Βέλτιστες Πρακτικές

Χρήση Βιβλιοθηκών και Εφαρμογών

Δρ. Δημήτρης Ντελλής

GRNET

ntell [at] grnet.gr

European Union
European Regional Development Fund
ΠΕΠ
PRACEGR
NSRF 2007-2013

Περιεχόμενα

- Compilers
- MPI
- Βιβλιοθήκες
- Εφαρμογές

Compilers

- Επειδή όλες οι βιβλιοθήκες/εφαρμογές έχουν γίνει compile με gnu/4.9.2, intel/15.0.3, intelmpi/5.0.3, τα αντίστοιχα modules ενεργοποιούνται με το login
- Λεπτομέρειες για τους compilers δόθηκαν στις χτεσινές παρουσιάσεις.
- Εγκατεστημένοι Compilers
  - Intel 15.0.3 (default), 16.0.0
    - module load intel (intel/16.0.0)
    - icc, icpc, ifort
    - Βασικά Flags : -O3 -xCORE-AVX-I (-xAVX)
    - OpenMP : -openmp
  - GNU 4.9.2 (default), 4.9.3, 5.1.0, 5.2.0
    - module load gnu (gnu/4.9.3, κλπ.)
    - gcc, g++, gfortran
    - Βασικά Flags : -O3 -mavx -march=ivybridge -mtune=ivybridge
    - OpenMP : -fopenmp

MPI

- Intel MPI
  - module load intelmpi (intelmpi/5.1.1)
  - Με χρήση του ενεργού GNU version
    - mpicc, mpicxx, mpif90
  - Με χρήση του ενεργού Intel version
    - mpiicc, mpiicpc, mpiifort (mpi+όνομα intel compiler)
- OpenMPI
  - module load openmpi/1.10.0/gnu
  - module load openmpi/1.10.0/intel
  - mpicc, mpicxx, mpif90
  - Με οποια από τις versions (gnu/intel)
  - OMPI_CC=icc mpicc, OMPI_CXX=icpc mpicxx, OMPI_FC=ifort mpif90

- MVAPICH2
  - module load mvapich2/gnu/2.2.2a
  - module load mvapich2/intel/2.2.2a
  - mpicc, mpicxx, mpif90
  - Με οποια από τις versions (gnu/intel)
  - mpicc -cc=icc, mpicxx -cxx=icpc, mpif90 -fc=ifort

MPI
Επιπλοκές με F90

- Κύρια επιπλοκή η χρήση του mpi module από f90
- Η χρήση modules στην f90 απαιτεί να έχουν γίνει compile με τον ίδιο compiler και version. Ο OpenMPI/1.10.0 έχει γίνει compile με gnu/4.9.2 ή intel/15.0.3
- Ο intelmpi έχει compiled versions του mpi.mod για gnu 4.1 - 4.9 => δεν μπορείτε να χρησιμοποιήσετε π.χ gnu/5.2.0
- Αντιθέτως, η 16.0.0 έχει υποστήριξη μέχρι και 5.2.x
- Αν στον κώδικα F90 υπάρχει statement : use mpi τότε πρέπει να χρησιμοποιηθούν αυτές οι version των compilers.
- Εκτός και ....
- Αντικαταστήστε το use mpi με include 'mpif.h'
- Αντίστοιχα θέματα υπάρχουν με τα modules άλλων βιβλιοθηκών.

MPI
Εκτέλεση MPI εφαρμογών

- Οι εκδόσεις του MPI έχουν η κάθε μια ένα mpirun/mpiexec κλπ.
- Προτείνεται να χρησιμοποιείται το srun για την εκτέλεση παράλληλων εργασιών.
- Κάποιοι από τους λόγους
  - Το srun ξεκινάει τα εκτελέσιμα σε όλους τους κόμβους οπότε έχει πιο πλήρη έλεγχο.
  - Το srun κάνει accounting κατανάλωσης ρεύματος, χρήση Infiniband, χρήση δίσκων, κλπ.
  - Είναι κοινός τρόπος για τις (3 προς στιγμήν) εκδόσεις MPI που υπάρχουν στο ARIS

