A few HPC topics for QE development

Nicola Varini
nicola.varini@epfl.ch
SCITAS, THEOS
EPFL
Path to Exascale?

Marmolada 3343 m

K2 8611 m
Unprecedented diversity

Industry switch to many-core architectures

- Unprecedented computing power for science/med apps
- Diverse architectural trends

- Unprecedented software challenges

Courtesy of Diego Rossinelli, Bridging the productivity gap in HPC
Productivity-performance gap

Fast evolution of computing hardware may lead to

- Frequent rewrite of software
- Unsustainable developing efforts
- Or a suboptimal use of the hardware

Courtesy of Diego Rossinelli, Bridging the productivity gap in HPC
From hardware to papers

- Goal: to minimize T software to paper
- Bug fixing and software testing, CI+?
- Software modularization
- Benchmark and profiling, top-down
- Miniapp and kernel patterns, bottom-up
- Identifying the critical components, FFT, algebra, IO
PW benchmarks

• The benchmark described below is done on Deneb at SCITAS
• 376 nodes
• Each node has 2 Ivy Bridge, each with 8 core and 64 GB of RAM
• Nodes are connected with Infiniband QDR
## PW benchmarks – BaTiO3
### 320 atoms

<table>
<thead>
<tr>
<th>Routine</th>
<th>mpi – 2 nodes</th>
<th>Openmp – 2 nodes</th>
<th>Mpi – 4 nodes</th>
<th>OpenMP – 4 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrons</td>
<td>4932.87</td>
<td>5278.90</td>
<td>2936.58</td>
<td>3412.46</td>
</tr>
<tr>
<td>Cegterg</td>
<td>4276.48</td>
<td>4052.28</td>
<td>2614.87</td>
<td>2773.67</td>
</tr>
<tr>
<td>H_psi</td>
<td>1225.81</td>
<td>1590.07</td>
<td>616.67</td>
<td>902.55</td>
</tr>
<tr>
<td>Cdiagc</td>
<td>2029.75</td>
<td>1274.28</td>
<td>1433.37</td>
<td>1260.25</td>
</tr>
<tr>
<td>add_vuspsi</td>
<td>323.67</td>
<td>328.57</td>
<td>176.67</td>
<td>164.90</td>
</tr>
<tr>
<td>calbec</td>
<td>456.47</td>
<td>464.12</td>
<td>255.97</td>
<td>239.14</td>
</tr>
<tr>
<td>fftw</td>
<td>518.20</td>
<td>827.63</td>
<td>256.89</td>
<td>515.41</td>
</tr>
<tr>
<td>fft_scatter</td>
<td>298.77</td>
<td>572.31</td>
<td>155.83</td>
<td>385.50</td>
</tr>
<tr>
<td>PAW</td>
<td>254.87</td>
<td>424.37</td>
<td>133.80</td>
<td>216.79</td>
</tr>
<tr>
<td>Routine - MPI</td>
<td>% - MPI</td>
<td>Routine - threads</td>
<td>% - threads</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>MPI_Barrier</td>
<td>23.59</td>
<td>pzheevd</td>
<td>12.94</td>
<td></td>
</tr>
<tr>
<td>Calbec</td>
<td>9.08</td>
<td>calbec</td>
<td>6.44</td>
<td></td>
</tr>
<tr>
<td>pzheevd</td>
<td>14.43</td>
<td>fft_scatter</td>
<td>7.93</td>
<td></td>
</tr>
<tr>
<td>add_vuspsi</td>
<td>6.34</td>
<td>update_distmat</td>
<td>4.88</td>
<td></td>
</tr>
<tr>
<td>s_psi</td>
<td>6.33</td>
<td>add_vuspsi</td>
<td>4.57</td>
<td></td>
</tr>
<tr>
<td>update_distmat</td>
<td>6.08</td>
<td>s_psi</td>
<td>4.54</td>
<td></td>
</tr>
<tr>
<td>MPI_Allreduce</td>
<td>5.53</td>
<td>all_reduce</td>
<td>4.06</td>
<td></td>
</tr>
<tr>
<td>MPI_Alltoall</td>
<td>5.25</td>
<td>h_psi_dot_v</td>
<td>3.59</td>
<td></td>
</tr>
<tr>
<td>h_psi_dot_v</td>
<td>4.66</td>
<td>MPI_Barrier</td>
<td>3.54</td>
<td></td>
</tr>
<tr>
<td>MPI_Bcast</td>
<td>3.17</td>
<td>funct.xc</td>
<td>5.46</td>
<td></td>
</tr>
<tr>
<td>Routine</td>
<td>mpi – 16 nodes</td>
<td>Openmp – 16 nodes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrons</td>
<td>4446.24</td>
<td>3923.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cgterg</td>
<td>3878.01</td>
<td>3333.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H_psi</td>
<td>1921.85</td>
<td>1450.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cdiaghg</td>
<td>350.54</td>
<td>277.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>add_vuspsi</td>
<td>268.86</td>
<td>281.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>calbec</td>
<td>938.50</td>
<td>986.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fftw</td>
<td>1168.10</td>
<td>518.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fft_scatter</td>
<td>1130.79</td>
<td>343.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Considerations on benchmarks

