HOW PENGUIN COMPUTING TUNED ITS NEWEST HPC CLOUD CLUSTER TO OPTIMALLY RUN THE WRF APPLICATION

Massimo Malagoli, Phd. MA, Sr. Cloud Architect
Penguin Computing

About

The Weather Research and Forecasting (WRF) model, a mesoscale numerical weather prediction system for atmospheric research and weather forecasting, is critical to researchers who want to simulate weather and climate models. It is also one of the most widely used applications on Penguin Computing® On-Demand™ (POD™) Penguin Computing's solution for customers who want to utilize a bare-metal, high-performance computing (HPC) computing environment without having to invest in on-premise infrastructure. To ensure that researchers reach the most accurate results possible, Penguin Computing engineers tuned the newest POD cluster, called MT2, to run optimally WRF.

The System

The Penguin Computing® On-Demand™ (POD™) public HPC cloud combines non-virtualized, bare-metal compute nodes, low-latency network, and fast storage with an optimized software stack and end-user applications specifically tuned for the hardware platform.

The MT2 cluster is powered by the B30 class of compute nodes, featuring Dual Intel® Xeon® E5-2680 v4 Broadwell processors with 28 non-hyperthreaded cores per node, 256 GB of DDR4 RAM and an Intel® Omni-Path Architecture (Intel® OPA) low-latency, non-blocking, 100Gb/s fabric. High speed storage is provided by a Lustre® parallel file system, delivered through the Penguin Computing® FrostByte™ storage solution.

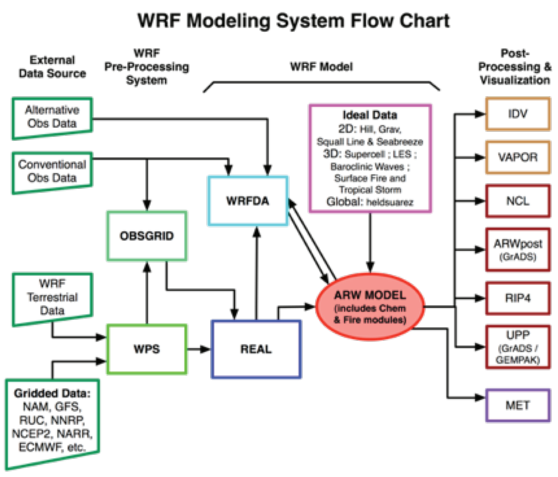
The Penguin Computing team used the version 3.8.1 of the WRF code, built with the Intel® Parallel Studio XE 2017 suite of compilers, the Intel® MPI message passing library for distributed memory parallelism (dmp) and OpenMP for shared memory parallelism (smp). The same compilers and MPI libraries were used for building the external libraries needed by WRF, for instance NetCDF.

We built a hybrid dmp + smp binary that can leverage both the distributed and shared memory models. Testing and tuning was done with the 4dbasic WRF benchmark, which is a test case designed to represent a typical end user scenario for regional weather modelling. Jobs were run using 5 of the B30 compute nodes, corresponding to 140 total processors. In a parallel run, WRF implements domain decomposition to divide the computational region into tasks, with one task assigned to each MPI rank of the job. Each task can be further divided into tiles, with the number of tiles set to 1 by default.

Due to WRF sensitivity to memory bandwidth, to ensure maximum efficiency it is very important for a tile to be small enough to fit inside the cache of the processor and/ or core. If the default tile is too large, the number of tiles can be increased (and thus the size of each tile can be reduced) by defining the numtiles input parameter or by setting the WRF_NUM_TILES environment variable.

