EESSI at Microsoft Azure

September 2022, hugo.meiland@microsoft.com

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

HPC at Microsoft

- From explaining we actually run Linux to
  - #10 in Top500 (Nov 2021)
  - #26,27,28,29 in Top500 (June 2021)
  - Supercomputers for several large customers

- Specialty HPC SKU's
  - 5/6 generations of CPU, 7-9 generations of GPU
  - Nvidia/Mellanox InfiniBand in most of these VM's

- 2 main scenarios:
  - Lift & shift: traditional schedulers Slurm, PBS, LSF & posix filesystems
  - Cloud native: serverless & object/blob storage

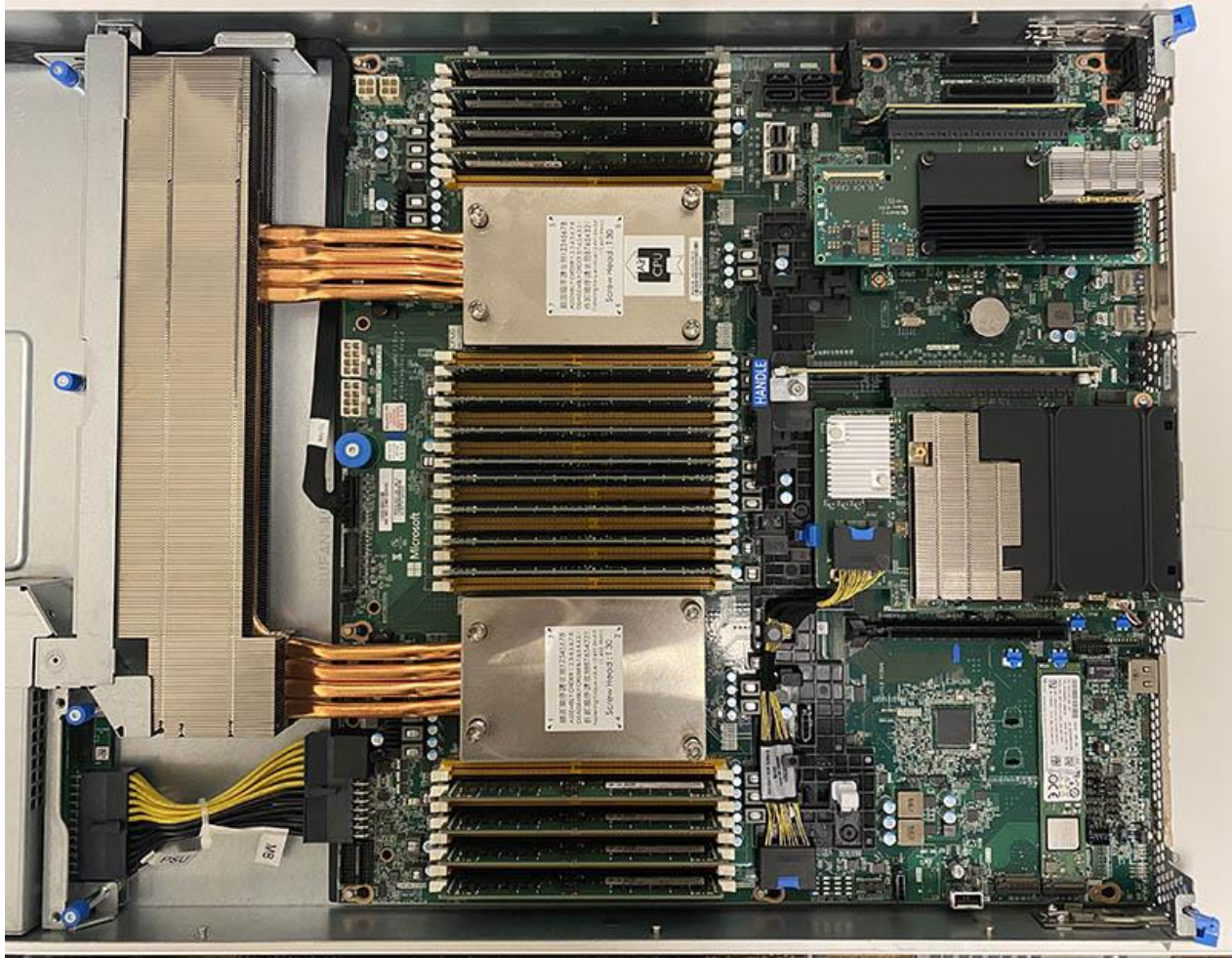
HPC vm fleet (InfiniBand enabled only…)

