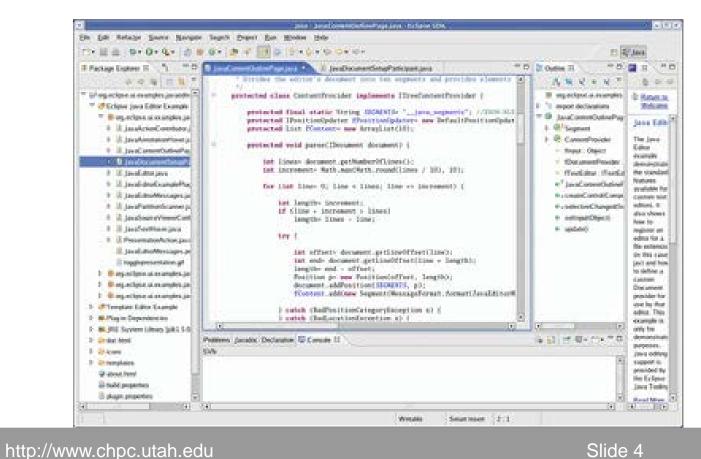
#### – vim, emacs, atom

| 🕻 iofunc_grmag.c (/uufs/chpc.utah.edu/com81/zhdanov/grmag/jun1  | .6/src/grmag) - GV 👘 💷 🗮 🕷                 |
|---|--|
| <u>File Edit T</u> ools <u>S</u> yntax <u>B</u> uffers <u>W</u> indow <u>H</u> elp  |  |
| 🕒 🖄 🕞 👍 i 🥱 🖉 i 🐰 🖫 🖺 i 🗛 🔶 i 😂 😭   | 🐉   🍓 🗉 🐟   👩                              |
| <pre>invdp[ip].mv[ii].Ho = Hos;     }     fscanf(fp,"\n");     }     fclose(fp); } MPI_Bcast(obsdata,invdp-&gt;Nrect,MPI_PFLOAT,0,MPI_COMM_WORLD); MPI_Bcast(invdp-&gt;rx,3*invdp-&gt;Nrec,MPI_DFLOAT,0,MPI_COMM_WORLD); if(invdp[ip].Hos&lt;0) {     MPI_Bcast(invdp-&gt;mv,5*invdp-&gt;Nrec,MPI_DFLOAT,0,MPI_COMM_WORLD);     return(0); } int write_par(char *fname, Pfloat *mfits, Pfloat *parfs, Pfloat *alps,     Pfloat *stabs, int n)</pre> | Ĩ  |
| <pre>Prioat *stabs, int n) {</pre>  |  |
|   |  |
| <pre>iof unc_grmag.c<br/>// first property is gravity (ndim=1)<br/>// 2nd property is either suceptibility (ndim=1) or mv (ndim=3)<br/>// stabilizer derivative (m*pdot) - separate for each vector component<br/>for (idp=0;idp<invdp[1].ndims;idp++)<br>{<br/>ii = idp*invd.Nxyz;<br/>#pragma omp parallel for private(i) if (invd.Nxyz &gt; MINLOOP)<br/>for [i=0;icinvd.Nxyz;i++]<br/>{</invdp[1].ndims;idp++)<br></pre>                        |  |
| <pre>// first property is gravity (ndim=1)<br/>// 2nd property is either suceptibility (ndim=1) or mv (ndim=3)<br/>// stabilizer derivative (m*pdot) - separate for each vector component<br/>for (idp=0;idp<invdp[1].ndims;idp++)<br>{<br/>ii = idp*invd.Nxyz;<br/>#pragma omp parallel for private(i) if (invd.Nxyz &gt; MINLOOP)<br/>for [i=0;i<invd.nxyz;i++)<br>{</invd.nxyz;i++)<br></invdp[1].ndims;idp++)<br></pre>                         | ▲<br>idp][0])/Wm[0][i];<br>])/Wm[1][i+ii]; |
| <pre>// first property is gravity (ndim=1) // 2nd property is gravity (ndim=1) or mv (ndim=3) // stabilizer derivative (m*pdot) - separate for each vector component for (idp=0;idp<invdp[1].ndims;idp++) #pragma="" (invd.nxyz="" for="" if="" ii="idp*invd.Nxyz;" omp="" parallel="" private(i)="" {=""> MINLOOP)     for [=0i<invd.nxyz;i++] (<="" td="" {=""><td>▲<br/>idp][0])/Wm[0][i];</td></invd.nxyz;i++]></invdp[1].ndims;idp++)></pre>   | ▲<br>idp][0])/Wm[0][i];                    |

8/1/2016

## IDEs

## - Visual \*, Eclipse



## Compilers



F.B. 28TH



- Open source
  - GNU
  - Open64, clang
- Commercial
  - Intel
  - Portland Group (PGI, owned by Nvidia)
  - Vendors (IBM XL, Cray)
  - Others (Absoft, CAPS, PathScale)

28TH



## Language support



- Languages
  - C/C++ GNU, Intel, PGI
  - Fortran GNU, Intel, PGI
- Interpreters
  - Matlab has its own ecosystem
  - Java reasonable ecosystem, not so popular in HPC, popular in HTC
  - Python attempts to have its own ecosystem, some tools can plug into Python (e.g. Intel VTune)



## Language/library support



- Language extensions
  - OpenMP (4.0+\*) GNU, Intel\*, PGI
  - OpenACC PGI, GNU very experimental
  - CUDA Nvidia GCC, PGI Fortran
- Libraries
  - Intel Math Kernel Library (MKL)
  - PGI packages open source (OpenBLAS?).

lance

#### THE UNIVERSITY OF UTAH<sup>™</sup>

# Version control

- Copies of programs
  - Good enough for simple code and quick tests/changes
- Version control software
  - Allow code merging, branching, etc
  - Essential for collaborative development
  - -RCS, CVS, SVN
  - Git integrated web services, free for open source, can run own server for private code





# DEBUGGING

FEB. 28TH

8/1/2016

## Program errors



- Crashes
  - Segmentation faults (bad memory access)
    - often writes core file snapshot of memory at the time of the crash

E.B. 28TH

- Wrong I/O (missing files)
- Hardware failures
- Incorrect results
  - Reasonable but incorrect results
  - NaNs not a numbers division by 0, ...

