QuantLib on Intel Xeon Phi

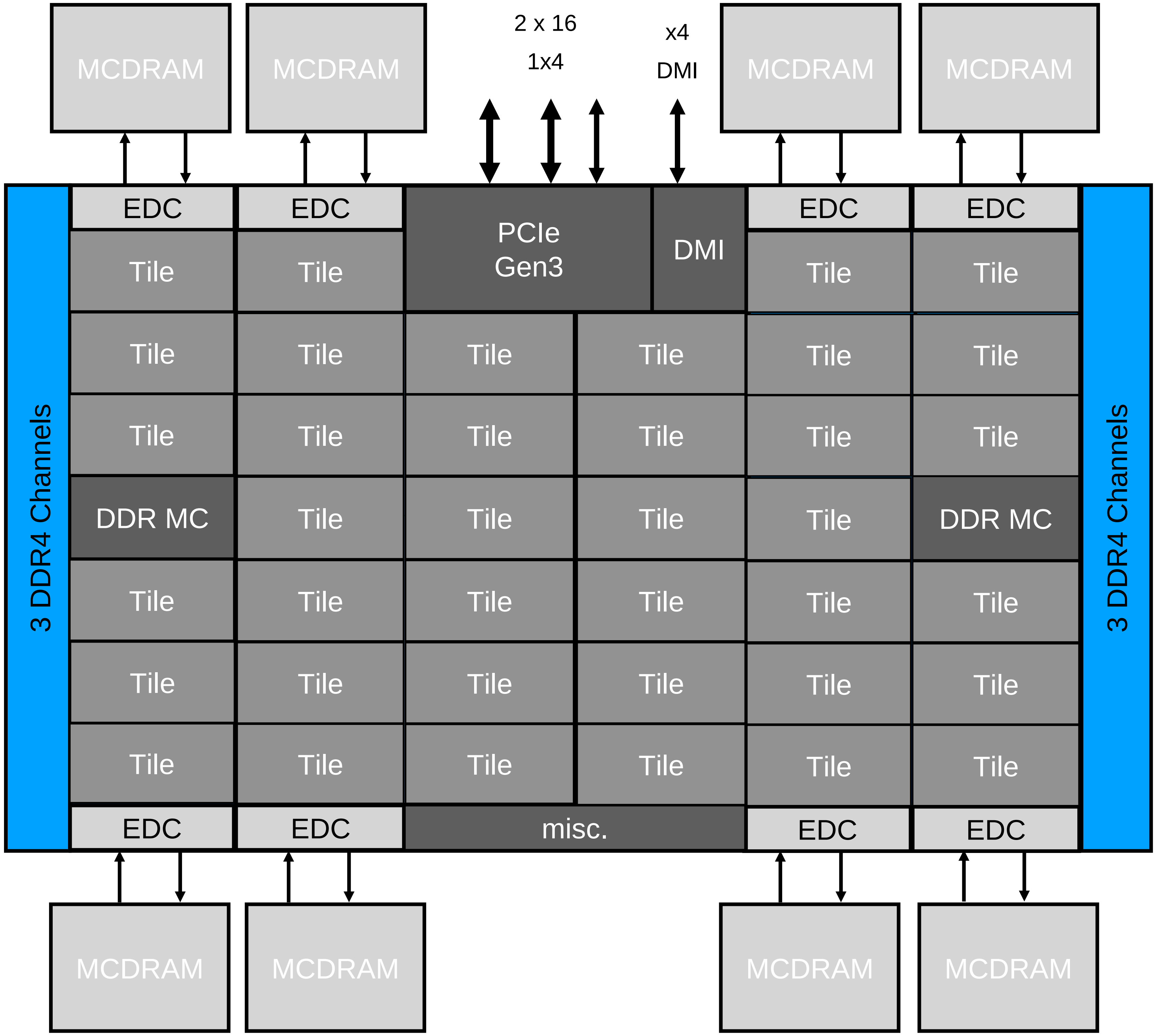
Thomas Rodgers
FCA Design
rodgert@twrodgers.com

My background

- Disclaimer - I am not a quant
- C++ developer with > 20 years of experience in Finance
- Member of the C++ Standards Committee
  - SG1 - Concurrency and Parallelism
  - SG14 - Games, HFT, Low-Latency systems
- My interest in this project is how to make C++ a better tool for heterogenous multi-core development

Project Background

- Work in support of research by Prof. Matthew Dixon, Stuart School of Business, Illinois Institute of Technology
    - Help Quants target highly parallel, or vector-parallel architectures, without needing a deep background in vector/parallel software engineering
    - Xeon Phi is one such target architecture
- Need a source of publicly releasable financial model codes
    - QuantLib is one possibility



Knights Landing

Die Layout

Knights Landing Die Features

- Chip organized around the concept of a Tile, each consists of -
  - 2 cores, 32kb L1 Instruction and 32kb L1 Data cache per core
  - 2 vector processing units per core, total 4 per Tile
  - 1M of L2 cache shared between the two cores
- All Tiles are connected to a 2D mesh with > 700GB/s bandwidth
  - Organized into rows and columns of “half” rings that fold upon themselves at the endpoints
  - Enforces a “YX routing” rule, transactions travel vertically to target row, then travel horizontally to destination.

Knights Landing Die Features (cont)

- 2 Memory controllers with 3 DDR4 channels each
  - Maximum of 384GB of DDR4
  - Aggregate DDR4 bandwidth is ~90 GB/s
- 8 MCDRAM (High Bandwidth Memory) devices, each 2GB capacity
  - Each MCDRAM is connected to it’s own memory controller (EDC)
  - Aggregate MCDRAM bandwidth is > 450 GB/s
  - Can be used in cache mode, flat mode (shares address space with DDR), or a mix of the two

Knights Landing
Core details

- Each Knights Landing die contains 36 tiles of 2 cores each for a total of 72 cores per die
- Each KNL core is based on a significantly modified Atom processor (Silvermont)
- Features -
  - ISA compatible with Haswell (except TSX support)
  - Support for up to 4 simultaneous threads (hyperthreading)
  - Out of order execution

Knights Landing Core details (cont)