VM	Cpu arch	Mem	Mem bw	InfiniBand	Local Scratch	Remarks
~~H16(m)r~~	~~Intel Haswell~~	~~112/224~~		~~56 Gb/s FDR~~	~~2 TB~~	~~EOL August 2022~~
HC44rs	Intel Skylake	352 (8)		100 Gb/s EDR	700 GB	
HB60	AMD Naples	228 (4)	260 GB/s	100 Gb/s EDR	700 GB	
HB120_v2	**AMD Rome**	**456 (4)**	**350 GB/s**	**200 Gb/s HDR**	**900 GB**	
HB120_v3	**AMD Milan(-X)**	**448 (4)**	**350 GB/s**	**200 Gb/s HDR**	**2.1 TB**	**Migrating to -X**

VM	Cpu arch	Mem	GPU	InfiniBand	Local Scratch	Remarks
~~NC24r~~	~~Intel Broadwell~~	~~224~~	~~K80 (4x)~~	~~56 Gb/s FDR~~	~~1.44 TB~~	
NC24rs_v2	Intel Broadwell	448	P100 (4x)	56 Gb/s FDR	3 TB	
NC24rs_v3	Intel Broadwell	448	V100 (4x)	56 Gb/s FDR	3 TB	
ND24rs	Intel Broadwell	448	P40 (4x)	56 Gb/s FDR	3 TB	
ND40rs_v2	Intel Skylake	672	V100 (8x)	100 Gb/s EDR	2.9 TB	NVlink
ND96asr_A100_v4	AMD Rome	900	A100 (8x)	200 Gb/s HDR (8x)	6.5 TB	40 GB A100 + NV
ND96amsr _A100_v4	AMD Rome	1900	A100 (8x)	200 Gb/s HDR (8x)	6.5 TB	80 GB A100 + NV

Infiniband in Azure

- InfiniBand for MPI/NCCL
  - Not for storage / heterogenous
- Stamps == cluster == IB connectivity
- SR-IOV:
  - 387e:00:02.0 Infiniband controller: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Function]
- InfiniBand Partitions
  - Subnet manager is provided
  - No access to vlane 0
    - So no ibtracert and friends
- Use through Availibility Zone or VMSS
  - Azure Cyclecloud for orchestration
  - we can combine multiple vmss in single cluster



© ServeTheHome

CVMFS repo or EESSI

- Several tries based on EasyBuild: arch vs skus? which OS? which compilers?
  - Our team can only run “best effort” services
  - Figuring out customer expectations
    - Do you really need compiler XYZ / flag PQR?
    - Or is ml load <application>, mpirun <application> compelling enough?
    - Better to build containers?

- Tracking / (working on) EESSI:
  - Integrated into Az-HOP today
  - Need to double down on CPU detection, GPU and end-2-end runs
  - (upcoming hackathon!)

EESSI client

- Client (with EESSI config)
  - Integrated into Az-HOP (HPC Ondemand Platform)
  - Testing in WSL2, lack of automount is still a gap (wsl2_config not working for me… mount –t cvmfs does…)