The optimal number of tiles depends on the characteristics of the model and on the hardware. For the present case we found the optimal number of tiles to be 16, so all the jobs reported here were done with WRF_NUM_TILES=16.

except:
print("please select exactly two objects, the last one gets the mo
OPERATOR CLASSES
# Mirror Tool


WRF Modeling System Flow Chart
External Data Source
WRF Pre-Processing System
WRF Model
Post-Processing & Visualization
Alternative Obs Data
Conventional Obs Data
WRF Terrestrial Data
Gridded Data: NAM, GFS, RUC, NNRP, NCEP2, NARR, ECMWF, etc.
OBSGRID
WPS
WRFDA
REAL
Ideal Data 2D: Hill, Grav, Squall Line & Seabreeze 3D: Supercell ; LES ; Baroclinic Waves ; Surface Fire and Tropical Storm Global: heldsuarez
ARW MODEL (includes Chem & Fire modules)
IDV
VAPOR
NCL
ARWpost (GrADS)
RIP4
UPP (GrADS / GEMPAK)
MET

The Experiment

We started our test by running a pure MPI job; i.e., starting 140 MPI ranks over five B30 nodes with a single thread each. We used the default Intel MPI settings for process pinning, which resulted in each MPI rank pinned to a single CPU core.

This job completed in 57 minutes, which established the baseline for our study and also provided a useful term of comparison with the previous generation of POD hardware, the T30 cluster. The T30 features dual Intel® E5-2600v3 Haswell processors with 20 cores per node, 128GB RAM, an Intel® QDR InfiniBand 40Gb/s intercon nect, and 10GigE data network. The equivalent job on the T30 cluster (140 MPI ranks over 7 T30 nodes, single thread with core pinning) completed in 66 minutes and 7 seconds, meaning the B30 nodes delivered a nearly 16% speedup over the T30 class.

We then started working on improving the performance of the B30 jobs: a first source of inefficiency could be contention between the WRF processes and the system processes on the compute nodes, in particular the processes used by the OPA drivers. On the B30 cluster, OPA provides not only the low latency interconnect used by MPI, but also the high speed Lustre storage. Thus, both communi cation and I/O processes generate a load on the OPA system.

In our baseline run, all the cores are used for the WRF computation. In such cases, inevitably, some of the compute processes will need to share resources with OPA. This disrupts the processor cache and, given the sensibility of WRF to memory bandwidth, can be a source of inefficiency.

We decided to trade compute processes for a more efficient use of the cache: we did a second run starting only 130 MPI ranks, thus leaving 2 cores per node for the OPA workload. We modified the pinning map of the job to make sure that the first core on each of the dual Broadwell chips was not used by the job (these are core 0 and core 14 in the Intel MPI pinning map).

We realized this with the following commands:

```
export I_MPI_JOB_RESPECT_PROCESS_PLACEMENT=0
export I_MPI_PIN_PROCESSOR_EXCLUDE_LIST=0,14
mpirun -perhost 26 -np 130 ./wrf.exe
```

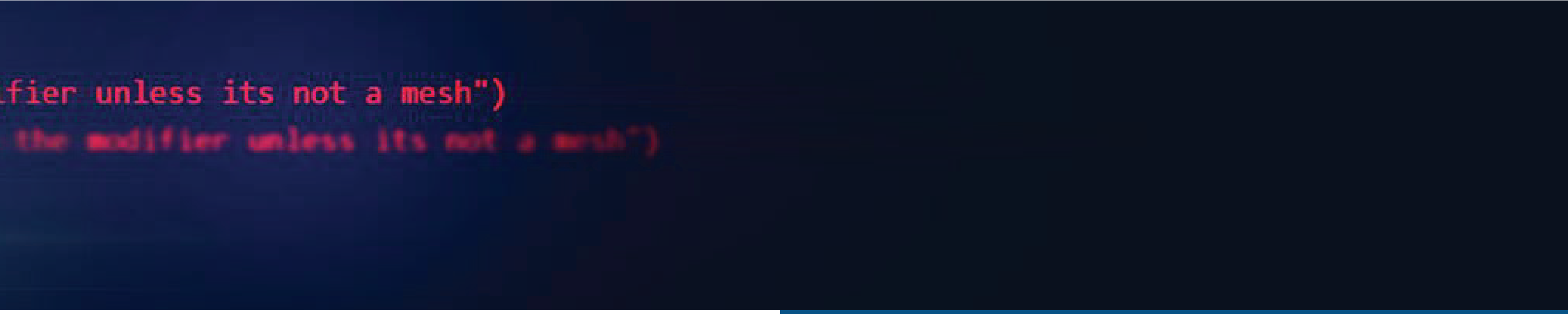
The first environment variable tells Intel MPI not to use the default process placement map (this is the map created by the job scheduler), while the second variable leaves cores 0 and 14 unused by the job and thus free for the OPA load. The mpirun command starts 130 MPI ranks, 26 on each B30 node. This run completed in 57 minutes and 42 seconds, just a little slower than the baseline run.