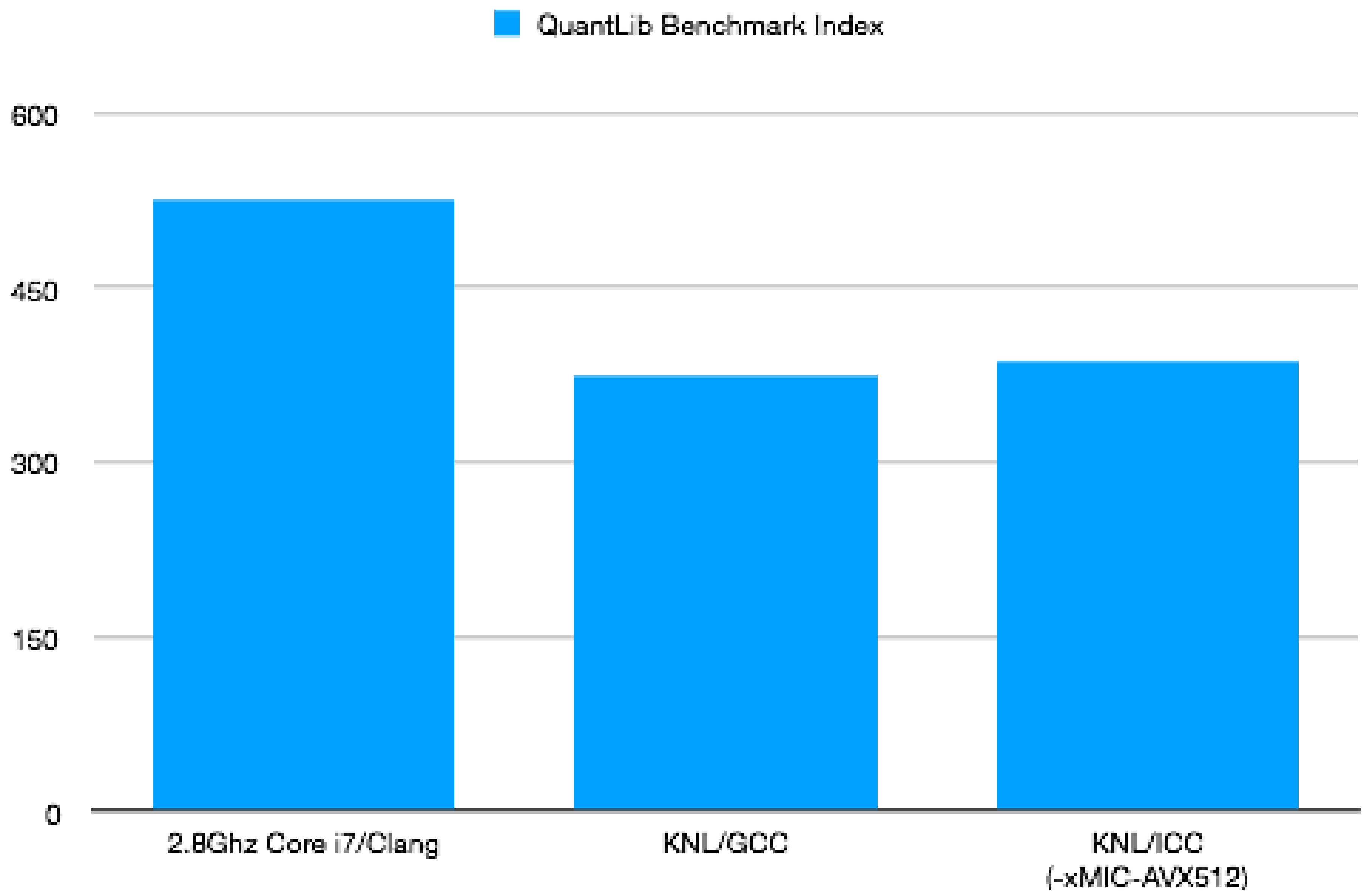
- Each Knights Landing tile includes two Vector Processing Units per core for a total of 144 VPUs per KNL die
- Each VPU can perform 16 double precision, or 32 single precision floating point operations
- Each VPU supports an extended 512 bit instruction set, aka AVX-512
  - Includes “foundation” support for floating point and scalar vector operations, as well as dedicated support for exponentiation, prefetch, and conflict detection
  - AVX-512 ISA support is available via compiler intrinsics in ICC, GCC, and Clang.
  - Support for AVX-512 compiler intrinsics are also available in ICC, GCC, and Clang
    - see https://software.intel.com/sites/landingpage/IntrinsicsGuide for more information

Knights Landing Configurations

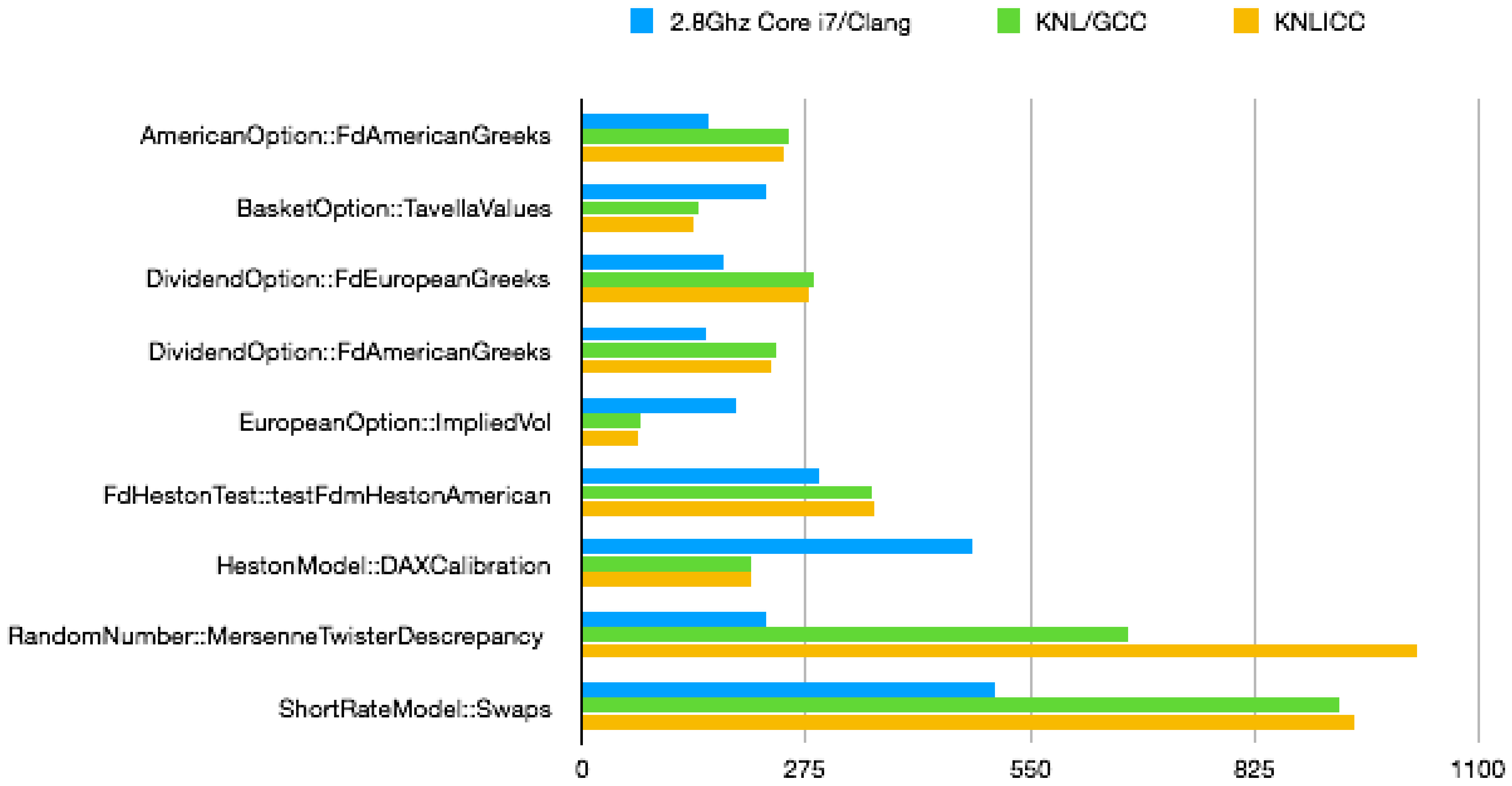
- The Knights Landing processor, differs from the previous generation Knights Corner processor -
  - Standard Intel Architecture device
  - Fully capable of booting stock operating systems
    - Currently using CentOS 7.4