hpcadmin@hc44rs-1:~

```
------------------------------------- /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/x86_64/intel/skylake_avx512/modules/all -------------------------------------
   ant/1.10.8-Java-11                                          Java/11.0.2                                (11)    ParaView/5.8.0-foss-2020a-Python-3.8.2-mpi
   Arrow/0.17.1-foss-2020a-Python-3.8.2                        jbigkit/2.1-GCCcore-9.3.0                          PCRE/8.44-GCCcore-9.3.0
   Bazel/3.6.0-GCCcore-9.3.0                                   JsonCpp/1.9.4-GCCcore-9.3.0                        PCRE2/10.34-GCCcore-9.3.0
   Bison/3.5.3-GCCcore-9.3.0                                   LAME/3.100-GCCcore-9.3.0                           Perl/5.30.2-GCCcore-9.3.0
   Boost/1.72.0-gompi-2020a                                    libarchive/3.5.1-GCCcore-10.3.0                    Perl/5.32.1-GCCcore-10.3.0                        (D)
   cairo/1.16.0-GCCcore-9.3.0                                  libcerf/1.13-GCCcore-9.3.0                         pixman/0.38.4-GCCcore-9.3.0
   CGAL/4.14.3-gompi-2020a-Python-3.8.2                        libdrm/2.4.100-GCCcore-9.3.0                       pkg-config/0.29.2-GCCcore-9.3.0
   CMake/3.16.4-GCCcore-9.3.0                                  libevent/2.1.11-GCCcore-9.3.0              (L)     pkg-config/0.29.2-GCCcore-10.3.0
   CMake/3.20.1-GCCcore-10.3.0                          (D)    libevent/2.1.12-GCCcore-10.3.0             (D)     pkg-config/0.29.2                                 (D)
   code-server/3.7.3                                           libfabric/1.11.0-GCCcore-9.3.0             (L)     pkgconfig/1.5.1-GCCcore-9.3.0-Python-3.8.2
   DB/18.1.32-GCCcore-9.3.0                                    libfabric/1.12.1-GCCcore-10.3.0            (D)     PMIx/3.1.5-GCCcore-9.3.0                          (L)
   DB/18.1.40-GCCcore-10.3.0                            (D)    libffi/3.3-GCCcore-9.3.0                           PMIx/3.2.3-GCCcore-10.3.0                         (D)
   double-conversion/3.1.5-GCCcore-9.3.0                       libffi/3.3-GCCcore-10.3.0                  (D)     poetry/1.0.9-GCCcore-9.3.0-Python-3.8.2
   Doxygen/1.8.17-GCCcore-9.3.0                                libgd/2.3.0-GCCcore-9.3.0                          protobuf-python/3.13.0-foss-2020a-Python-3.8.2
   EasyBuild/4.5.0                                             libGLU/9.0.1-GCCcore-9.3.0                         protobuf/3.13.0-GCCcore-9.3.0
   EasyBuild/4.5.1                                      (D)    libglvnd/1.2.0-GCCcore-9.3.0                       pybind11/2.4.3-GCCcore-9.3.0-Python-3.8.2
   Eigen/3.3.7-GCCcore-9.3.0                                   libiconv/1.16-GCCcore-9.3.0                        pybind11/2.6.2-GCCcore-10.3.0                     (D)
   Eigen/3.3.9-GCCcore-10.3.0                           (D)    libjpeg-turbo/2.0.4-GCCcore-9.3.0                  Python/2.7.18-GCCcore-9.3.0
   ELPA/2019.11.001-foss-2020a                                 libpciaccess/0.16-GCCcore-9.3.0            (L)     Python/3.8.2-GCCcore-9.3.0
   expat/2.2.9-GCCcore-9.3.0                                   libpciaccess/0.16-GCCcore-10.3.0           (D)     Python/3.9.5-GCCcore-10.3.0-bare
   expat/2.2.9-GCCcore-10.3.0                           (D)    libpng/1.6.37-GCCcore-9.3.0                        Python/3.9.5-GCCcore-10.3.0                       (D)
```

hpcadmin@hc44rs-1:~

```
[EESSI pilot 2021.12] $ ml load WRF
[EESSI pilot 2021.12] $ which wrf.exe
/cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/x86_64/intel/skylake_avx512/software/WRF/3.9.1.1-foss-2020a-dmpar/WRFV3/main/wrf.exe
[EESSI pilot 2021.12] $
```

Using InfiniBand on Azure

```
#!/bin/bash
module load OSU-Micro-Benchmarks/5.6.3-gompi-2020a
scontrol show hostname $SLURM_JOB_NODELIST > hostfile
export OMPI_MCA_pml=ucx
mpirun --hostfile hostfile -n 2  --map-by ppr:1:node osu_bw
```