Center

for High-

nance

#### THE UNIVERSITY OF UTAH<sup>™</sup>

## write/printf

- Write variables of interest into the stdout or file
- Simplest but cumbersome
  - Need to recompile and rerun
  - Need to browse through potentially large output

| oot@p8:/uu | /s/cnpc.utan.edu/sys/build | ldir/oc 💥 u0101881@p8:~/tests |  |
|------------|----------------------------|-------------------------------|--|
| 81         | 0.80500000000000005        | 272.32451601382030            |  |
| 82         | 0.815000000000000006       | 274.72803717231750            |  |
| 83         | 0.82500000000000007        | 277.10810411170019            |  |
| 84         | 0.834999999999999996       | 279.46489239846238            |  |
| 85         | 0.844999999999999997       | 281.79858064688352            |  |
| 86         | 0.854999999999999998       | 284.10935027759365            |  |
| 87         | 0.864999999999999999       | 286.39738528452926            |  |
| 88         | 0.875000000000000000       | 288.66287201019298            |  |
| 89         | 0.88500000000000001        | 290.90599892911797            |  |
| 90         | 0.89500000000000002        | 293.12695643942459            |  |
| 91         | 0.90500000000000003        | 295.32593666234624            |  |
| 92         | 0.91500000000000004        | 297.50313324959058            |  |
| 93         | 0.92500000000000004        | 299.65874119839486            |  |
| 94         | 0.93500000000000005        | 301.79295667412481            |  |
| 95         | 0.945000000000000006       | 303.90597684026102            |  |
| 96         | 0.95500000000000007        | 305.99799969561070            |  |
| 97         | 0.964999999999999997       | 308.06922391857796            |  |
| 98         | 0.974999999999999998       | 310.11984871832163            |  |
| 99         | 0.9849999999999999999      | 312.15007369262662            |  |
| 100        | 0.995000000000000000       | 314.16009869231254            |  |

F.B. 28TH

Center

nance

#### THE UNIVERSITY OF UTAH<sup>™</sup>

## Terminal debuggers

- Text only, e.g. gdb, idb
- Need to remember commands or their abbreviations
- Need to know lines in the code (or have it opened in other window)
- Useful for quick code checking on compute nodes and core dump analysis

|   | Terminal - u0101881@p8:~/tests  | $1 \times$ |
|---|---|------------|
|   | root@p8:/uufs/chpc.utah.edu/sys/builddir/oc 💥 u0101881@p8:~/tests   | ×          |
| J | <pre>Starting program: /uufs/chpc.utah.edu/common/home/u0101881/tests/pi3 warning: File "/opt/at9.0/lib64/power8/libthread_db-1.0.so" auto-loading has been declined b your `auto-load safe-path' set to "\$debugdir:\$datadir/auto-load". To enable execution of this file add</pre> |            |
|   | Breakpoint 1, MAIN () at pi3.f:50<br>50   |            |
|   | (gdb) p n<br>\$1 = 100000<br>(adb) □  |            |

iance



## GUI debuggers

Center for High-Perform Computing

B. 28TH

- Have graphical user interface
- Some free, mostly commercial
- Eclipse CDT (C/C++ Development Tooling), PTP (Parallel Tools Platform) - free
- PGI's pdbg part of PGI compiler suite
- Intel development tools
- Rogue Wave Totalview commercial
- Allinea DDT commercial







- The only real alternative for parallel or accelerator debugging
- Cost a lot of money (thousands of \$), but, worth it
- We have Totalview license (for historical reasons), 32 tokens enough for our needs (renewal ~\$1500/yr).
- XSEDE systems have DDT.

## How to use Totalview





- 1. Compile binary with debugging information
- flag -g

gcc -g test.f -o test

- 2. Load module and run Totalview module load totalview
- TV + executable

totalview executable

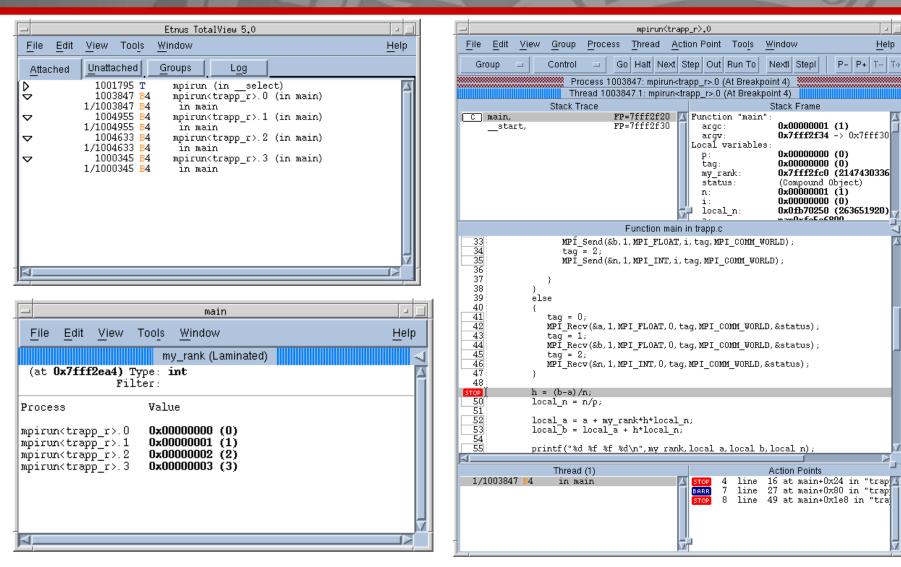
- TV + core file
   totalview executable core\_file
- Run TV and choose what to debug in a startup dialog totalview