Knights Landing
Boot-time Configuration

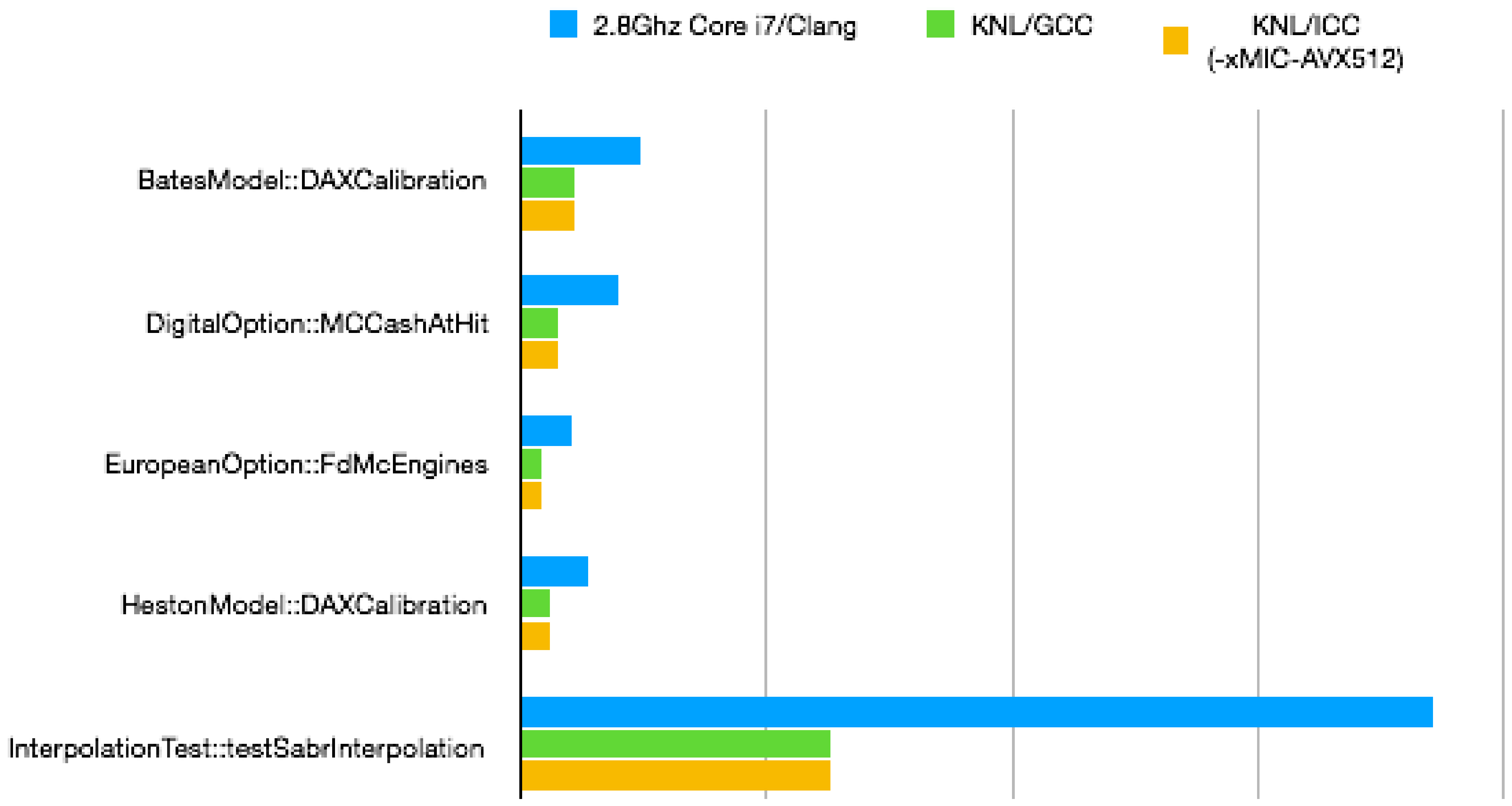
- Unlike standard Intel Xeon CPUs the Knights Landing CPU supports several boot-time configuration parameters
  - Cluster Mode - Configures the on chip mesh in one of
    - all-to-all mode (used with mixed DRAM configurations)
    - quadrant mode (default)
    - Sub-NUMA clustering (SNC)
  - Memory Mode -
    - cache, flat, hybrid



Benchmark Index (MIPS)