[EESSI pilot 2021.12] $ cat slurm-39.out

OSU MPI Bandwidth Test v5.6.3

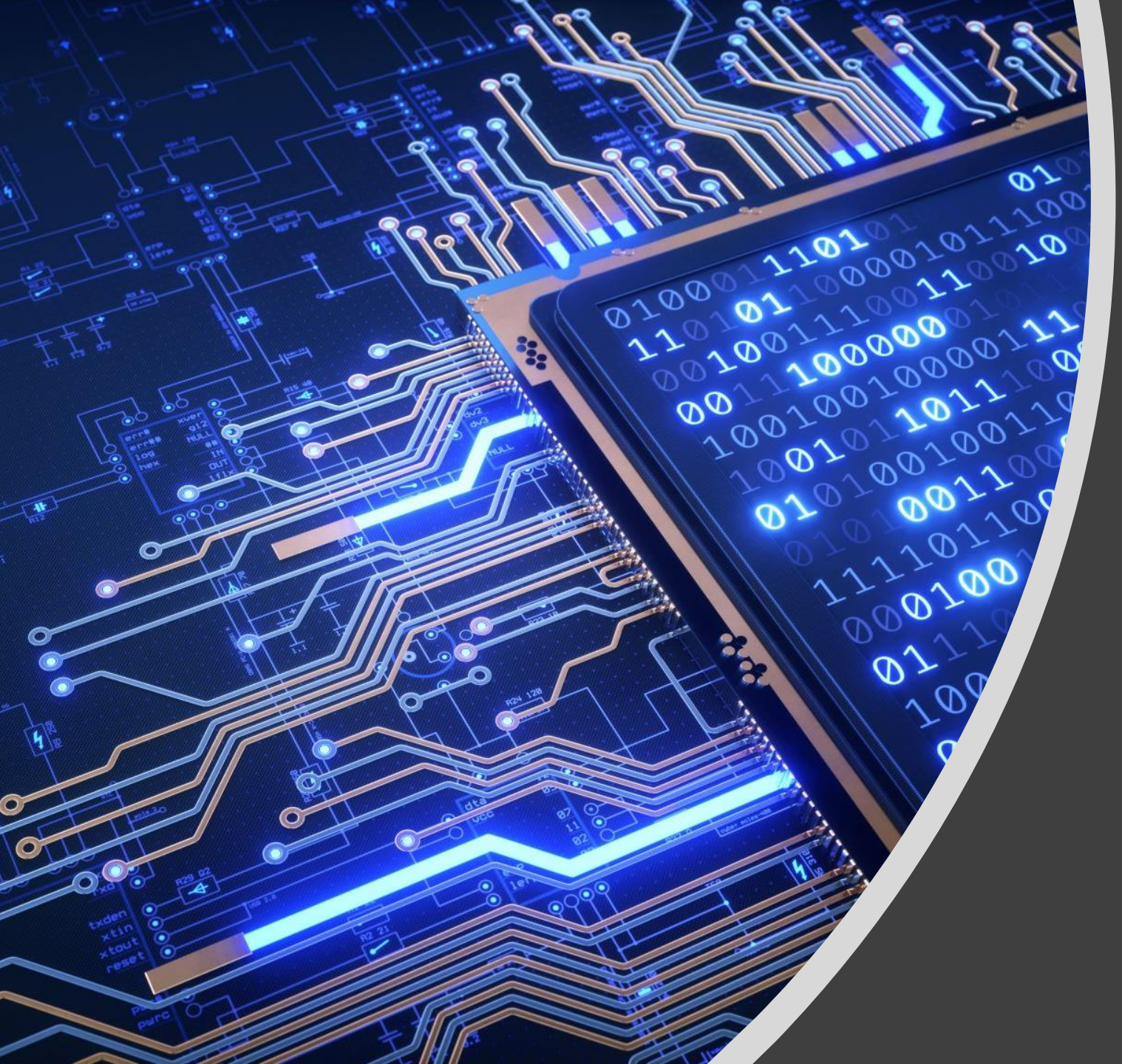
# Size	Bandwidth (MB/s)
1	4.39
2	8.84
4	17.60
8	35.28
16	69.55
32	141.00
64	254.08
128	475.12
256	833.28
512	1505.10
1024	2603.79
2048	3944.87
4096	5403.71
8192	7621.57
16384	7533.24
32768	10164.94
65536	11419.85
131072	11607.19
262144	11511.06
524288	11538.26
1048576	11538.91
2097152	11408.83
4194304	11321.95