- Σε κάποιες περιπτώσεις, χρησιμοποιώντας mpiexec.hydra με IntelMPI έχουμε κάπως πιο γρήγορη εκτέλεση των εφαρμογών.
- Σε περιπτώσεις που η εφαρμογή έχει προβλήματα και χρειαστεί να σταματήσει ίσως να παρουσιαστούν προβλήματα (zombie procs) στη χρήση του scancel.
- Αν πάραυτα θέλετε να χρησιμοποιήσετε π.χ. mpirun, χρησιμοποιήστε το χωρίς τα συνήθη -np ,-machinefile κλπ.
- Συμβαίνει όταν χρησιμοποιούνται, να μην αλλάζει ταυτόχρονα ο αριθμός των tasks στο SLURM και ο αριθμός των tasks στο mpirun -np π.χ.

```
#SBATCH --nodes=10
#SBATCH --ntasks=200
mpirun -np 8
```

Δεσμεύετε (και χρεώνεστε) για 200 cores ενώ χρησιμοποιείτε μόλις 8.

- Η χρήση mvapich2 υποστηρίζεται (προς στιγμήν) MONO με srun.

Χρήση Βιβλιοθηκών

- Τι είναι οι βιβλιοθήκες ?
  - Συλλογή από ρουτίνες που κάνουν συγκεκριμένες εργασίες
- Και γιατί να τις χρησιμοποιήσω ?
  - Υπάρχουν διαθέσιμες
  - Συνήθως έχουν μεταφερθεί σε διάφορες αρχιτεκτονικές
  - Συνήθως είναι πολύ καλά ελεγμένες για ορθότητα αποτελεσμάτων
  - Είναι συνήθως βελτιστοποιημένες
  - Έχουν λιγότερα bugs

- Χρησιμοποιώντας βιβλιοθήκες κάνουμε τον κώδικά μας περισότερο μεταφέρσιμο. π.χ. μπορεί κάποιος να χρησιμοποιήσει GPUs ή Xeon Phi εάν η βιβλιοθήκη έχει μεταφερθεί σε GPUs/Xeon Phi.
- Μπορεί κάποιος να τις χρησιμοποιήσει μέσα από διάφορες γλώσσες προγραμματισμού π.χ. C, Fortran, Python

Κατηγορίες Βιβλιοθηκών

- Βιβλιοθήκες I/O
  - MPI-I/O, HDF5, NetCDF
- Αριθμητικές Βιβλιοθήκες
  - Γραμμική Άλγεβρα
  - Μετασχηματισμοί Fourier
  - ...
- Βιβλιοθήκες για συγκεκριμένα επιστημονικά πεδία
- Γραφικά

Βιβλιοθήκες στο ARIS

```
acml/5.3.1
atlas/3.11.34
boost/1.58.0
cgnslib/3.2.1/intel
elpa/2015.05.001/intel
fftw/2.1.5
fftw/3.3.4/avx
fftw/3.3.4/sse2
flame/5.0/gnu
flame/5.0/intel
glpk/4.55
grib_api/1.14.0
gsl/1.16/gnu
hdf5/1.8.12/gnu
hdf5/1.8.12/intel
hdf5/1.8.15/gnu
hdf5/1.8.15/intel
jasper/1.900.1
libint/1.1.5
libjpeg-turbo/1.4.1
libsmm/gnu
libsmm/intel
libxc/2.2.2
netcdf/3.6.3/intel
netcdf/4.1.3/gnu
netcdf/4.1.3/intel
netcdf-c/4.3.3.1/gnu
netcdf-c/4.3.3.1/intel
netcdf-combined/4.3.3.1/intel
netcdf-fortran/4.4.2/gnu
netcdf-fortran/4.4.2/intel
openblas/0.2.14/gnu/int4
openblas/0.2.14/gnu/int8
openblas/0.2.14/intel/int4
openblas/0.2.14/intel/int8
parmetis/4.0.3/gnu
parmetis/4.0.3/intel
pnetcdf/1.6.1/gnu
pnetcdf/1.6.1/intel
scalapack/2.0.2/gnu
scalapack/2.0.2/intel
szip/2.1
udunits2/2.2.19
voro++/0.4.6
```