Benchmark Sample

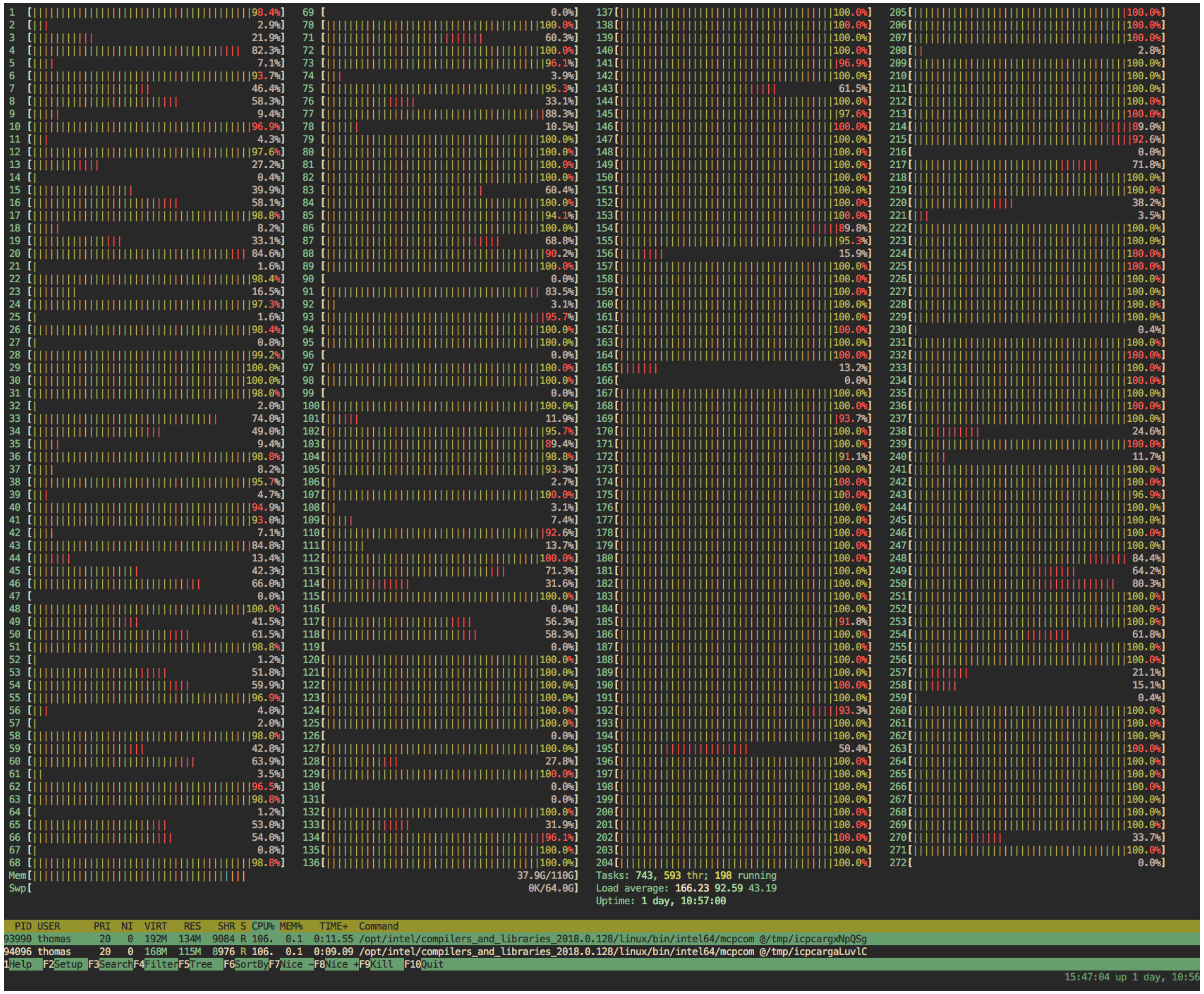


Benchmark Sample

Observations

- The benchmark suite is single threaded
- Aggregate single core performance for Phi vs. a typical Intel Core/Xeon is lower, this result is generally expected, but…
  - Some individual models >2x slower
- This appears at least in part due to how the QuantLib benchmark suite is structured
  - Significant amount of scalar set-up code inside deeply nested loops

