JCAHPC

Oakforest-PACS Tuning Guide

Joint Center for Advanced High Performance Computing (JCAHPC)

For Intel Parallel Studio 2017 update 1

January 27, 2017

東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO

筑波大学
計算科学研究センター
Center for Computational Sciences

Compile

Mandatory

•-xMIC-AVX512 –qopenmp

(If Flat MPI, faster without –qopenmp)

Advisable

- -align array64byte
  - Align to cache line size

Advisable to try

- -qopt-streaming-stores= always / never / auto
  - Default: auto
  - Assign “always” in Stream bench
  - Control possible by each string
  - #pragma vector nontemporal (in C/C++)
  - !DEC$ vector nontemporal (in F)

Options during execution (Common)

Auxiliary commands

- numactl
  - For “affinity" setting
  - Also used for MCDRAM specification
  - numactl -s $$: Core used
  - numactl -H : HW data
- taskset
  - Convenient for limiting cores used
  - taskset -c 2-64,67-127,130-191,194-255 ./a.out: Displays cores used

Environment variables

Settings under export commands (omitted below)

- I_MPI_XXX
  - Option unique to Intel MPI
  - Prioritized above KMP_XXX
- KMP_XXX
  - Option unique to Intel Compiler
- OMP_XXX
  - OpenMP runtime option (general-purpose)

Options during execution (affinity 1/2)

Mandatory during debugging

Check if allocation is carried out as anticipated

- I_MPI_DEBUG=5
  - Status of allocation of compute nodes and cores to each rank
- KMP_AFFINITY=verbose
  - Status of allocation of cores within each process
- TMI_DEBUG=1
  - Debugging data when protocol is specified as “tmi"

Tickless setting

To avoid OS jitter, only Core 0 (and its HT core) are set to accept timer interruption

- Avoid the use of cores 0, 1 (tile 1 unit) of KNL
- I_MPI_PIN_PROCESSOR_EXCL UDE_LIST=0,1,68,69,136,137, 204,205
- When MPI is not used: KMP_HW_SUBSET=64c@2,1t
  - Secure 64 cores without using the first two cores

Options during execution (affinity 2/2)

Allocate one to each physical core (Assumption that HyperThreading core is not used)

- KMP_HW_SUBSET=1T
- unset KMP_AFFINITY (As “compact” has already been specified, performance will deteriorate if “unset” is forgotten)

Allocate two (or more) to each physical core (Use HyperThreading core)

- KMP_HW_SUBSET=2T
  - Up to 4T possible
- It is better to specify in combination with KMP_AFFINITY=compact (already done)

Options during execution (Multiple MPI processes per node) (1/2)

<u>When there are two or more PPN (Process Per Node)</u>

- To allocate multiple (N) per node, add “-ppn N” to the argument
  - I_MPI_PERHOST=N

  or
  - mpiexec.hydra, mpirun

- How to allocate by the process unit
  - I_MPI_PIN_DOMAIN=PN
  - For example, if 256 cores were used overall, PN=256/N
  - Refer to the next page for an example

Options during execution (Multiple MPI processes per node) (2/2)

Example
PPN=4 (4MPI 16 threads)

- OMP_NUM_THREADS=16
- I_MPI_PIN_DOMAIN=64
- I_MPI_PERHOST=4

When PPN=8 (8MPI 8 threads)

- I_MPI_PIN_DOMAIN=32
- OMP_NUM_THREADS=8
- I_MPI_PERHOST=8

Options during execution (related to MCDRAM)

- numactl --membind=1
  ./a.out
  - When it does not fit into MCDRAM, set
    numactl --preferred=1
    ./a.out
    And use MCDRAM through best efforts
  - Set numactl --interleave=0,1
    ./a.out
    Use MCDRAM and DDR4 with round-robin

MPI buffer, etc. on MCDRAM

(Take note as latency is somewhat greater than DDR4)

- I_MPI_HBW_POLICY=<A>, <B>, <C>
  - Options: hbw_bind / hbw_preferred / hbw_interleave / Do not write
  - <A>: User process
    - With this, the numactl on the left is not necessary
  - <B>: MPI buffer
  - <C>: Win allocate (MPI-3)

Options during execution (related to communications)

Select protocol stack

- I_MPI_FABRICS_LIST=tmi
  - tmi: OmniPath recommendation
  - ofi: Future industry standard, may be better than tmi sometimes
  - ofa: Slow, used for debugging

  （IB compatibility mode）
- I_MPI_FABRICS=shm:tmi
  - shm: Shared memory (within nodes)
  - Sometimes may be better to use tmi instead of shm
    - In short, I_MPI_FABRICS=tmi:tmi

Advisable to try

- HFI_NO_CPUAFFINITY=1
  - There are cases where there are differences and cases where there aren’t?
    - It may be better to specify when executing multiple processes within the nodes
  - Impact when tmi is selected? Unverified

Check: numactl –H of each execution mode

```
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87
88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122
123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184
185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246
247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271
node 0 size: 98147 MB
node 0 free: 85936 MB
node 1 cpus:
node 1 size: 16384 MB
node 1 free: 15799 MB
node distances:
node   0   1
  0:  10  31
  1:  31  10
```

FLAT QUADRANT

- CPU: node 0 only
- DDR4: node 0
- MCDRAM: node 1

available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86
87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132
133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174
175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195
196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237
238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258
259 260 261 262 263 264 265 266 267 268 269 270 271
node 0 size: 98147 MB
node 0 free: 85492 MB
node distances:
node 0
0: 10

CACHE QUADRANT

- One memory only for CPU as well