Χρήση βιβλιοθηκών στο ARIS

- Όπως και κάθε πακέτο που είναι διαθέσιμο μέσω modules, τα modules των βιβλιοθηκών θέτουν μια μεταβλητή περιβάλλοντος που δηλώνει το PATH στο οποίο βρίσκονται εγκατεστημένες π.χ.

```
$ module show netcdf/4.1.3/gnu
-------------------------------------------------------------------
/apps/modulefiles/libraries/netcdf/4.1.3/gnu:

module-whatis  Enable usage for netcdf version 4.1.3, compiled with gnu 4.9.2
setenv  NETCDFROOT /apps/libraries/netcdf/4.1.3/gnu
setenv  NETCDF /apps/libraries/netcdf/4.1.3/gnu
prepend-path  PATH /apps/libraries/netcdf/4.1.3/gnu/bin
prepend-path  INCLUDE /apps/libraries/netcdf/4.1.3/gnu/include
prepend-path  LD_LIBRARY_PATH /apps/libraries/netcdf/4.1.3/gnu/lib
-------------------------------------------------------------------
```

- Δηλώνεται η μεταβλητή MODULENAMEROOT, στη συγκεκριμένη περίπτωση NETCDFROOT, ρυθμίζεται το PATH να περιλαμβάνει την $NETCDFROOT/bin, το LD_LIBRARY_PATH να περιλαμβάνει την $NETCDFROOT/lib.
- Όταν χρειάζεται να γίνει compile κάποια εφαρμογή που χρησιμοποιεί τη συγκεκριμένη version της netcdf, στα flags του compiler προστίθενται τα :

```
gcc -I$NETCDFROOT/include myfile.[c|f] -L$NETCDFROOT/lib -lnetcdff -lnetcdf
```

Χρήση βιβλιοθηκών στο ARIS

- Αρκετές βιβλιοθήκες, περιλαμβάνουν εργαλεία που μας δίνουν πρόσθετες πληροφορίες για το πως χρησιμοποιούνται. Στη συγκεκριμένη βιβλιοθήκη :

```
$ nc-config --cflags
-I/apps/libraries/netcdf/4.1.3/gnu/include -DpgiFortran
-I/apps/libraries/hdf5/1.8.12/gnu/include
$ nc-config --fflags
-O3 -mavx -march=ivybridge -I/apps/libraries/netcdf/4.1.3/gnu/include
$ nc-config --includedir
/apps/libraries/netcdf/4.1.3/gnu/include
$ nc-config --libs
-L/apps/libraries/netcdf/4.1.3/gnu/lib -O3 -mavx -march=ivybridge -lgpfs
-lnetcdf -lhdf5_hl -lm -lz -L/apps/libraries/hdf5/1.8.12/gnu/lib -lhdf5
$ nc-config --cc
mpicc
$ nc-config --cxx
mpicxx
```

- Στο compilation του WRF, το configure του δεν μπορεί να ανιχνεύσει την εξάρτηση από hdf5, MPI και libgpfs. Χρησιμοποιώντας το nc-config μπορούμε να έχουμε αυτή την extra πληροφορία ώστε να προστεθεί.