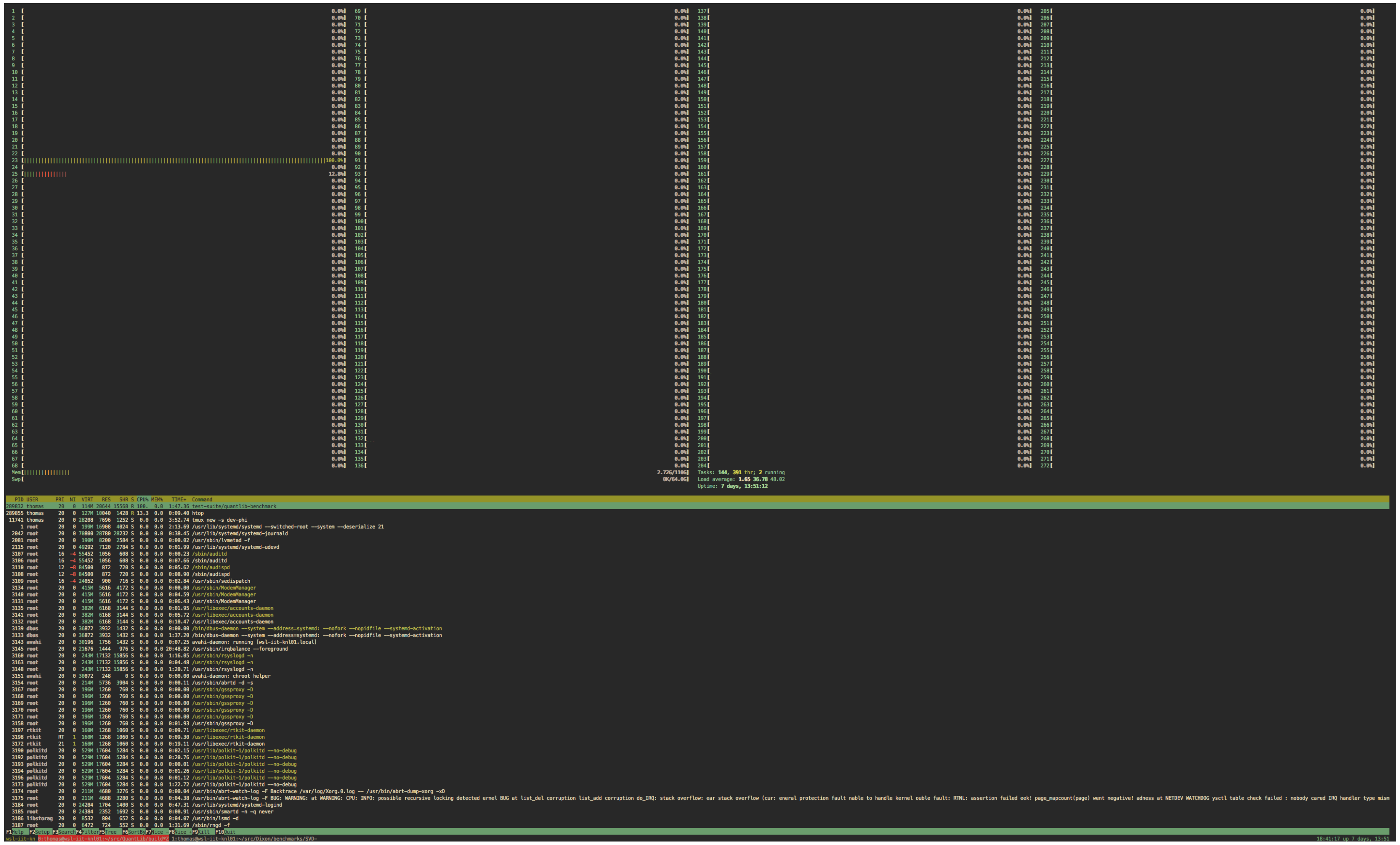
It’s Full of Cores…



```
1  [ 98.4%]   69 [  0.0%]   137[100.0%]   205[100.0%]
2  [  2.9%]   70 [100.0%]   138[100.0%]   206[100.0%]
3  [ 21.9%]   71 [ 60.3%]   139[100.0%]   207[100.0%]
4  [ 82.3%]   72 [100.0%]   140[100.0%]   208[  2.8%]
5  [  7.1%]   73 [ 96.1%]   141[ 96.9%]   209[100.0%]
6  [ 93.7%]   74 [  3.9%]   142[100.0%]   210[100.0%]
7  [ 46.4%]   75 [ 95.3%]   143[ 61.5%]   211[100.0%]
8  [ 58.3%]   76 [ 33.1%]   144[100.0%]   212[100.0%]
9  [  9.4%]   77 [ 88.3%]   145[ 97.6%]   213[100.0%]
10 [ 96.9%]   78 [ 10.5%]   146[100.0%]   214[ 89.0%]
11 [  4.3%]   79 [100.0%]   147[100.0%]   215[ 92.6%]
12 [ 97.6%]   80 [100.0%]   148[100.0%]   216[  0.0%]
13 [ 27.2%]   81 [100.0%]   149[100.0%]   217[ 71.8%]
14 [  0.4%]   82 [100.0%]   150[100.0%]   218[100.0%]
15 [ 39.9%]   83 [ 60.4%]   151[100.0%]   219[100.0%]
16 [ 58.1%]   84 [100.0%]   152[100.0%]   220[ 38.2%]
17 [ 98.8%]   85 [ 94.1%]   153[100.0%]   221[  3.5%]
18 [  8.2%]   86 [100.0%]   154[ 89.8%]   222[100.0%]
19 [ 33.1%]   87 [ 68.8%]   155[ 95.3%]   223[100.0%]
20 [ 84.6%]   88 [ 90.2%]   156[ 15.9%]   224[100.0%]
21 [  1.6%]   89 [100.0%]   157[100.0%]   225[100.0%]
22 [ 98.4%]   90 [  0.0%]   158[100.0%]   226[100.0%]
23 [ 16.5%]   91 [ 83.5%]   159[100.0%]   227[100.0%]
24 [ 97.3%]   92 [  3.1%]   160[100.0%]   228[100.0%]
25 [  1.6%]   93 [ 95.7%]   161[100.0%]   229[100.0%]
26 [ 98.4%]   94 [100.0%]   162[100.0%]   230[  0.4%]
27 [  0.8%]   95 [100.0%]   163[100.0%]   231[100.0%]
28 [ 99.2%]   96 [  0.0%]   164[100.0%]   232[100.0%]
29 [100.0%]   97 [100.0%]   165[ 13.2%]   233[100.0%]
30 [100.0%]   98 [100.0%]   166[  0.0%]   234[100.0%]
31 [ 98.0%]   99 [  0.0%]   167[100.0%]   235[100.0%]
32 [  2.0%]   100[100.0%]   168[100.0%]   236[100.0%]
33 [ 74.0%]   101[ 11.9%]   169[ 93.7%]   237[100.0%]
34 [ 49.0%]   102[ 95.7%]   170[100.0%]   238[ 24.6%]
35 [  9.4%]   103[ 89.4%]   171[100.0%]   239[100.0%]
36 [ 98.8%]   104[ 98.8%]   172[ 91.1%]   240[ 11.7%]
37 [  8.2%]   105[ 93.3%]   173[100.0%]   241[100.0%]
38 [ 95.7%]   106[  2.7%]   174[100.0%]   242[100.0%]
39 [  4.7%]   107[100.0%]   175[100.0%]   243[ 96.9%]
40 [ 94.9%]   108[  3.1%]   176[100.0%]   244[100.0%]
41 [ 93.0%]   109[  7.4%]   177[100.0%]   245[100.0%]
42 [  7.1%]   110[ 92.6%]   178[100.0%]   246[100.0%]
43 [ 84.8%]   111[ 13.7%]   179[100.0%]   247[100.0%]
44 [ 13.4%]   112[100.0%]   180[100.0%]   248[ 84.4%]
45 [ 42.3%]   113[ 71.3%]   181[100.0%]   249[ 64.2%]
46 [ 66.0%]   114[ 31.6%]   182[100.0%]   250[ 80.3%]
47 [  0.0%]   115[100.0%]   183[100.0%]   251[100.0%]
48 [100.0%]   116[  0.0%]   184[100.0%]   252[100.0%]
49 [ 41.5%]   117[ 56.3%]   185[ 91.8%]   253[100.0%]
50 [ 61.5%]   118[ 58.3%]   186[100.0%]   254[ 61.8%]
51 [ 98.8%]   119[  0.0%]   187[100.0%]   255[100.0%]
52 [  1.2%]   120[100.0%]   188[100.0%]   256[100.0%]
53 [ 51.8%]   121[100.0%]   189[100.0%]   257[ 21.1%]
54 [ 59.9%]   122[100.0%]   190[100.0%]   258[ 15.1%]
55 [ 96.9%]   123[100.0%]   191[100.0%]   259[  0.4%]
56 [  4.0%]   124[100.0%]   192[ 93.3%]   260[100.0%]
57 [  2.0%]   125[100.0%]   193[100.0%]   261[100.0%]
58 [ 98.0%]   126[  0.0%]   194[100.0%]   262[100.0%]
59 [ 42.8%]   127[100.0%]   195[ 50.4%]   263[100.0%]
60 [ 63.9%]   128[ 27.8%]   196[100.0%]   264[100.0%]
61 [  3.5%]   129[100.0%]   197[100.0%]   265[100.0%]
62 [ 96.5%]   130[  0.0%]   198[100.0%]   266[100.0%]
63 [ 98.8%]   131[  0.0%]   199[100.0%]   267[100.0%]
64 [  1.2%]   132[100.0%]   200[100.0%]   268[100.0%]
65 [ 53.0%]   133[ 31.9%]   201[100.0%]   269[100.0%]
66 [ 54.0%]   134[ 96.1%]   202[100.0%]   270[ 33.7%]
67 [  0.8%]   135[100.0%]   203[100.0%]   271[100.0%]
68 [ 98.8%]   136[100.0%]   204[100.0%]   272[  0.0%]
Mem[                  37.9G/110G]   Tasks: 743, 593 thr; 198 running
Swp[                     0K/64.0G]   Load average: 166.23 92.59 43.19
                                     Uptime: 1 day, 10:57:00

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
93990 thomas     20   0  192M  134M  9084 R 106.  0.1  0:11.55 /opt/intel/compilers_and_libraries_2018.0.128/linux/bin/intel64/mcpcom @/tmp/icpcargxNpQSg
94096 thomas     20   0  168M  115M  8976 R 106.  0.1  0:09.09 /opt/intel/compilers_and_libraries_2018.0.128/linux/bin/intel64/mcpcom @/tmp/icpcargaLuvlC
1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
                                                                                                   15:47:04 up 1 day, 10:56
```