OSU MPI Latency Test v5.6.3 **using RDMA**

# Size	Latency (us)
0	1.53
1	1.52
2	1.52
4	1.52
8	1.52
16	1.52
32	1.58
64	1.76
128	1.84

OSU MPI Latency Test v5.6.3 **using TCP**

# Size	Latency (us)
0	74.98
1	72.37
2	68.46
4	76.83
8	73.02
16	69.52
32	70.65
64	80.39
128	76.94



Leveraging EESSI for WRF simulations at scale on Azure HPC

davide.vanzo@microsoft.com

hugo.meiland@microsoft.com

Microsoft Azure

Running WRF3 on Zen3

```
#!/bin/bash
#SBATCH --nodes=<N>
#SBATCH --tasks-per-node=120

export EESSI_SOFTWARE_SUBDIR_OVERRIDE=x86_64/amd/zen3     #archspec/pku
source /cvmfs/pilot.eessi-hpc.org/versions/2021.12/init/bash
module load WRF/3.9.1.1-foss-2020a-dmpar

mkdir wrf_job_2.5
cd wrf_job_2.5
ln -s `dirname $(which wrf.exe)`/../run/* .
rm namelist.input
ln -s ~/WRF_test/bench_2.5km/* .

export OMPI_MCA_pml=ucx      #fixed in foss2021a/OpenMPI4.1.1
time mpirun wrf.exe
```

EESSI / software-layer Public

Code | Issues 54 | Pull requests 14 | Actions | Projects | Wiki | Security | Insights

alternative to archspec for detecting cpu arch Tests for eessi_archdetect.sh #21

Summary

Jobs

- build (x86_64/intel/haswell)
- build (x86_64/intel/skylake_avx512)
- build (x86_64/amd/zen2)
- build (x86_64/amd/zen3)
- build (ppc64le/power9le)
- build (aarch64/graviton2)
- build (aarch64/graviton3)
- build (aarch64/arm/neoverse-n1)

Triggered via pull request 6 minutes ago

hmeiland synchronize #187 `hmeiland:feature-archdetect`

Status: Success

Total duration: 22s

tests_archdetect.yml

on: pull_request

Matrix: build

8 jobs completed

Show all jobs

CVMFS code + Stratus network

Tags (edit) : created-by : hugo | state : production | do-not-delete : please

Resources | Recommendations (1)

Filter for any field... | Type equals all | Location equals all | Add filter

Showing 1 to 5 of 5 records. | Show hidden types | No grouping

Name	Type	Location
cvmfs	Traffic Manager profile	Global
cvmfs	Storage account	West Europe
cvmfs-kv	Key vault	West Europe
cvmfseastus	Storage account	East US
cvmfssouthcentral	Storage account	South Central US

- Added support for Azure Blob next to S3 (okt 2021?)
- Cvmfs_server can build stratum0 directly on Azure Blob
  - With keys in keyvault, build machine is expandable
- Adding Azurite tests for Azure Blob over HTTPS
  - Technically verified this week ;)

- **Todo: set up stratum1 based on Azure Blob**
- Using traffic manager i.s.o. geo-ip
- Sync containers i.s.o. stratum0 -> stratum1

Next steps

- Extending work with/on EESSI with focus on WRF & MD
  - To better support end-to-end applications in Az-HOP
  - Including ReFrame (performance) testing for regression
  - Refresh CiTC work or switch to Magic Castle?

- Dive deeper and pick up learnings on container deployments
  - As alternative to container registry for HPC

- Happy to discuss/test/extend serverless CVMFS

Thank you!

hugo.meiland@microsoft.com