#### THE UNIVERSITY OF UTAH<sup>™</sup>

## Totalview windows



<sup>for</sup> High-Performance <sup>Computing</sup>



## UNIVERSITY OF UTAH<sup>™</sup>

# DDT screenshot



center <sup>for</sup> High-Performance computing

| <pre>if (pe == 0) (     loops = 5; /* times round ring */     MPI_Send(&amp;loops, 1, MPI_INT, to, tag, MPI_COMM_WORLD);     MPI_Send(&amp;loops, 1, MPI_INT, to, tag, MPI_COMM_WORLD);     vhile (1) {         Type: int         if (pe == 0) loops;         /* delaying tactics */         a=-2.2;         /* Type: int         // Type: int         //</pre>  | ent Group: All 🔶  | Focus on current: 🛞 Group | O Process O Thread 🗌 Step Threads Together   |   |   |  |
|--|---|---------------------------|--|---|---|--|
| tt Flos () () () () () () () () () () () () ()   |   |                           |  |   |   |  |
| tring.c X       T promm_marks: X       T precve: X         Application Code       Int pe, nproce, tag, to, from, loope;<br>floet a;<br>int i;<br>MPT Status status;       Int pe, nproce, tag, to, from, loope;<br>floet a;<br>int i;<br>MPT Status status;       Int pe, nproce, tag, to, from, loope;<br>floet a;<br>int i;<br>MPT Status status;         Image: Sources       Int pe, nproce, tag, to, from, loope;<br>from ank(MED)       Int pe, nproce, tag, to, from, loope;<br>floet a;<br>int i;<br>MPT Status status;       Int pe, nproce, tag, to, from, loope;<br>floet a;<br>int i;<br>MPT Status status;         Image: Status status;       Int pe, nproce, tag, to, from, loope;<br>floet a;<br>int i;<br>MPT Status status;       Int pe, nproce, tag, to, from, loope;<br>floet a;<br>int i;<br>MPT Status status;         Image: Status status;       Int pe, nproce, tag, to, from, loope;<br>from = (pe + nproce, fargv);<br>if tag=1;<br>if (pe = 0) {<br>if (pe = -0) {<br>if (                  |   | urrently selected:        | 0 (on david-laptop, pid 24887, main thread   | IWP 24887)  |   |  |
| <pre>h (Chri+K)  Appl(zation Code / Jost s; Int pe, sprocs, tag, to, from, loops; Itot s; Int i; Itot s; Int i; Int pe, sprocs, tag, to, from, loops; Itot s; Int i; Int i</pre>   |   | Tring c M. Transmi        |  | ſ   | Locals Current Line   | current Stark  |
| Application Code       Application Code       Filest s:         /       int i;       int i;         /       MPT Status status;   |   |                           |  | G   | and the second se | S) Current Stock   |
| Application Code       iii       iiii       iiii       iiii       iiii       iiiii       iiiiiiii       iiiiiiiiii       iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii  | h (Ctrl+K)  | A float a;                | ca, cay, co, riva, roopa,  | the second se |   |  |
| <pre>Sources Sources S</pre>   |   |                           | 202303   | -   |   |  |
| if ing.c       if ing.c       if ing.c       if ing.c         if ing.c       if ing.c       if ing.c <td></td> <td></td> <td>Latus;</td> <td></td> <td>k:/50/503</td> <td></td>  |   |                           | Latus;   |   | k:/50/503   |  |
| Immediate and<br>External Code       III<br>INPT Comm rank (MPI Code MORLD, &pa);<br>MPT Comm_rank (MPI Code MORLD, &pa);<br>MPT Comm_rank (MPI Code MORLD, &pa);<br>MPI Comm_size (MPI_Code MORLD, &pa);<br>MPI_comm_size (MPI_Code MORLD, &pa);<br>MPI_comm_size (MPI_Code MORLD, &pa);<br>MPI_comm_size (MPI_code MORLD, &pa);<br>MPI_status;<br>MPI_status;<br>MPI_status;<br>MPI_status;<br>MPI_status;<br>MPI_status;<br>MPI_status;<br>MPI_mecv(kloops, 1, MPI_INT, to, tag, MPI_COMM_MORLD, &status);<br>MPI_mecv(kloops, 1, MPI_INT, from, tag, MPI_comm_MorLD, &status);<br>MPI_mecv(kloops, 1, MPI_mecv(kloops, 1, MPI_INT, from, tag, MPI_comm_MorLD, &status);<br>MPI_mecv(kloops, 1, MPI_INT, from, tag, MPI_comm_MorLD, &status);       MPI_mecv(kloops, 1, MPI_mecv(kloops, 1, MPI_INT, from, tag, MPI_comm_MorLD, &status);  |   | 12 tag=1;                 |  |   | 12/20/2020  |  |
| External Code       15       MpT Comm_sank (MPI_cover WORLD, Spe);<br>MPI_Comm_sine MPI_Cover WORLD, Snproce);<br>MPI_Comm_sine MPI_Cover WORLD, Snproce);<br>To = (pe + 1) % nproce;<br>from = (pe + nproce - 1) % nproce;<br>from = (pe + nproce - 1) % nproce;<br>from = (pe + nproce - 1) % nproce;<br>MPI_SOURCE = -11658<br>// Dops = 5; /* times round ring */<br>MPI_SOURCE = -11658<br>// MPI_SOURCE = -11658<br>// MPI_SOURCE = -11658<br>// MPI_SOURCE = -11658<br>// MPI_SOURCE = -11658         if (pe == 0) (<br>Loops = 5; /* times round ring */<br>MPI_Send(Sloops, 1, MPI_INT, to, tag, MPI_COMM_WORLD);<br>// delaying tactice */<br>a=-2.2;       MPI_Comm_senk(MPI_cover, MORLD);<br>/* delaying tactice */<br>a=-2.2;         if (pe == 0) loope;<br>/* delaying tactice */<br>a=-2.2;       Image from 0 to<br>1/128 processes  |   |                           |  |   | 1   | 127  |
| # Headors       # HP1_Com_sire (MP1_COMM_WORLD, & sproce);       5         Sources       if       hprocs;       5         if       to = (pe + 1) % nprocs;       ppe       0         if       if (pe == 0) (       loops = 5; /* times round ring */       MP1_SOURCE = -11658         if       if (pe == 0) (       loops = 5; /* times round ring */       MP1_DOMM_WORLD);       MP1_ERROR         if       (pe == 0) (       loops = 1, NP1_INT, to, tag, MP1_DOMM_WORLD);       MP1_ERROR       4197477         if       (pe == 0) loops;       if       (pe == 0) loops;       1         if       (pe == 0) loops;       if       (pe == 0) loops;       1         if       (pe == 0) loops;       if       (pe == 0) loops;       1         if       (pe == 0) loops;       if       (pe == 0) loops;       1         if       (pe == 0) loops;       if       (pe == 0) loops;       1/128 processes         it/Output       Breakpoints       Watchpoints       Tacepoint Output       Logbook       Evaluate   |   | HP1 Init(&dc              |  |   | 1   | -11696   |
