APPENDIX A
ARTIFACT DESCRIPTION APPENDIX: EXPLORING APPLICATION PERFORMANCE ON FAT-TREE NETWORKS IN THE PRESENCE OF CONGESTION

A. Abstract

We provide information about the three applications used in this study, how they were run. We also provide additional details on Quartz, the system used for the experiments discussed in the poster.

B. Description (AMG)

1) Check-list (artifact meta information):

- **Program: AMG2018**
- **Compilation: Included Makefile, Intel compiler (icc) version 18.0.1**
- **Run-time environment: Intel MPI 2018.0**
- **Execution:** `srun -N 224 -n 8064 ./amg -problem 2 -n 64 72 128 -P 24 21 16`
- **Publicly available?: Yes**

2) How software can be obtained (if available): Part of the CORAL 2 benchmark suite: https://asc.llnl.gov/coral-2-benchmarks/downloads/AMG-master-5.zip

3) Figure of merit calculation: We used the application-calculated figure of merit (`FOM_2`).

C. Description (UMT)

1) Check-list (artifact meta information):

- **Program: UMT2013**
- **Compilation: Included Makefile, Intel compilers (mpiicpc, ifort) version 18.0.1**
- **Run-time environment: mvapich2 version 2.2**
- **Execution:** `srun -N 224 -n 8064 ./SuOlsonTest custom_224node_8064MPI.cmg 16 2 16 8 4`
- **Publicly available?: Yes**

2) How software can be obtained (if available): Part of the CORAL benchmark suite: https://asc.llnl.gov/CORAL-benchmarks/Throughput/UMT2013-20140204.tar.gz

3) Problem size details: We used a problem with a $16 \times 21 \times 24$ decomposition, and $10 \times 10 \times 10$ zones per MPI rank. We ran with 16 groups, the product quadrature, an order of 16, 8 polar angles, and 4 azimuthal angles.

4) Figure of merit calculation: We used the application-calculated figure of merit.

D. Description (pF3D)

1) Check-list (artifact meta information):

- **Program: pF3D**
- **Compilation: Included build script `build-toss3.sh`, Intel compilers version 18.0.1, FFTW version 3**
- **Run-time environment: mvapich2 version 2.2, Intel Math Kernel Library version 11.3.3**
- **Execution:** `./Mtoss3_pf3d.ksh`
- **Publicly available?: No**

2) Problem size details: We used a problem with a $18 \times 8 \times 56$. So that pF3D terminated in approximately the same amount of time as the other applications, we modified the default problem to terminate after two rounds.

3) Figure of merit calculation: We used a constant divided by the sum of the communication time and the computation time (both computed and printed by pF3D) to finish the two rounds, excluding time spent in checkpoint I/O.

E. Experiment workflow

We used a script to generate run scripts for each experiment. When an experiment called for n nodes per leaf switch, we took the n lowest-numbered functioning nodes. In the case that one or more nodes were down for maintenence, the bully that was supposed to run on $31 - n$ nodes ran on one fewer. This happened rarely enough that we believe it did not cause any major difference.

All experiments were repeated five times, though for various reasons, not all runs produced a result. All figures of merit mentioned in the poster and extended abstract are from at least three repetitions. Tests of significance take into account the number of trials and the spread between them.

Prior to running any experiments, a single-node AMG problem was run on each node that we planned to use during the actual experiments. Approximately five nodes performed significantly worse than the mean value. The actual experiments specifically avoided these slow nodes to reduce variation due to node choice.

F. System details

As mentioned in the extended abstract, all of our tests were run on the Quartz cluster at LLNL. Quartz is a 3-level fat-tree using 100 Gbps Omni-Path with FTree routing to connect its 2,688 nodes together. Each node contains two 18-core 2.1 GHz Broadwell processors. The network has a 2 to 1 tapper at the leaf switches and 32 nodes per leaf switch. Each leaf switch has 31 user nodes and 1 system node. The second level switches connect 256 nodes together with two links to each leaf switch.

In other words,

- Each node is connected up to 1 leaf switch by a single link.
- Each leaf switch is connected to 8 second-level switches by double links (two ports, 200G total bandwidth).
- Each second-level switch is connected to 16 core module switches by single links. Note that all 16 are in the same core switch.

In the other direction,

- Each core module switch is connected to 42 second-level switches by single links.
- Each second-level switch is connected to 8 leaf switches by double links.
- Each leaf switch is connected to 32 nodes.

Quartz is comprised of 10 groups of 256 nodes and one group of 128 nodes. To avoid another source of variability, we did not use any nodes in the 128-node group.