Εφαρμογές στο ARIS

- Κατηγορίες
  - Υπολογιστική Χημεία, Φυσική, Επιστήμη Υλικών, Βιολογία/Βιοιατρική
    - abinit, bigdft, cp2k, dl_poly, gromacs, lammps, mdynamix, mpqc, namd, nwchem, octopus, openmd, quantum-espresso, towhee gOpenMol, molden, molekel, openbabel, vmd
    - cpmd, gamessUS : Δεν είναι διαθέσιμα σε όλους (προς στιγμήν για την εκπαίδευση είναι) λόγω του ότι ο χρήστης πρέπει να έχει free academic license για να τα χρησιμοποιήσει.
  - Περιβάλλον
    - cosmo, wrf, cdo, ncarg, ncview
  - CFD

- Code_Saturn
- Γενικής Χρήσης
  - octave, R, paraview, qhull, gnuplot, grace, gimp

- Η λίστα δημιουργήθηκε από απαντήσεις εν δυνάμει χρηστών σε ερωτηματολόγια κατά τη διάρκεια υλοποίησης του έργου.
- Όλες οι εφαρμογές μεταγλωτίστηκαν με : IntelMPI/5.0.3. Οι περισότερες με Intel 15.0.3 και κάποιες με GNU/4.9.2

Πολλαπλά εκτελέσιμα της ίδιας εφαρμογής/version

- Αρκετές εφαρμογές μπορούν μεταγλωτιστούν είτε με MPI μόνο είτε Υβριδικά MPI/OpenMP.
- Σε αρκετές εφαρμογές, η εκτέλεση της υβριδικής version με 1 thread/task δεν είναι ισοδύναμη με την καθαρά MPI version π.χ. namd.
- Στις περιπτώσεις όπου η διαδικασία μεταγλώτισης επιτρέπει τη χρήση suffix, τα εκτελέσιμα βρίσκονται στο ίδιο PATH με διαφορετικό suffix. π.χ. Gromacs/5.1

```
ls $GROMACSROOT/bin/gmx*
gmx        Single Precision, Multithread, no MPI
gmx_d      Double Precision, Multithread, no MPI
gmx_mpi    Single Precision, Hybrid MPI/OpenMP
gmx_d_mpi Single Precision, Hybrid MPI/OpenMP
```

- Στις περιπτώσεις η μεταγλώτιση παράγει πάντα τα ίδια ονόματα εκτελέσιμων, η εγκατάσταση έχει γίνει σε διαφορετικό PATH και ενεργοποιείται με διαφορετικό module, π.χ.

```
wrf/3.7/hybrid
wrf/3.7/purempi
```

- Οι περισότερες εφαρμογές στο ARIS είναι εκτελούνται πιο γρήγορα με MPI μόνο. Μετρήσεις είναι απαραίτητες: Η ταχύτητα και κλιμάκωση μιας εφαρμογής εξαρτάται σε πολύ μεγάλο βαθμό από το Input της.

- Στις περισότερες εφαρμογές που η υβριδική version είναι ισοδύναμη με την MPI μόνο κάνοντας τις απαραίτητες ενέργειες, υπάρχει και η υβριδική και η MPI μόνο version.
  Ο κύριος λόγος είναι ότι είτε η εφαρμογή χρειάζεται επιπλέον ορίσματα για να χρησιμοποιήσει 1 thread/task ή και δεν συμπεριφέρεται σωστά, είτε οι χρήστες δεν προσέχουν τις λεπτομέρειες για τα threads/task.
  Δεν είναι σπάνιο, σε συστήματα με 20 cores/node όπως το ARIS, να εμφανίζεται load 400 στα nodes (20 tasks * 20 threads/task).