| <pre>If Headers Sources If to = (pe + 1) % nprocs; from = (pe + 1) % nprocs; from = (pe + 1) % nprocs; if (pe == 0) ( Loops = 5; /' times round ring */ MPI_SOURCE = 0) ( Loops = 5; /' times round ring */ MPI_SOURCE = 0) ( Loops = 5; /' times round ring */ MPI_SOURCE = 0) ( Loops = 5; /' times round ring */ MPI_SOURCE = 0) ( Loops = 5; /' times round ring */ MPI_SOURCE = 0) ( Loops = 0) ( Loop</pre>   |   |                           |  |   | - loops   | 5  |
| <pre>tyOutput Breakpoints Watchpoints Stacks Pacepoints Tracepoint Output Logbook</pre>  |   | 17                        | and the second |   | - nprocs  |  |
| <pre>if (pe == 0) {     loops = 57 /* times round ring */     MPI_Source ==1165     if (pe == 0) {         loops = 57 /* times round ring */         MPI_Send(&amp;loops, 1, NPI_INF, to, tag, NPI_COMM_WORLD);         // MPI_ERROR 4197477         // cancelled 0         // ucount 15774463         // agg == 1         // MPI_Recv(&amp;loops, 1, NPI_INF, from, tag, MPI_COMM_WORLD, &amp;status);         // Type: int         // Agg == 1         // 11         // Type: int         // Agg == 1         // 128 processes         // Type: int         // Type: int         // Status         // Statu</pre> | E Sources   | 18 to - (pe + 1           |  |   | - pe  | -0   |
| <pre>if (pe == 0) {     Loops = 57 /* times round ring */     MPI_Send(&amp;loops, 1, NPI_INT, to, tag, MPI_COMM_WORLD);     // MPI_Recv(&amp;loops, 1, NPI_INT, from, tag, MPI_COMM_WORLD);     // MPI_Recv(&amp;loops, 1, NPI_INT, from, tag, MPI_COMM_WORLD, &amp;status);     if (pe == 0) loops;     /* delaying tactics */     a=2.2;     // delaying ta</pre>                               |   |                           | nprocs - 1) % nprocs;  | 1   | status  | (MPI_SOURCE = -11658, MPI_TA   |
| MPT_Send(&loops, 1, MPT_INT, to, tag, MPT_COMM_WORLD);<br>MPT_Send(&loops, 1, MPT_INT, to, tag, MPT_COMM_WORLD);<br>while (1) {<br>27<br>MPT_Recv(&loops, 1, MPT_INT, from, tag, MPT_COMM_WORLD, &status);<br>16<br>17<br>17<br>17<br>17<br>17<br>17<br>17<br>17<br>17<br>17   |   | 21 [ if  pe == 0)         |  |   |   |  |
| 24     24     24     25     cancelled     0       25     while (1) {   |   |                           |  | c   |   | South and the state of the stat |
| vhile (1) {  |   |                           | loops, 1, MPI_INT, to, tag, MPI_COMM_WORLD)  |   |   |  |
| 26 B       while (1) {       Isyn4463         27       MPI Recv(kloops, 1, MPI INT, from, tag, MPI COMM MORLO, kstatus);       Itag         1       if (pe == 0) loops;       1         30       /* delaying tactics */       1         31       a=2,2;       Type: int         1/Uoutput       Breakpoints       Watchpoints       Tacepoint Output   |   |                           |  |   | 1. March 2011 (2011) 12 (2011)  | 0.5%   |
| If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops:     If (pe == 0) loops:       If (pe == 0) loops-::     If (pe == 0) loops-::       If (pe == 0) loops-::     If (pe == 0) loops-::       If (pe == 0) loops-::     If (pe == 0) loops-::       If (pe == 0) loops-::     If (pe == 0) loops-::       If (pe == 0) loops-::     If (pe == 0) loops-::       If (pe == 0) loops-::     If (pe == 0) loops-::       If (pe == 0) loops-::     If (pe == 0) loops-:: <t< td=""><td></td><td>26 g while (1) {</td><td>Y</td><td>1</td><td>kooset.</td><td></td></t<>   |   | 26 g while (1) {          | Y  | 1   | kooset.   |  |
| if (pe == 01 loops:     Type: int       if (pe == 01 loops:     ************************************   |   |                           | loops, 1, MPI INT, from, tag, MPI COMM WORL  | D, éstatus);  | to  |  |
| 30     31     /* delaying tactics */     Type: int       1     32     a-2.2;     Type: int       it/Output     Breakpoints     Watchpoints     Stacks       Tacepoint Output     Logbook     Evaluate  |   |                           | 0) loops;  |   | and a second  | ~ 1-   |
| It/Output Breakpoints Watchpoints Stacks Tracepoint Output Logbook Evaluate  |   | 30                        | and the second   |   |   |  |
| t/Output Breakpoints Watchpoints Stacks Tracepoint Output Logbook Evaluate   |   | 1 A                       | g tactics */   |   |   | Range from 0 to 127  |
|  |   | a=2.24                    | 111.03   |   | ype: int  | They be dealed address   |
|  |   | Ulatchooiste Starke       | Tracepoints Tracepoint Output Logbook  |   |   | 1  |
|  | t/Output Breakpoints  | wateripoints statuts      |  | 10  | ALL CONTRACTOR CONTRACTOR   |  |
| sses Function +  | t/Output Breakpoints  | wateripoints status       |  |   | in loops 5  |  |
| e main (ring.c:27)   | t/Output Breakpoints<br>s   | Watenpoints Starks        | ¥.   |   | 2,560   |  |
| E PMPI_Recv (precv.c:78)   | t/Output Breakpoints<br>s<br>sses Function                                      |                           | •  |   | 0.000   |  |
| E mca_pml_obl_recv(pml_obl_irecv.c:105)<br>E ompi request wait completion (request.h:377)  | t/Output Breakpoints<br>s<br>sses Function<br>= main (monoce)<br>= PMPL Recy (p | 7)<br>recv.c:78)          |  |   |   |  |