The challenge is using them.



```
  1  [                0.0%]   69 [                0.0%]   137[                0.0%]   205[                0.0%]
  2  [                0.0%]   70 [                0.0%]   138[                0.0%]   206[                0.0%]
  3  [                0.0%]   71 [                0.0%]   139[                0.0%]   207[                0.0%]
  4  [                0.0%]   72 [                0.0%]   140[                0.0%]   208[                0.0%]
  5  [                0.0%]   73 [                0.0%]   141[                0.0%]   209[                0.0%]
  6  [                0.0%]   74 [                0.0%]   142[                0.0%]   210[                0.0%]
  7  [                0.0%]   75 [                0.0%]   143[                0.0%]   211[                0.0%]
  8  [                0.0%]   76 [                0.0%]   144[                0.0%]   212[                0.0%]
  9  [                0.0%]   77 [                0.0%]   145[                0.0%]   213[                0.0%]
  10 [                0.0%]   78 [                0.0%]   146[                0.0%]   214[                0.0%]
  11 [                0.0%]   79 [                0.0%]   147[                0.0%]   215[                0.0%]
  12 [                0.0%]   80 [                0.0%]   148[                0.0%]   216[                0.0%]
  13 [                0.0%]   81 [                0.0%]   149[                0.0%]   217[                0.0%]
  14 [                0.0%]   82 [                0.0%]   150[                0.0%]   218[                0.0%]
  15 [                0.0%]   83 [                0.0%]   151[                0.0%]   219[                0.0%]
  16 [                0.0%]   84 [                0.0%]   152[                0.0%]   220[                0.0%]
  17 [                0.0%]   85 [                0.0%]   153[                0.0%]   221[                0.0%]
  18 [                0.0%]   86 [                0.0%]   154[                0.0%]   222[                0.0%]
  19 [                0.0%]   87 [                0.0%]   155[                0.0%]   223[                0.0%]
  20 [                0.0%]   88 [                0.0%]   156[                0.0%]   224[                0.0%]
  21 [                0.0%]   89 [                0.0%]   157[                0.0%]   225[                0.0%]
  22 [                0.0%]   90 [                0.0%]   158[                0.0%]   226[                0.0%]
  23 [||||||||||||||100.0%]   91 [                0.0%]   159[                0.0%]   227[                0.0%]
  24 [                0.0%]   92 [                0.0%]   160[                0.0%]   228[                0.0%]
  25 [|||||||||||||  12.0%]   93 [                0.0%]   161[                0.0%]   229[                0.0%]
  26 [                0.0%]   94 [                0.0%]   162[                0.0%]   230[                0.0%]
  27 [                0.0%]   95 [                0.0%]   163[                0.0%]   231[                0.0%]
  28 [                0.0%]   96 [                0.0%]   164[                0.0%]   232[                0.0%]
  29 [                0.0%]   97 [                0.0%]   165[                0.0%]   233[                0.0%]
  30 [                0.0%]   98 [                0.0%]   166[                0.0%]   234[                0.0%]
  31 [                0.0%]   99 [                0.0%]   167[                0.0%]   235[                0.0%]
  32 [                0.0%]   100[                0.0%]   168[                0.0%]   236[                0.0%]
  33 [                0.0%]   101[                0.0%]   169[                0.0%]   237[                0.0%]
  34 [                0.0%]   102[                0.0%]   170[                0.0%]   238[                0.0%]
  35 [                0.0%]   103[                0.0%]   171[                0.0%]   239[                0.0%]
  36 [                0.0%]   104[                0.0%]   172[                0.0%]   240[                0.0%]
  37 [                0.0%]   105[                0.0%]   173[                0.0%]   241[                0.0%]
  38 [                0.0%]   106[                0.0%]   174[                0.0%]   242[                0.0%]
  39 [                0.0%]   107[                0.0%]   175[                0.0%]   243[                0.0%]
  40 [                0.0%]   108[                0.0%]   176[                0.0%]   244[                0.0%]
  41 [                0.0%]   109[                0.0%]   177[                0.0%]   245[                0.0%]
  42 [                0.0%]   110[                0.0%]   178[                0.0%]   246[                0.0%]
  43 [                0.0%]   111[                0.0%]   179[                0.0%]   247[                0.0%]
  44 [                0.0%]   112[                0.0%]   180[                0.0%]   248[                0.0%]
  45 [                0.0%]   113[                0.0%]   181[                0.0%]   249[                0.0%]
  46 [                0.0%]   114[                0.0%]   182[                0.0%]   250[                0.0%]
  47 [                0.0%]   115[                0.0%]   183[                0.0%]   251[                0.0%]
  48 [                0.0%]   116[                0.0%]   184[                0.0%]   252[                0.0%]
  49 [                0.0%]   117[                0.0%]   185[                0.0%]   253[                0.0%]
  50 [                0.0%]   118[                0.0%]   186[                0.0%]   254[                0.0%]
  51 [                0.0%]   119[                0.0%]   187[                0.0%]   255[                0.0%]
  52 [                0.0%]   120[                0.0%]   188[                0.0%]   256[                0.0%]
  53 [                0.0%]   121[                0.0%]   189[                0.0%]   257[                0.0%]
  54 [                0.0%]   122[                0.0%]   190[                0.0%]   258[                0.0%]
  55 [                0.0%]   123[                0.0%]   191[                0.0%]   259[                0.0%]
  56 [                0.0%]   124[                0.0%]   192[                0.0%]   260[                0.0%]
  57 [                0.0%]   125[                0.0%]   193[                0.0%]   261[                0.0%]
  58 [                0.0%]   126[                0.0%]   194[                0.0%]   262[                0.0%]