- OpenMP behaved reasonably well
- Manual benchmarking is a tedious procedure
- Hard to keep track of simulations
- AiiDA for benchmarks automatic benchmarks
- 1 input and many runs in parallel by changing pools, ndiag, ntg, ...
- Store the results in a database and share among the community

Automated Interactive Infrastructure and Database for Computational Science
Miniapp for FFT – kernel pattern

```fortran
do ibnd=1, m
    psic = (0.0d0,0.0d0)
    psic(nls(igk(1:n))) = psi(1:n,ibnd)
    call invffft('Wave',psic,dffts)
    do j=1, dffts%nnr
        psic(j) = psic(j)*v(j)
    enddo
    CALL fffft ('Wave', psic, dffts)
    do j=1, n
        hpsi(j,ibnd) = hpsi(j,ibnd) + psic(nls(igk(j)))
    enddo
enddo
```
Input/Output – HDF5 terminology

- **HDF5 group**: grouping structure containing instances of zero or more groups or datasets, together with supporting metadata.
  - Groups can be thought as unix directories and files.
- **HDF5 dataset**: a multidimensional array of data elements, together with supporting metadata.
- Any HDF5 group or dataset may have associated an attribute list with additional information about an HDF5 object.
- **Dataspace** specify the dimensionality of a dataset.
- **HDF5** allows you to read or write to a portion of a dataset by use of hyperslab selection.
TYPE HDF5_type
INTEGER(HID_T) :: file_id       ! File identifier
INTEGER(HID_T) :: dset_id       ! Dataset identifier
INTEGER(HID_T) :: file_space    ! Dataspace identifier in file
INTEGER(HID_T) :: mem_space     ! Dataspace identifier in memory
INTEGER(HID_T) :: plist_id      ! Property list identifier
CHARACTER(LEN=3) :: dsetname     ! Dataset name
INTEGER(HSIZE_T), DIMENSION(2) :: counts, counts_g, offset
INTEGER       :: comm
INTEGER       :: rank
END TYPE HDF5_type
HDF5 in pw – setup file and property

```fortran
subroutine setup_file_property_hdf5(hdf5desc, filename)
  use MPI
  implicit none
  type(HDF5_type), intent(inout) :: hdf5desc
  character(len=*) , intent(in) :: filename
  integer(HID_T) :: plist_id
  integer :: error, info
  info = MPI_INFO_NULL
  CALL h5pcreate_f(H5P_FILE_ACCESS_F, hdf5desc%plist_id, error) ! Properties for file creation
  CALL h5pset_fapl_mpio_f(hdf5desc%plist_id, hdf5desc%comm, info, error) ! Stores MPI IO communicator information to the file access property list
  CALL h5fcreate_f(filename, HSF_ACC_TRUNC_F, hdf5desc%file_id, error, access_prp = hdf5desc%plist_id) ! create the file collectively
  CALL h5fclose_f(hdf5desc%plist_id, error)
end subroutine setup_file_property_hdf5
```
HDF5 in pw – define dataset

```fortran
subroutine define_dataset_hdf5( hdf5desc)
  implicit none
  type(HDF5_type), intent(inout) :: hdf5desc
  integer :: error

  CALL hSscreate_simple_f(hdf5desc%rank, hdf5desc%counts_g, hdf5desc%filespace, error) !define HDF5 dataset
  CALL hSdc create F(hdf5desc%file_id, hdf5desc%dsetname, HST_NATIVE_DOUBLE, hdf5desc%filespace, &
                      hdf5desc%dset_id, error)
  CALL hSsclose_f(hdf5desc%filespace, error)

  CALL hSscreate_simple F(hdf5desc%rank, hdf5desc%counts, hdf5desc%memspace, error)
  CALL hSdget_space F(hdf5desc%dset_id, hdf5desc%filespace, error)
  CALL hSsselect_hyperslab F(hdf5desc%filespace, HSS_SELECT_SET_F, hdf5desc%offset, hdf5desc%counts, error) !create hyperslab to read from more than 1 proc
end subroutine define_dataset_hdf5
```
HDF5 in pw – write data

```
subroutine write_data_hdf5(hdf5desc, data, rank)
USE kinds, ONLY : DP
USE ISO_C_BINDING
implicit none

  type(HDF5_type), intent(inout) :: hdf5desc
  complex(kind=dp), intent(inout) :: data(:,:)
  integer, intent(in) :: rank

  !INTEGER, ALLOCATABLE :: data (:,:) ! Data to write
  integer :: error
  real(kind=dp) :: tmp

  integer(HID_T) :: complex_id, double_id
  integer(HSIZE_T) :: double_size, complex_size

  TYPE(C_PTR) :: f_ptr

  CALL h5pcreate_f(H5P_DATASET_XFER_F, hdf5desc%plist_id, error)
  CALL h5pset_dxpl_mpio_f(hdf5desc%plist_id, H5FD_MPIO_COLLECTIVE_F, error)

  f_ptr = C_LOC(data(1,1))
  CALL h5dwrite_f(hdf5desc%dset_id, H5T_NATIVE_DOUBLE, f_ptr, error,
                  file_space_id = hdf5desc%filespace, mem_space_id = hdf5desc%memspace,
                  xfer_prp = hdf5desc%plist_id)

end subroutine write_data_hdf5
```
HDF5 in pw – read data

```fortran
subroutine read_data_hdf5(hdf5desc, data, rank)
  type(HDF5_type), intent(inout) :: hdf5desc
  complex(kind=dp), intent(inout) :: data(:,::)
  integer, intent(in) :: rank
  integer :: error
  TYPE(C_PTR) :: f_ptr

  f_ptr = C_LOC(data(1,1))
  CALL H5dread_f(hdf5desc%dset_id, H5T_NATIVE_DOUBLE, f_ptr, error, &
               mem_space_id = hdf5desc%memspace, file_space_id = hdf5desc%filespace, &
               xfer_prp = hdf5desc%plist_id)

end subroutine read_data_hdf5
```
subroutine initialize_io_hpc(which, comm)