Debugger basic operations

- Data examination
- view data in the variable windows
- change the values of variables
- modify display of the variables
- visualize data
- Action points
- breakpoints and barriers (static or conditional)
- watchpoints
- evaluation of expressions

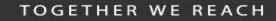
TOGETHER WE REACH

Center

for High-

nance

F.B. 28TH

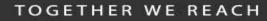


Multiprocess debugging

For High-Performance Computing

- Automatic attachment of child processes
- Create process groups
- Share breakpoints among processes
- Process barrier breakpoints
- Process group single-stepping
- View variables across procs/threads
- Display MPI message queue state

## Additional Totalview tools





- Memoryscape
  - Dynamic memory debugging tool
- Replay Engine
  - Allows to reversely debug the code
- Accelerator debugging
  - CUDA and OpenACC





- Compilers check for syntax errors
  - lint based tools
  - Runtime checks through compiler flags (-fbounds-check, -check\*, -Mbounds)
- DDT has a built in syntax checker
  - Matlab does too
- Memory checking tools many errors are due to bad memory management
  - -valgrind easy to use, many false positives
  - Intel Inspector intuitive GUI



Intel software development products





- We have a 2 concurrent user license
  - One license locks all the tools
  - Cost ~\$2000/year
- Tools for all stages of development
  - Compilers and libraries
  - Verification tools
  - Profilers
- More info

https://software.intel.com/en-us/intel-parallel-studio-xe

281

for High-

nance

#### THE UNIVERSITY OF UTAH<sup>™</sup>

# Intel Inspector

- Thread checking
  - Data races and deadlocks
- Memory checker
  - Like leaks or corruption
  - Good alternative to Totalview MemoryScape
- Standalone or GUI integration
- More info

http://software.intel.com/en-us/intel-inspector-xe/

#### THE UNIVERSITY OF UTAH<sup>™</sup>

# Intel Inspector



- Source the environment module load inspectorxe
- Compile with -tcheck -g ifort -openmp -tcheck -g trap.f
- Run tcheck

inspxe-gui – graphical user interface inspxe-cl – command line

## Tutorial

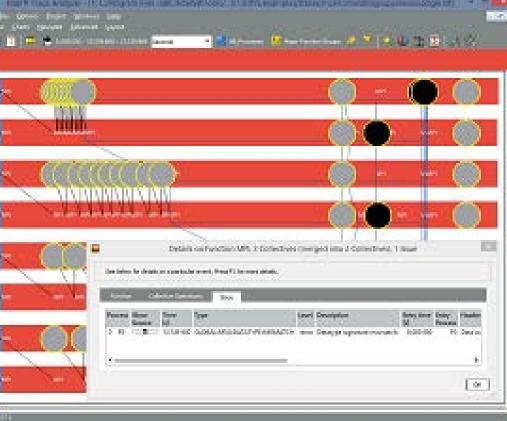
https://software.intel.com/en-us/articles/inspectorxe-tutorials

| 🧾 Dete      | el Ins                             | spector | XE 2016                |         |                          |         |
|-------------|------------------------------------|---------|------------------------|---------|--------------------------|---------|
| 🛛 🕀 Targ    | et Å Analysis Type 🛃 Collection    | Log     | Summary                |         |                          | D       |
| Problems    |                                    |         |                        |         |                          | 8       |
| ID 🔺 🔍      | Туре                               | Sourc   | es                     | State   | N                        | Aodules |
| 🗄 P1  🙆     | Mismatched allocation/deallocation | find_a  | and_fix_memory_errors. | P Co    | nfirmed fi               | ind_and |
| 🗄 P2 🛛 🔕    | Memory leak                        | find_a  | and_fix_memory_errors. | P De    | ferred fi                | ind_and |
| 🖽 P3 🛛 🚳    | Invalid memory access              | find_a  | and_fix_memory_errors. | 降 Ne    | w f                      | ind_and |
| 🗄 P4 / 🛆    | Memory not deallocated             | 隆 Ne    | w fi                   | ind_and |                          |         |
| ₫ 1         | 1of4 D                             | All     | Code Locations: Inval  | id mem  | iory access              | 8       |
| Description | Source Fund                        | tion    | Module                 |         | Object Size              | Offset  |
| Write       | find_and_fix_memory_errors oper    | ator()  | find_and_fix_memory_   | errors  |                          |         |
| 164         | for (unsigned int i=0;             | i<=(m   | boxsize/(sizeof(un     |         | ind_and_f                | _       |
| 166         | <pre>local_mbox[i]=0;</pre>        |         |                        |         | ind_and_f                | _       |
| 167<br>168  | for (int y = r.begin()             |         |                        |         | ind_and_f:<br>bb_debug.( | _       |

# MPI profiler and correctness checker Detects violations of MPI standard

- and errors in execution environment
- To use correctness checker module load intel impi itac setenv VT\_CHECK\_TRACING 0 mpirun -check-mpi -n 4 ./myApp
- ITAC documentation

https://software.intel.com/en-us/intel-trace-analyzersupport/documentation







# PROFILING

EEB. 28TH

8/1/2016

E.B. 28TH

#### THE UNIVERSITY OF UTAH<sup>™</sup>

## Why to profile

For High-Performance Computing

• Evaluate performance

- Find the performance bottlenecks
  - Inefficient programming
  - Memory or I/O bottlenecks
  - Parallel scaling



## • Time program runtime

NIVERSITY

OF UTAH"

- get an idea on time to run and parallel scaling

- Many programs include benchmark problems
  - Some also accessible via "make test"
- Consider scripts, especially if doing parallel performance evaluation

| Terminal - u0101881@p8:~/tests  | + _ □ × |
|---|---------|
| root@p8:/uufs/chpc.utah.edu/sys/builddir/oc 💥 u0101881@p8:~/tests   | ×       |
| <pre>u0101881@p8 ~/tests]\$ mpicc -03 cpi.c -o cpi<br/>u0101881@p8 ~/tests]\$ time mpirun -np 16 ./cpi<br/>vi is approximately 3.1415926535898451, Error is 0.000000000000000520<br/>vall clock time = 2.338931<br/>vocess 0 before finalize<br/>7.959u 0.194s 0:02.40 1589.1% 0+0k 0+0io 0pf+0w<br/>u0101881@p8 ~/tests]\$ time mpirun -bind-to numa -map-by numa -np 16 ./cpi<br/>vi is approximately 3.1415926535898451, Error is 0.0000000000000000000000000000000000</pre> |         |

Program runtime



Center

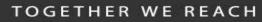
ance

#### UNIVERSITY OF UTAH<sup>™</sup>

## Profiling categories

- Hardware counters
  - count events from CPU perspective (# of flops, memory loads, etc)
  - usually need Linux kernel module installed (>2.6.31 has it)
- Statistical profilers (sampling)
  - interrupt program at given intervals to find what routine/line the program is in
- Event based profilers (tracing)
  - collect information on each function call

## Hardware counters





- CPUs include counters to count important events
  - Flops, instructions, cache/memory access
  - Access through kernel or PAPI (Performance Application
  - Programming Interface)
- Tools to analyze the counters
  - perf hardware counter collection, part of Linux
  - oprofile profiler + hw counters
  - Intel VTune
- Drawback harder to analyze the profiling results (exc. VTune)

|    | nonnance /              | Application               |      |                         |          |          |
|----|-------------------------|---------------------------|------|-------------------------|----------|----------|
|    |                         | Terminal - u0101881       | @p8  | :~/tests                |          | ↑ _ □ X  |
|    | root@p8:/uufs/chpc.ut   | ah.edu/sys/builddir/oc 🔀  | u01  | 01881@p8:~/tests        |          | ×        |
|    | [u0101881@p8 ~/tests]\$ |                           |      |                         | cpi      | <u></u>  |
|    | Performance counter s   | tats for 'mpirun -bind-to | core | -map-by core -np 1 ./cp | i':      |          |
|    | 5176.211056             | task-clock (msec)         | #    | 0.999 CPUs utilized     |          |          |
|    | 78                      | context-switches          | #    |                         |          |          |
|    | 3                       | cpu-migrations            | #    |                         |          |          |
|    | 4,255                   | page-faults               | #    |                         |          |          |
|    | 17,641,344,721          | cycles                    | #    | 3.408 GHz               |          | (66.76%) |
|    | 16,802,571              | stalled-cycles-frontend   |      |                         |          | (50.17%) |
|    | 12,934,161,831          |                           | #    |                         |          | (50.16%) |
|    | 13,393,245,960          | instructions              | #    |                         |          |          |
|    |                         |                           | #    |                         | per insn | (66.78%) |
|    | 1,083,269,385           |                           |      | 209.278 M/sec           |          | (49.84%) |
| ~\ | 1,479,078               | branch-misses             | #    | 0.14% of all branches   |          | (50.10%) |
| e) | 5.180999586 sec         | onds time elapsed         |      |                         |          |          |
|    | [u0101881@p8 ~/tests]\$ |                           |      |                         |          | v        |