  59 [                0.0%]   127[                0.0%]   195[                0.0%]   263[                0.0%]
  60 [                0.0%]   128[                0.0%]   196[                0.0%]   264[                0.0%]
  61 [                0.0%]   129[                0.0%]   197[                0.0%]   265[                0.0%]
  62 [                0.0%]   130[                0.0%]   198[                0.0%]   266[                0.0%]
  63 [                0.0%]   131[                0.0%]   199[                0.0%]   267[                0.0%]
  64 [                0.0%]   132[                0.0%]   200[                0.0%]   268[                0.0%]
  65 [                0.0%]   133[                0.0%]   201[                0.0%]   269[                0.0%]
  66 [                0.0%]   134[                0.0%]   202[                0.0%]   270[                0.0%]
  67 [                0.0%]   135[                0.0%]   203[                0.0%]   271[                0.0%]
  68 [                0.0%]   136[                0.0%]   204[                0.0%]   272[                0.0%]
  Mem[||||||||||||||||                                        2.72G/110G]   Tasks: 144, 391 thr; 2 running
  Swp[                                                           0K/64.0G]   Load average: 1.65 36.78 48.02
                                                                             Uptime: 7 days, 13:51:12

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 289832 thomas     20   0  114M 20644 15568 R 100.  0.0  1:47.36 test-suite/quantlib-benchmark
 289855 thomas     20   0  127M 10040  1428 R 13.3  0.0  0:09.40 htop
  11741 thomas     20   0 28208  7696  1252 S  0.0  0.0  3:52.74 tmux new -s dev-phi
      1 root       20   0  199M 16908  4024 S  0.0  0.0  2:13.69 /usr/lib/systemd/systemd --switched-root --system --deserialize 21
   2042 root       20   0 70800 28780 28232 S  0.0  0.0  0:38.45 /usr/lib/systemd/systemd-journald
   2081 root       20   0  190M  8200  2584 S  0.0  0.0  0:00.02 /usr/sbin/lvmetad -f
   2115 root       20   0 49292  7120  2784 S  0.0  0.0  0:01.99 /usr/lib/systemd/systemd-udevd
   3107 root       16  -4 55452  1056   608 S  0.0  0.0  0:00.23 /sbin/auditd
   3106 root       16  -4 55452  1056   608 S  0.0  0.0  0:07.66 /sbin/auditd
   3110 root       12  -8 84500   872   720 S  0.0  0.0  0:05.62 /sbin/audispd
   3108 root       12  -8 84500   872   720 S  0.0  0.0  0:08.90 /sbin/audispd
   3109 root       16  -4 24052   900   716 S  0.0  0.0  0:02.84 /usr/sbin/sedispatch
   3134 root       20   0  415M  5616  4172 S  0.0  0.0  0:00.00 /usr/sbin/ModemManager
   3140 root       20   0  415M  5616  4172 S  0.0  0.0  0:04.59 /usr/sbin/ModemManager
   3131 root       20   0  415M  5616  4172 S  0.0  0.0  0:06.43 /usr/sbin/ModemManager
   3135 root       20   0  382M  6168  3144 S  0.0  0.0  0:01.95 /usr/libexec/accounts-daemon
   3141 root       20   0  382M  6168  3144 S  0.0  0.0  0:05.72 /usr/libexec/accounts-daemon
   3132 root       20   0  382M  6168  3144 S  0.0  0.0  0:10.47 /usr/libexec/accounts-daemon
   3139 dbus       20   0 36872  3932  1432 S  0.0  0.0  0:00.00 /bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation
   3133 dbus       20   0 36872  3932  1432 S  0.0  0.0  1:37.20 /bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation
   3143 avahi      20   0 30196  1756  1432 S  0.0  0.0  0:07.25 avahi-daemon: running [wsl-iit-knl01.local]
   3145 root       20   0 21676  1444   976 S  0.0  0.0 20:48.82 /usr/sbin/irqbalance --foreground
   3160 root       20   0  243M 17132 15856 S  0.0  0.0  1:16.05 /usr/sbin/rsyslogd -n
   3163 root       20   0  243M 17132 15856 S  0.0  0.0  0:04.48 /usr/sbin/rsyslogd -n
   3148 root       20   0  243M 17132 15856 S  0.0  0.0  1:20.71 /usr/sbin/rsyslogd -n
   3151 avahi      20   0 30072   248     0 S  0.0  0.0  0:00.00 avahi-daemon: chroot helper
   3154 root       20   0  214M  5736  3904 S  0.0  0.0  0:00.11 /usr/sbin/abrtd -d -s
   3167 root       20   0  196M  1260   760 S  0.0  0.0  0:00.00 /usr/sbin/gssproxy -D
   3168 root       20   0  196M  1260   760 S  0.0  0.0  0:00.00 /usr/sbin/gssproxy -D
   3169 root       20   0  196M  1260   760 S  0.0  0.0  0:00.00 /usr/sbin/gssproxy -D
   3170 root       20   0  196M  1260   760 S  0.0  0.0  0:00.00 /usr/sbin/gssproxy -D
   3171 root       20   0  196M  1260   760 S  0.0  0.0  0:00.00 /usr/sbin/gssproxy -D
   3158 root       20   0  196M  1260   760 S  0.0  0.0  0:01.93 /usr/sbin/gssproxy -D
   3197 rtkit      20   0  160M  1268  1060 S  0.0  0.0  0:09.71 /usr/libexec/rtkit-daemon
   3198 rtkit      RT   1  160M  1268  1060 S  0.0  0.0  0:09.30 /usr/libexec/rtkit-daemon
   3172 rtkit      21   1  160M  1268  1060 S  0.0  0.0  0:19.11 /usr/libexec/rtkit-daemon
   3190 polkitd    20   0  529M 17604  5284 S  0.0  0.0  0:02.15 /usr/lib/polkit-1/polkitd --no-debug
   3192 polkitd    20   0  529M 17604  5284 S  0.0  0.0  0:20.76 /usr/lib/polkit-1/polkitd --no-debug
   3193 polkitd    20   0  529M 17604  5284 S  0.0  0.0  0:00.01 /usr/lib/polkit-1/polkitd --no-debug
   3194 polkitd    20   0  529M 17604  5284 S  0.0  0.0  0:01.26 /usr/lib/polkit-1/polkitd --no-debug
   3196 polkitd    20   0  529M 17604  5284 S  0.0  0.0  0:01.12 /usr/lib/polkit-1/polkitd --no-debug
   3173 polkitd    20   0  529M 17604  5284 S  0.0  0.0  1:22.72 /usr/lib/polkit-1/polkitd --no-debug
   3174 root       20   0  211M  4680  3276 S  0.0  0.0  0:00.04 /usr/bin/abrt-watch-log -F Backtrace /var/log/Xorg.0.log -- /usr/bin/abrt-dump-xorg -xD
   3175 root       20   0  211M  4688  3280 S  0.0  0.0  0:04.38 /usr/bin/abrt-watch-log -F BUG: WARNING: at WARNING: CPU: INFO: possible recursive locking detected ernel BUG at list_del corruption list_add corruption do_IRQ: stack overflow: ear stack overflow (cur: eneral protection fault nable to handle kernel ouble fault: RTNL: assertion failed eek! page_mapcount(page) went negative! adness at NETDEV WATCHDOG ysctl table check failed : nobody cared IRQ handler type mism
   3184 root       20   0 24204  1704  1400 S  0.0  0.0  0:47.31 /usr/lib/systemd/systemd-logind
   3185 root       20   0 24384  2352  1692 S  0.0  0.0  0:00.91 /usr/sbin/smartd -n -q never
   3186 libstorag  20   0  8532   804   652 S  0.0  0.0  0:04.07 /usr/bin/lsmd -d
   3187 root       20   0  6472   724   552 S  0.0  0.0  1:31.69 /sbin/rngd -f
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
wsl-iit-kn 0:thomas@wsl-iit-knl01:~/src/QuantLib/build*Z 1:thomas@wsl-iit-knl01:~/src/Dixon/benchmarks/SVD-          18:41:17 up 7 days, 13:51
```