    implicit none
    integer, intent(in) :: which, comm

if(which.eq.1)
    call initialize_hdf5()
    evc_hdf5%dsetname="evc"
    evc_hdf5%comm=comm
    evc_hdf5%rank = 2
    CALL setup_file_property_hdf5(evc_hdf5, "rho_hdf5")
    CALL prepare_index_hdf5(npwx,off_npw,npw_g,evc_hdf5%comm,nproc)
    CALL set_index_hdf5(evc_hdf5, evc, off_npw, npw_g, 2)
    CALL define_dataset_hdf5(evc_hdf5)
else
endif
end subroutine initialize_io_hpc
CALL wfcinit()

#if defined __HDF5
    CALL initialize_io_hpc(1, world_comm)
    CALL write_data_hdf5(evc_hdf5, evc, mpime)
    CALL read_data_hdf5(evc_hdf5, evc, mpime)
#endif
Spack

- Proper software installation is problematic for any HPC center
- A flexible package manager designed to support multiple versions, configurations, platforms, and compilers. [http://llnl.github.io/spack](http://llnl.github.io/spack)

```
$ spack install --keep-stage espresso +mpi+openmp+scalapack
--- Installing espresso
--- mpich is already installed in /home/nvarini/spack/opt/spack/x86_64/gcc-4.8/mpich-3.1.4-xuwrgqga7xcmffxdtz63d6vm2blbdh
--- Trying to fetch from http://www.qe-forge.org/gf/download/frsrelease/204/912/espresso-5.3.0.tar.gz
#------------------------------------------------- 100.0%
--- Staging archive: /home/nvarini/spack/var/spack/stage/espresso-5.3.0-wuoi653a3axy5zhpaf2zes6mfpf4xcm5/espresso-5.3.0.tar.gz
--- Created stage in /home/nvarini/spack/var/spack/stage/espresso-5.3.0-wuoi653a3axy5zhpaf2zes6mfpf4xcm5.
--- No patches needed for espresso.
--- Building espresso.
--- Successfully installed espresso.
Fetch: 1m 8.21s. Build: 8m 24.70s. Total: 9m 32.91s.
```
```python
class Espresso(Package):
    """FIXME: put a proper description of your package here."""
    # FIXME: add a proper url for your package's homepage here.
    homepage = "http://quantum-espresso.org"
    url = "http://www.qe-forge.org/gf/download/frsrelease/204/912/espresso-5.3.0.tar.gz"

    version('5.3.0', '6848fecaeb118587d6be36bd10b7f2c3')
    variant('mpi', default=True, description='Build Quantum-ESPRESSO with mpi support')
    variant('openmp', default=False, description='Build Quantum-ESPRESSO with mpi openmp')
    variant('scalapack', default=False, description='Build Quantum-ESPRESSO with mpi openmp')

    # FIXME: Add dependencies if this package requires them.
    # depends_on('foo')
    depends_on('mpi', when='+mpi')

    # def install(self, spec, prefix):
    #     # FIXME: Modify the configure line to suit your build system here.
    #     configure('--prefix=%s' % prefix)
    #     # FIXME: Add logic to build and install here
    #     make()
    #     make('install')

    def install(self, spec, prefix):
        # TAU isn't happy with directories that have '@' in the path. Sigh.

        # TAU configure, despite the name, seems to be a manually written script (nothing related to autotools).
        # As such it has a few #peculiarities# that make this build quite hackish.
        options = ["-prefix=%s" % prefix,
                   "--enable-parallel"]

        if '+openmp' in spec:
            options.append('--enable-openmp')

        if '+scalapack' in spec:
            options.append('--with-scalapack=yes')
```