# Serial profiling

Center for High-Performance Computing

- Discover inefficient programming
- Computer architecture slowdowns
- Compiler optimizations evaluation
- gprof
- Compiler vendor supplied (e.g. pgprof, nvvp)
- Intel tools on serial programs
   AdvisorXE, VTune

|                      |                         |                        |  |              |         |   | PGPROF (on sbe01) |             |                    |                       |                  |
|----------------------|-------------------------|------------------------|--|--------------|---------|---|-------------------|-------------|--------------------|-----------------------|------------------|
| <u>F</u> ile ⊻i      | iew <u>W</u> indov      | v <u>R</u> un <u>H</u> | elp  |              |         |   |                   |             |                    |                       |                  |
| 9 🔛 🛛                | i 💷 🗣 🔍                 | -   🕀 🤆                | <b>⊇                                    </b> | " 🥆 🛛 🔣 📮    |         |   |                   |             |                    |                       |                  |
| 👃 *swir              | m-acc-data.             | prof 🛙                 |  |              |         |   |                   |             |                    |                       | - 1              |
|                      |                         |                        | 0 s  |              | 1 s     |   | 2 s               | 3 s         |                    | 4 s                   | 5 s              |
| Proce                | ess "swim-acc           | -data" (23.            |  |              |         |   |                   |             |                    |                       |                  |
| 🖃 Thr                | read 1776556            | 288                    |  |              |         |   |                   |             |                    |                       |                  |
| ∟ (                  | OpenACC                 |                        |  |              |         |   | acc               |             |                    |                       |                  |
| L [                  | Driver API              |                        |  |              |         |   |                   |             |                    |                       |                  |
|                      | ofiling Overhe          | ad                     |  |              |         |   |                   |             |                    |                       |                  |
|                      | esla K40c               |                        |  |              |         |   |                   |             |                    |                       |                  |
|                      | ontext 1 (CUD/          |                        |  |              |         |   |                   |             |                    |                       |                  |
|                      | 🍸 MemCpy (H             |                        |  |              |         |   |                   |             |                    |                       |                  |
|                      | TMemCpy (D              | /toH)                  |  |              |         |   |                   |             |                    |                       |                  |
|                      | Compute                 |                        |  |              |         |   |                   |             |                    |                       |                  |
|                      | Streams                 | _                      | -  |              |         |   |                   |             |                    |                       |                  |
| a Analy              | ysis 🛅 GPU              | Det 🛿 🥊                | Console                                      | e 📑 Settings |         |   | 🛅 CPU Details 🛿   | <b>1</b>    | 1 🖫 🗖 🗖            | □ Properties 🛛        | -                |
|                      |                         |                        |  | E 4          | • 🖾     | ~ | Thread 0 🔻        |             |                    | Memcpy HtoD [async]   |                  |
| Name                 | Start Time              | Duration               | Grid Size                                    | Block Size   | Regs    | ^ | Event             | %           | Time               | Start                 | 1.574 s (        |
| Memc                 | .65.759 ms              | 3.104 µs               | n/a  | n/a          | n/a     |   | z main            | 97.597%     | 5.736 s            | End                   | 1.576 s (        |
| Memc                 | .66.058 ms              | 2.944 µs               | n/a  | n/a          | n/a     |   | ✓ main ✓ MAIN     | 97.597%     | 5.736 s            | Duration              | 1.591 m          |
| Memc                 | .66.334 ms              | 2.848 µs               | n/a  | n/a          | n/a     |   | -                 |             |                    | Size                  | 16.777 N         |
|                      | .66.596 ms              | 2.88 µs                | n/a  | n/a          | n/a     |   | swim_mod_calc2_   | 25.878%     | 1.521 s            | Throughput            | 10.548 0         |
| Memc                 | 1                       | 3 008 115              | n/a  | n/a          | n/a     |   |                   |             | 1.521 s            | Stream                | Stream 1         |
| 1                    | .66.853 ms              | 5.000 µ5               |  | 1            | n/a     |   | ✓ _pgi_uacc_cuda  | -           | 1.521 s            | ✓ Memory Type         |                  |
| Memc                 | .66.853 ms<br>167.11 ms |                        | n/a  | n/a          | II/a II |   | cuStreamsvn/      | chi 25.878% | 1.521 s            |                       |                  |
| Memc<br>Memc         | 1                       | 3.072 μs               | n/a<br>n/a                                   |              | n/a     |   |                   |             |                    | Source                | Pinned           |
| Memc<br>Memc<br>Memc | 167.11 ms               | 3.072 μs<br>2.849 μs   | n/a  | n/a          | n/a     |   |                   | 21.627%     | 1.271 s<br>1.271 s | Source<br>Destination | Pinned<br>Device |



## HPC open source tools



- HPC Toolkit
  - A few years old, did not find it as straightforward to use
- TAU (Tuning and Analysis Utilities)
  - Lots of features, which makes the learning curve slow
- Scalasca
  - Developed by European consortium, did not try yet



## Intel tools

for High-Performance Computing

B-281

- Intel Parallel Studio XE 2016 Cluster Edition
  - Compilers (C/C++, Fortran)
  - Math library (MKL)
  - Threading library (TBB)
  - Thread design and prototype (Advisor)
  - Memory and thread debugging (Inspector)
  - Profiler (VTune Amplifier)
  - MPI library (Intel MPI)
  - MPI analyzer and profiler (ITAC)

nance



## Intel VTune Amplifier

- Serial and parallel profiler
  - Multicore support for OpenMP and OpenCL on CPUs, GPUs and Xeon Phi
- Quick identification of performance bottlenecks
  - Various analyses and points of view in the GUI
  - Makes choice of analysis and results inspection easier
- GUI and command line use
- More info

https://software.intel.com/en-us/intel-vtune-amplifier-xe

nance

## UNIVERSITY OF UTAH<sup>™</sup>

## Intel VTune Amplifier

Center for High-Perform Computing

- Source the environment module load vtune
- Run VTune amplxe-gui – GUI amplxe-cl – CLI
  - Can be used also for remote profiling (e.g. on Xeon Phi)
- Tuning guides for specific architectures

