Installed oneAPI Base toolkit and oneAPI HPC toolkit on 9/2021 for test purposes. Installed on compute nodes, head nodes, Mark6'es.

Base toolkit: Develop performant, data-centric applications across Intel® CPUs, GPUs, and FPGAs with this foundational toolset. HPC toolkit: Build, analyze, and scale applications across shared- and distributed-memory computing systems.

For additional info from Intel see https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html

Complications: 1) not on Mark5?, 2) process placement on specific nodes is trickier with Intel mpirun than OpenMPI mpirun, 3) SLURM salloc/srun allocations should go via pmi1 or pmi2 ("srun --mpi=list" to see available MPI types).

Components

The installation is under */opt/intel/oneapi/*

The installation provides its own Intel IPP and Intel MPI Library.

It also comes with Intel VTune Profiler, Cluster Checker, Trace Analyzer and Collector, and Intel compilers (CC is *icc*, CXX is *icpc*).

Activate with: source /opt/intel/oneapi/setvars.sh

MPI code analysis, cf. also Intel code profiling pages:

```
mpirun -n 7 -perhost 1 --machinefile job.macines -l \
  vtune -quiet -collect hotspots -trace-mpi -result-dir /data/TESTS/
vtune/ \
  <program, program args>
```

Repository

See Intel web pages. For CentOS on our cluster got a by default disabled /etc/yum.repos.d/oneAPI.repo with:

```
[oneAPI]
name=Intel® oneAPI repository
baseurl=https://yum.repos.intel.com/oneapi
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://yum.repos.intel.com/intel-gpg-keys/
GPG-PUB-KEY-INTEL-SW-PRODUCTS.PUB
```

The repo should be enabled temporarily only! Avoids accidentally pulling any Intel-customized RHEL/CentOS packages into the vlbi cluster.

```
gxr -c "yum-config-manager --enable oneAPI" --group /hardware/nodes/
mark6,/hardware/nodes/compute
```

```
(gxr -c "yum install -y  ...)
gxr -c "yum-config-manager --disable oneAPI" --group /hardware/nodes/
mark6,/hardware/nodes/compute
```

Intel oneAPI Packages

Packages for devel nodes:

```
yum install intel-basekit
yum install intel-hpckit
yum install intel-oneapi-vtune-2021.7.1-492
```

Runtime packages for nodes:

```
gxr -c "yum install -y intel-oneapi-clck" --group /hardware/nodes/
mark6,/hardware/nodes/compute
gxr -c "yum install -y intel-oneapi-runtime-mpi \
  intel-oneapi-runtime-ipp \
  intel-oneapi-compiler-shared-runtime-2021.3.0.x86_64 \
  intel-oneapi-compiler-shared-common-runtime-2021.3.0.noarch \
  intel-oneapi-mpi-2021.3.1 \
  intel-oneapi-vtune-2021.7.1-492" --group /hardware/nodes/
mark6,/hardware/nodes/compute
# Dependency bug in runtimes, need to install also /opt/intel/oneapi/lib/
intel64/libimf.so via:
gxr -c "yum install -y intel-oneapi-runtime-compilers-2021.3.0" \
  --group /hardware/nodes/mark6,/hardware/nodes/compute
```

Same node packages also installed on io01, io02, io08.

InfiniBand CentOS Packages

These following CentOS packages appear to be required by Intel MPI middleware, especially libfabric and ucx, on all nodes:

```
yum install infiniband-diags fabtests
yum install libfabric ucx
```

Environment

Activate with:

```
source /opt/intel/oneapi/setvars.sh
```

Intel MPI test:

```
$ ssh oper@fxmanager
$ source /opt/intel/oneapi/setvars.sh
$ which mpirun
```

```
/opt/intel/oneapi/mpi/2021.3.1/bin/mpirun

$ cat > intel.hostfile <<EOF
fxmanager
mark6-01
node02.service
node67.service
node10.service
node11.service
node12.service
EOF

$ mpirun -print-rank-map -prepend-rank -v -n 7 -perhost 1 -machinefile
intel.hostfile /usr/bin/hostname
```

Environment - DiFX

The startdifx(.py) script of DiFX has to be modified slightly on one source code line:

```
cmd = 'mpirun -np %d --hostfile %s.machines ...
  change that into
cmd = 'mpirun -np %d --machinefile %s.machines ...
```

The usual setup_difx script(s) should have:

```
#!/bin/bash

## Get oneAPI environment
# Prepare IPPROOT, CMPLR_ROOT, I_MPI_ROOT, ...
if [[ "$I_MPI_ROOT" == "" ]]; then
        . /opt/intel/oneapi/setvars.sh
fi
if [[ "$IPPROOT" == "" ]]; then
        export IPPROOT=$ONEAPI_ROOT/lib/intel64/
fi
# Intel MPI
export OPENMPIROOT=$I_MPI_ROOT
# Intel Compilers
# Note on icc, vs icpc, vs icx, vs icpx, vs icl:
#  https://software.intel.com/content/www/us/en/develop/articles/
porting-guide-for-icc-users-to-dpcpp-or-icx.html
export CC=icc
export CXX=icpc
export MPIXCC=${OPENMPIROOT}/bin/mpicxx
export MPICXX=${OPENMPIROOT}/bin/mpicxx

# Add Intel-libs path in case the current host has only the runtimes but
not entire
# devel toolkit installed; the automatic setvars.sh does not appear to
```

```
cover that case
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/intel/oneapi/lib/intel64/

## Rest of DiFX setup
# note: OpenMPI params in DIFX_MPIRUNOPTIONS and the OMPI_MCA_* env vars
probably irrelevat for Intel MPI
```