Scripts Εκτέλεσης Εφαρμογών στο ARIS

- Αρκετές εφαρμογές χρειάζονται σχετικά πολύπλοκα scripts για να εκτελεστούν.
- Σε όσες έχει ταυτοποιηθεί η ανάγκη για scripts, αυτά έχουν φτιαχεί και εγκατασταθεί στο PATH.
- nwchem/6.5/bin/runnw

```
#!/bin/bash
export I_MPI_FABRICS=shm:dapl
if [ $# -lt 1 ]; then
  echo ”Usage : $0 input_file ”
  exit;
fi
EXE=nwchem
JOBNAME=‘/bin/basename $1 .nw‘
if [ -e $JOBNAME.nw ] ; then
SCRATCH=/work/scratch/$SLURM_JOB_ID
INDIR=‘pwd‘
if [ -d $SCRATCH ]; then
echo directory exists. continuing ....
```

```
else
mkdir -p $SCRATCH
fi
export NWCHEM_BASIS_LIBRARY=$NWCHEMROOT/data/libraries/
cp $JOBNAME.nw $SCRATCH
cd $SCRATCH
echo ”OUTPUT of the program is in file $INDIR/$JOBNAME.output”
echo ”Temporary Files are in directory $SCRATCH”
echo ”Starting Parallel execution...”
srun $EXE $JOBNAME.nw > $INDIR/$JOBNAME.out
echo ”Ececution finished.”
cd $INDIR
rm   -f $SCRATCH/$JOBNAME.nw
tempfiles=‘ls $SCRATCH‘
rm -rf $SCRATCH
else
echo ”$JOBNAME.nw doesnot exist in $INDIR”
exit
fi
```

Άν το input file λέγεται testinput.nw, στο SLURM script μετά από τα Job specifications χρειάζεται μόνο : runnw testinput. Το output θα βρίσκεται στο testinput.out.

- GAMESS-US. Χρησιμοποιεί Global Arrays => Κάθε compute process χρειάζεται και ένα data server process το οποίο αν και δεν καταναλώνει σημαντικό ποσοστό του χρόνου, πρέπει να εκτελείται => με π.χ. 20 tasks υπάρχει η ανάγκη να εκτελούνται 20 compute tasks και 20 dataserver tasks = 40 MPI tasks. Λεπτομέρειες από terminal λόγω μεγέθους.