| Basic Hotspots Hotspots b   |  |                    |                           | Intel VTune Amplifier XE 2016   |
|---|--|--------------------|---------------------------|---|
| Grouping: Function / Call Stack   | rget 🕺 Analysis Type 🚺 Summary                                     | Botto              | om-up 🔽 Cal               | ler/Callee 😵 Top-down Tree 🔁 Platforr 🕨<br>Data Of Interest (CPU Metrics) 🗸 🗸   |
| Function / Call Stack   | CPU Time<br>Effective Time by Utilization I Idle Poor OK Ideal Ove | Spin<br>Time       | ☆ ≪ ^<br>Overhead<br>Time | ★ Viewing 4 1 of 56 ▷ selected stack(s)     26.6% (2.038s of 7.650s)     SystemProceduralFireion - fireobject.cpp   |
| FireObject::checkCollision     Junc@0x1000e190  | 7.650s   | 0s<br>2.020s       | Os<br>Os                  | SystemProceduralFire fireobject.cpp:1459  |
| <ul> <li>□ FireObject::ProcessFireCollisionsRange</li> <li>              FireObject::FireCollisionCallback ←      </li> <li>             FireObject::EmitterCollisionCheck ←         </li> <li>             func@0x7545a064         </li> </ul>   | 5.013s<br>4.025s<br>0.988s<br>3.811s                               | 0s<br>0s<br>0.675s | 0s<br>0s<br>0s            | SystemProceduralFire fireobject.cpp:1377<br>Smoke.exe!ParallelFmanagertbb.cpp:573<br>Smoke.exe![TBB parall parallel_for.h:212<br>Smoke.exe!tbb::intern parallel_for.h:150   |
| Selected 1 row(s):  | 7.6  |                    | 0s ¥                      | Smoke.exelTaskMananagertbb.cpp:606  |
| Qto Q+ Q-Q+       31         Frame Rate       Image: Comparison of the comparison |  |                    |                           | ✓       Thread       ✓         ✓       Thread       ✓         ✓       Running       ✓         ✓       Will CPU Time       ✓         ✓       ✓       CPU Sample         ✓       ✓       Tasks         ✓       ✓       CPU Usage         ✓       ✓       ✓         ✓       ✓       CPU Time         ✓       ✓       ✓         ✓       ✓       ✓         ✓       ✓       ✓         ✓       ✓       ✓         ✓       ✓       ✓         ✓       ✓       ✓         ✓       ✓       ✓ |
| 🖈 🛛 No filters are applied. 🛛 💝   | Any Process 🗸 Any Thread   |                    | ✓ Any Module              | opin and overhead mine  |
| Call Stack Mode: User functions + 1   | 🔽 Inline Mode: 🛛 🔽 Lo  | op Mode:           | Functions only            | <b>v</b>  |

https://software.intel.com/en-us/articles/processor-specificperformance-analysis-papers







- Vectorization advisor
  - Identify loops that benefit from vectorization, what is blocking efficient vectorization and explore benefit of data reorganization
- Thread design and prototyping
  - Analyze, design, tune and check threading design without disrupting normal development
- More info

http://software.intel.com/en-us/intel-advisor-xe/

#### TOGETHER WE REACH



Intel Advisor



- Source the environment module load advisorxe
- Run Advisor advixe-gui – GUI advixe-cl – CLI
- Create project and choose appropriate modeling
- Getting started guide

https://software.intel.com/en-us/get-started-with-advisor

| 🖉 Where should I add vectorization and/or threading parallelism? 📁   |      |                            |            |          |             |                 |  |          | Intel Advisor XE 2016 |   |  |
|--|------|----------------------------|------------|----------|-------------|-----------------|--|----------|-----------------------|---|--|
| 🤗 Summary 🛭 😂 Survey Report 🔅 🍅  | Refi | nement Reports 🛛 💧 Annotat | ion Report | : 📲 Suit | tability Re | port            |  |          |                       |   |  |
| Elapsed time: 54.44s Vectorized  | Not  | Vectorized 5 FILTE         | R: All Mo  | dules    | ¥ A         | II Sources      | ~  |          |                       | ୍ |  |
| Function Call Sites and Leans  | ۵    | P Vector Issues            | Self       | Total    | Trip 🔊      | <b>T</b>        | Martha Martha Martha a 2                       | Vectoriz | ed Loops              | ^ |  |
| Function Call Sites and Loops  |      | w vector issues            | Time▼      | Time     | Counts      | Loop Type       | op Type Why No Vectorization? Vecto Efficiency |          |                       |   |  |
| i> 🖱 [loop at stl_algo.h:4740 in std::tr   |      |                            | 0.170s I   | 0.170s I |             | Scalar          | non-vectorizable loop ins                      |          |                       |   |  |
| 🖃 🛄 [loop at loopstl.cpp:2449 in s234_]  |      | 💡 2 Ineffective peeled/rem | 0.170s l   | 0.170s I | 12; 4       | <u>Collapse</u> | <u>Collapse</u>                                | AVX      | ~100 <mark>%</mark>   |   |  |
| 🗈 🐸 [loop at loopstl.cpp:2449 in s   |      |                            | 0.150s l   | 0.150s I | 12          | Vectorized (B   |  | AVX      |                       |   |  |
| ₅> 🝊 [loop at loopstl.cpp:2449 in s  |      |                            | 0.020s1    | 0.020s I | 4           | Remainder       |  |          |                       |   |  |
| i> 🝊 [loop at loopstl.cpp:7900 in vas_]  |      |                            | 0.170s l   | 0.170s I | 500         | Scalar          | vectorization possible but                     |          |                       |   |  |
| 🗄 ⊍ [loop at loopstl.cpp:3509 in s2  |      |                            | 0.160s l   | 0.160s   | 12          | Expand          | Expand   | AVX      | ~6 <mark>9%</mark>    |   |  |
| 🗄  [loop at loopstl.cpp:3891 in s279_]   |      |                            | 0.150s l   | 0.150s I | 125; 4      | Expand          | Expand   | AVX      | ~9 <mark>6%</mark>    |   |  |
| 🗄 🐸 [loop at loopstl.cpp:6249 in s414_]  |      |                            | 0.150s l   | 0.150s I | 12          | Expand          | Expand   | AVX      | ~100 <mark>%</mark>   |   |  |
| المعن المعن المعن المعن المعنى ال |      | Assumed dependency         | 0.150s l   | 0.150s I | 49          | Scalar          | vector dependence preve                        |          |                       |   |  |
| <  | _    |                            |            |          | 1           |                 |  |          |                       |   |  |