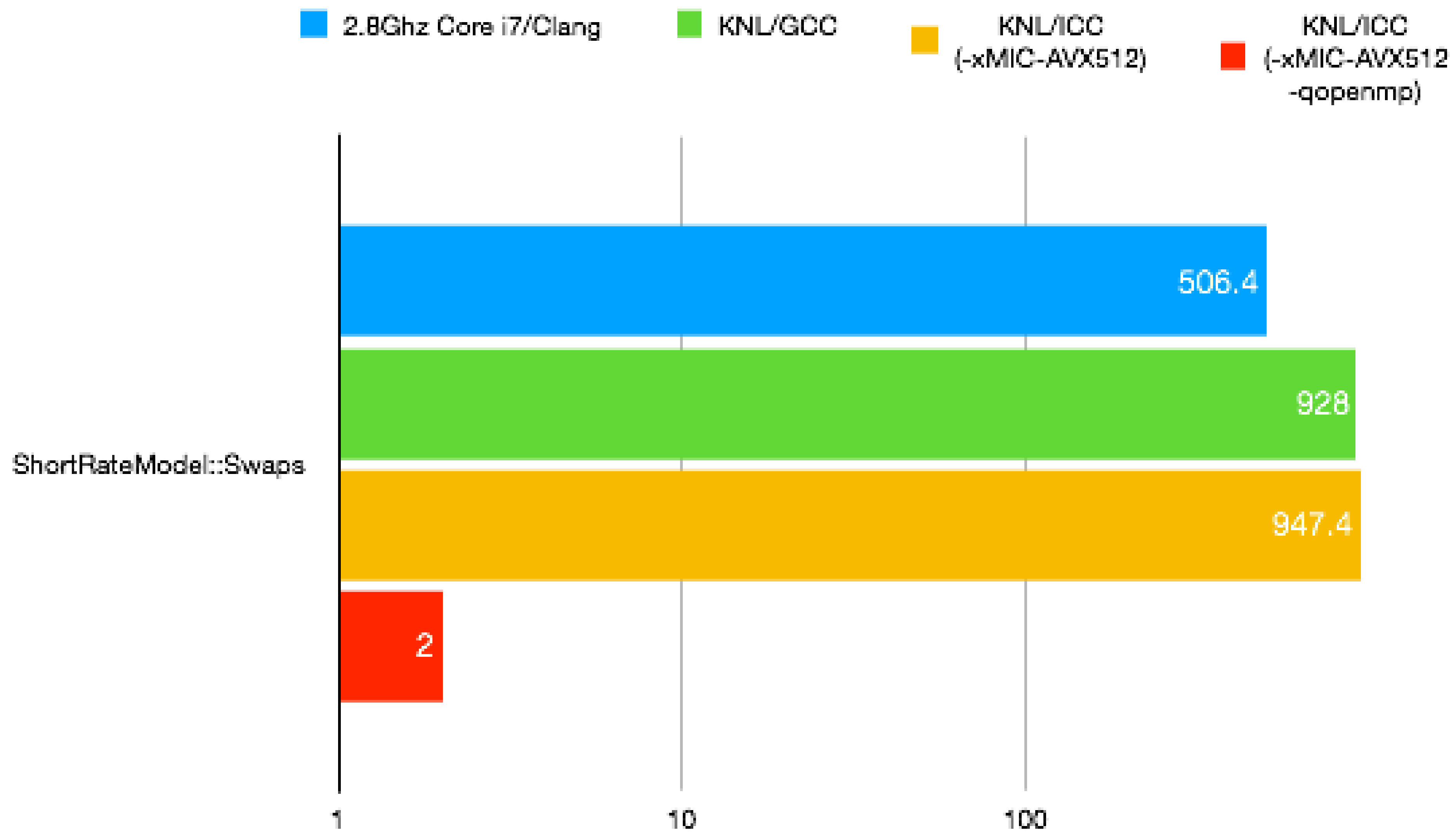
Scaling up

- Parallelization
  - 72 physical scalar cores
  - High bandwidth on-chip mesh
  - High bandwidth CPU adjacent memory (MCDRAM)
- Vectorization
  - Up to 16 double or 32 single precision floating point operations per vector unit, per clock
  - Two vector units per physical core

Intel Compiler Collection Auto Parallelization

- Using #pragma directives to guide code generation
- Can parallelize candidate loops with either OpenMP or TBB
  - OpenMP for homogenous workloads
  - TBB for variable workloads

Auto parallelization is not always a win



What’s going on here?

- Initial guess is that there’s not enough work per thread to offset the communication overhead

Intel Compiler Collection

- Compiler vectorization reports
  - -qopt-report, -qopt-report-phase
- Vectorization Advisor
  - Extensive tools for analyzing and recommending vectorization opportunities
  - Not explicitly tied to code generated by the Intel Compiler
- VTune profiler
  - Sample based and “uncore” counter profiling

What’s going on here?

- VTune Says -
  - ~21 of 272 logical cores utilized, ~7.8% utilization
  - ~57% time spent in serial code
  - ~50% scalar/50% SIMD instruction mix
  - Significant fraction of parallel time is load-imbalanced
  - 83% of pipeline slots remain empty

What’s going on here?

- Top 3 serial hotspots -
  - QuantLib::TrinomialTree::TrinomialTree
  - QuantLib::TreeLattice<QuantLib::OneFactorModel::ShortRateTree>::computeStatePrices
  - malloc

QuantLib is a non-trivial codebase

- Approximately 2280 source files, ~300k lines of code
- Makes heavy use of allocations, and shared pointers
  - Difficult to optimize without significant restructuring
- Detailed optimization of QuantLib is outside the scope of the work for this project

Narrowing the scope

- Currently evaluating optimization of
  - Heston model
  - SVD model w/parallel RNG
- Stripped down implementations, not part of QuantLib
- Explicit use of Intel’s Math Kernel Libraries where appropriate

Early results
Scaling individual models

- Heston Model
  - Peak core utilization ~14%,
  - 75% scalar to 25% SIMD instruction mix
- SVD & Parallel RNG
  - Peak core utilization ~19%
  - 100% Packed SIMD instruction mix

Next steps

- Long term goal is not focused on individual model optimization
  - Enable Quants to describe compute intensive problems in terms of high level composition of numerical algorithms
- Focus evaluation on scaling up compute utilization via explicitly parallel work partitioning

Next steps (cont)

- Custom C++ Allocators
  - Using C++ vector types backed by MCDRAM for frequently accessed data
  - Align vector types on cache-line boundary
- Non-Temporal Store
  - Avoid perturbing cache with data that is only ever written (e.g. result sets)