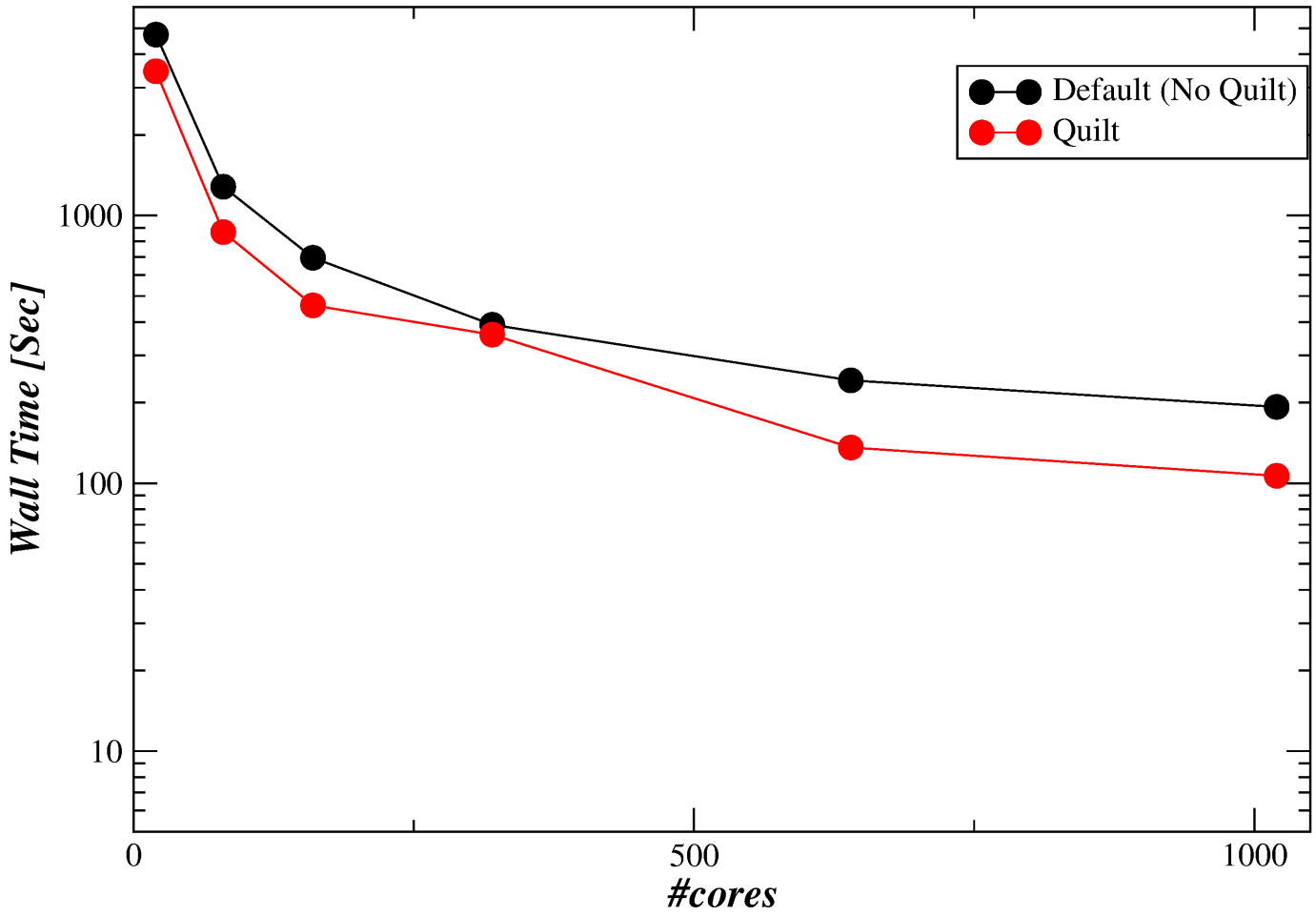
Ιδιαιτερότητες εφαρμογών : WRF

- Τι κάνει το WRF ?
- (Από το description του WRF) : The Weather Research and Forecasting (WRF) Model is a next-generation mesoscale numerical weather prediction system designed to serve both operational forecasting and atmospheric research needs. => Weather Forecast, modeling κλπ.
- Ιδιαιτερότητες :

- Τουλάχιστον μέχρι version 3.4 έχει hardcoded μέγιστο όριο 1024 tasks. Κάποιος μπορεί να χρησιμοποιήσει την υβριδική έκδοση (διαθέσιμη στο ARIS) ώστε να ξεπεράσει το όριο αυτό. Λόγω του 20 cores/node, το πλέον λογικό είναι είτε 50 ή 51 nodes (1000 ή 1020 tasks), που με π.χ. 2 threads/task => 2000/2020 cores με 4 threads/task => 4000/4080 cores κλπ.
- Quilting : Έχει μια παράμετρο στο namelist.input που του λέει, να χρησιμοποιήσει κάποιο αριθμό cores / node για το I/O και τα υπόλοιπα να κάνουν μόνο computation.
- Συνήθως η καλύτερη επιλογή είναι 1 task/node, εξαρτάται και από διάφορες άλλες παραμέτρους ”μεγέθους” που επιλύει.
- Ε, και ?

Ιδιαιτερότητες εφαρμογών : WRF

- Performance/Scaling of WRF 3.4 With/Without Quilting (Στο ARIS). Σημειώστε το λογαριθμικό της κλίμακας στό επόμενο σχήμα



Ιδιαιτερότητες εφαρμογών : WRF

Και σε αριθμούς :

24ωρη πρόγνωση για Ελλάδα.

#Cores	Time [sec]		Core Hours	
	No Quilt	Quilt	No Quilt	Quilt
20	4733.36	3455.81	26.30	19.20
80	1280.50	866.51	28.46	19.26
160	695.09	461.49	30.89	20.51
320	390.76	358.91	34.73	31.90
640	242.12	136.02	43.04	24.18
1020	192.81	106.58	54.63	30.20

- Χρησιμοποιώντας την τεχνική έχω το αποτέλεσμα που θέλω εντός 136 sec, με κόστος 24.18 core hours αντί 192 sec με κόστος 54 core hours.

Ιδιαιτερότητες εφαρμογών

- Quantum Mechanics : Scratch vs Direct
- 2/3D partitioning (cores - problem geometry)
- Problem Size - cores