Now that the cache is used more efficiently, we could try to utilize some of the cores left free in the previous run for the job I/O, using WRF asynchronous I/O quilting facility. Since the I/O processes do not do any computation, they can share cores with the OPA processes.

The Results

We set up a run using 135 MPI ranks, of which 5 (one per node) were used for the I/O quilting. In this run only core 0 of each node was left free:

```
export I_MPI_JOB_RESPECT_PROCESS_PLACEMENT=0
export I_MPI_PIN_PROCESSOR_EXCLUDE_LIST=0,
export OMP_NUM_THREADS=2
mpirun -perhost 27 -np 135 ./wrf.exe
```



I/O quilting with 5 processes was configured with the WRF input parameter nio_tasks_per_group = 5. This run completed in 52 minutes and 49 seconds, nearly 8% faster than the baseline run.

As a final experiment, we tried a hybrid dmp + smp run, reducing the number of MPI ranks and increasing the number of threads. Here, the best results were obtained with 65 MPI ranks (13 per node), 2 threads for each MPI rank, and 5 I/O processes. In this run two cores per node were left free:

```
export I_MPI_JOB_RESPECT_PROCESS_PLACEMENT=0
export I_MPI_PIN_PROCESSOR_EXCLUDE_LIST=0,1
export OMP_NUM_THREADS=2
mpirun -perhost 13 -np 65 ./wrf.exe
```

This run completed in 52 minutes and 57 seconds, indicating that, for this workload, pure MPI and hybrid runs provide virtually the same performance.

In conclusion, we have tuned a WRF workload on a Broadwell cluster with OPA fabric: under-subscribing the nodes to leave one or two cores free for the OPA loads provides for a more efficient use of the processor cache. Further gains can be achieved by using some of the free cores for I/O quilting. Pure dmp and hybrid dmp + smp runs provide virtually the same performance for the 4dbasic benchmark.

Researchers using POD to run WRF will get all the benefits of cloud and quickly being able to access a powerful, bare metal HPC solution but also be able to get even greater speed on a cluster optimized for the application that is core to their success.

Penguin Computing on Demand (POD)
This bare-metal, HPC computing environment in the cloud gives you results without having to invest in on-premise infrastructure. In addition to POD's pay-per use basis, you also get scalability, performance and free support from Linux experts in a secure, on-demand environment.

Penguin Scyld Cloud Workstation
This remote desktop solution delivers real-time interactive 2D desktop GUIs and 3D visualization through a standard browser without plugins.

Penguin Computing Capital
Penguin Computing offers financing on any advanced product or service to ensure that your projects achieve results more quickly

Learn More
Learn more about POD at **www.penguincomputing.com/pod**

For pricing on your specific design needs, contact a representative by email at **podsales@penguincomputing.com** or call **1-888-PENGUIN (736-4846).**

Purchase with Financing
Finance products, services, even soft costs with Penguin Computing Capital. Choose from options such as no money down, flexible billing choices, extended repayment timelines, and a variety of end-of-term alternatives.

About Penguin Computing, a SMART Global Holdings Company
Penguin Computing, a U.S.-based global provider of high-performance computing (HPC), artificial intelligence (AI), and data center solutions, has been serving industry for over 20 years with more than 2,500 customers in 40 countries across eight major vertical markets. Penguin Computing offers a comprehensive portfolio of hardware and software including solutions based on the Open Compute Project (OCP), as well as extensive services including financing, and top-rated customer support. Penguin Computing products include Linux-based servers, software, integrated turn-key clusters, enterprise-grade storage, and bare metal HPC, all available in hardware or cloud-based solutions via Penguin Computing® On-Demand™ (POD). Penguin Computing is a subsidiary of SMART Global Holdings, Inc., and the cornerstone of SMART's newest business unit, Specialty Compute & Storage Solutions (SCSS).