## Intel Trace Analyzer and Collector

#### TOGETHER WE REACH



- MPI profiler
  - traces MPI code
  - identifies communication inefficiencies
- Collector collects the data and Analyzer visualizes them
- More info

https://software.intel.com/en-us/intel-trace-analyzer

### MPI Performance Snapshot Summary

ngeligelove). Refere 4

Oversiani Oversiani

Mits and
 Computation
 Copolition
 Copolition
 Response

erspey Ri

Property Rentworks

| hubda.  |                      |   |                   |  |  |  |  |  |
|---|----------------------|---|-------------------|--|--|--|--|--|
| v pos. M., NSOwi  |                      |   |                   |  |  |  |  |  |
|   |                      | Performance by Metric   |                   |  |  |  |  |  |
| LTO-GHC<br>BIOCOL TELERI GHC  | 11.00%<br>01.00%     | I IdealCook Barel 3000 in<br>Total Ages another The tream inspectative to the Novel groups. The hereit<br>is the fast of the KN Terre and the Computative Terrebras.  |                   |  |  |  |  |  |
| NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERA<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL<br>NUMERAL | ASPA<br>ASPA<br>1504 | MMI Three 18.28 GeV     The operational the Ministry MgR values de visuely thet     The second with the operation of Continue State Root, 1928  | in inte<br>ceccu. |  |  |  |  |  |
|   |                      | • Per endersing 112.00 eV.<br>Massa approximation wait the parameter spectra that there<br>exists a pro-basic existing for data. This tensor part of the end of<br>regionalized the consideration.<br>The spectra effect that approximate and the end of the<br>parameter of the construction.  |                   |  |  |  |  |  |
|   |                      | Computation Term 11.20-per     Provide the period of  | LEADER<br>LADEL   |  |  |  |  |  |
|   |                      | R Equative Variation of A Review.<br>Physicillary per primary speed in the Careford' particles repairs, in<br>which seture is a set of our constraints in a set of our<br>Web seture is a seture of the seture of the seture of the<br>Web seture is a seture of the seture of the seture of the<br>Web seture is a seture of the seture of the seture of the<br>Web seture is a seture of the seture of the seture of the<br>Web seture of the seture of the seture of the seture of the<br>Web seture of the seture of the seture of the seture of the<br>Web seture of the seture of the seture of the seture of the seture of the<br>Web seture of the seture of the seture of the seture of the seture of the<br>Web seture of the seture of the seture of the seture of the seture of the<br>Web seture of the seture of the seture of the seture of the seture of the<br>Web seture of the seture of the seture of the seture of the seture of the<br>seture of the seture of the seture of the seture of the seture of the<br>seture of the seture of the<br>seture of the seture of the<br>seture of the seture of the<br>seture of the seture of th |                   |  |  |  |  |  |
|   | 0.9 Mil<br>9.5 Mil   | B Constitution on the 20 million of the part of the   | •                 |  |  |  |  |  |
|   |                      | <ul> <li>a Senal Theor 6.40 per<br/>Merch specification per services specification (specific per<br/>High space cars to great in ball depending on the application<br/>Microsoft (100,1000). This approximation is interaction<br/>attention.</li> </ul>  | aperies.          |  |  |  |  |  |
|   |                      |   |                   |  |  |  |  |  |

### UNIVERSITY OF UTAH<sup>™</sup>

## Intel TAC

- Source the environment module load itac
- Using Intel compilers, can compile with -trace

mpiifort -openmp -trace trap.f

• Run MPI code

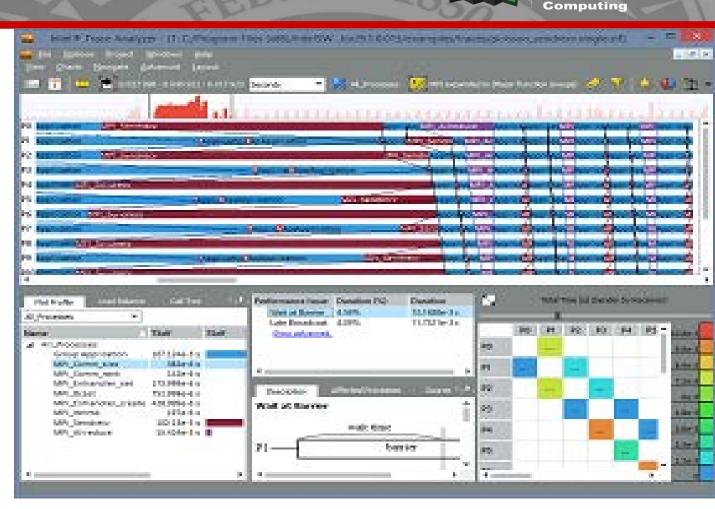
mpirun -trace -n 4 ./a.out

• Run visualizer

traceanalyzer a.out.stf &

• Getting started guide

https://software.intel.com/en-us/get-started-with-itac-for-linux



B-281

### 8/1/2016

### TOGETHER WE REACH

Center

for High-

ŏrmance





# **RUNTIME MONITORING**

EEB. 28TH

8/2/2016

http://www.chpc.utah.edu

Slide 40



## Why runtime monitoring?



- Make sure program is running right
  - Hardware problems
  - Correct parallel mapping / process affinity
- Careful about overhead



# Runtime monitoring



- Self checking
  - ssh to node(s), run "top", or look at "sar" logs
  - SLURM (or other scheduler) logs and statistics
- Tools
  - XDMoD XSEDE Metrics on Demand (through SUPReMM module)
  - REMORA REsource MOnitoring for Remote Applications





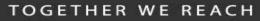
# BENCHMARKING

EEB. 28TH

8/2/2016



## Why to benchmark?





- Evaluate system's performance
  - Testing new hardware
- Verify correct hardware and software installation
  - New cluster/node deployment
    - There are tools for cluster checking (Intel Cluster Checker, cluster distros, ...)
  - Checking newly built programs
    - Sometimes we leave this to the users



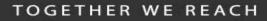
New system evaluation



- Simple synthetic benchmarks
   FLOPS, STREAM
- Synthetic benchmarks
  - HPL High Performance Linpack dense linear algebra problems – cache friendly
  - HPCC HPC Challenge Benchmark collection of dense, sparse and other (FFT) benchmarks
  - NPB NAS Parallel Benchmarks mesh based solvers OpenMP, MPI, OpenACC implementations

### THE UNIVERSITY OF UTAH<sup>™</sup>

## New system evaluation





- Real applications benchmarks
  - Depend on local usage
  - Gaussian, VASP
  - Amber, LAMMPS, NAMD, Gromacs
  - ANSYS, Abaqus, StarCCM+
  - Own codes
- Script if possible
  - A lot of combinations of test cases vs. number of MPI tasks/OpenMP cores



## Cluster deployment



- Whole cluster
  - Some vendors have cluster verification tools
  - We have a set of scripts that run basic checks and HPL at the end
- New cluster nodes
  - Verify received hardware configuration, then rack
  - Basic system tests (node health check)
  - HPL get expected performance per node (CPU or memory issues), or across more nodes (network issues)





# BACKUP

FEB. 28TH

8/2/2016

#### TOGETHER WE REACH

FEB. 28TH 7



• Totalview

Demos

• Advisor

THE UNIVERSITY OF UTAH<sup>™</sup>

- Inspector
- VTune