Available online at www.prace-ri.eu

**Partnership for Advanced Computing in Europe**

# Scaling and Performance Improvements in Elmer/Ice

Thomas Zwinger,[a]* Mika Malinen,[a] Juha Ruokolainen,[a] Peter Råback [a]

[a]*CSC - IT Center for Science, P.O. Box 405, FI-02101 Espoo,Finland*

**Abstract**

By gaining and losing mass, glaciers and ice-sheets play a key role in sea level evolution. This is obvious when considering the past 20000 years, during which the collapse of the large northern hemisphere ice-sheets after the Last Glacial Maximum contributed to a 120m rise in sea level. This is particularly worrying when the future is considered. Indeed, recent observations clearly indicate that important changes in the velocity structure of both the Antarctic and Greenland ice-sheets are occurring, suggesting that large and irreversible changes may already have been initiated. This was clearly emphasised in the last report published by the Intergovernmental Panel on Climate Change (IPCC) [7]. The IPCC also asserted that current knowledge of key processes causing the observed accelerations was poor, and concluded that reliable projections obtained with process-based models for sea-level rise (SLR) are currently unavailable. Most of these uncertain key processes have in common that their physical/numerical characteristics, such as shallow ice approximation (SIA), are not accordingly reflected or even completely missing in the established simplified models that have been in use since decades. Whereas those simplified models run on common PC systems, the new approaches require higher resolution and larger computational models, which demand High Performance Computing (HPC) methods to be applied. In other words, numerical glaciology, like climatology and oceanography decades ago, needs to be updated for HPC with scalable codes, in order to deliver the prognostic simulations demanded by the IPCC. The DECI project ElmerIce, and enabling work associated with it, improved simulations of key processes that lead to continental ice loss. The project also developed new data assimilation methods. This was intended to decrease the degree of uncertainty affecting future SLR scenarios and consequently contribute to on-going international debates surrounding coastal adaptation and sea-defence planning. These results directly feed into existing projects, such as the European FP7 project ice2sea [9], which has the objective of improving projections of the contribution of continental ice to future sea-level rise and the French ANR ADAGe project [10], coordinated by O. Gagliardini, which has the objective to develop data assimilation methods dedicated to ice flow studies. Results from these projects will directly impact the upcoming IPCC assessment report (AR5).

## Introduction

Using the finite element code Elmer/Ice, the group at Laboratoire de Glaciologie et Géophysique de l'Environnement (LGGE) in Grenoble, France, performed transient simulations of ice flow for different glaciers/ice-sheets. These applications were run using a relatively small number of partitions (< 64) due to a limitation induced by the direct solver MUMPS (a MUltifrontal Massively Parallel sparse direct Solver) to solve for the Stokes system. However, larger applications or sensitivity experiments require improving the parallel performance of glaciological application problems using Elmer/Ice. Recently, a block preconditioned method to solve the from the linear system of the Stokes equations ill-conditioned matrix system has been developed in cooperation with CSC and Uppsala University. During this project this new precondition technique was implemented within a real glaciological application and tested for its scalability on the PDC machine Lindgren. This new solver developed by CSC within the DECI project ElmerIce clearly allows an increased number of partitions, enabling solutions for larger problems that utilize modern computer clusters or supercomputers. In

addition, the allocated CPU hours were used to perform simulations of grounding line dynamics and demonstrated that grounding lines are not unconditionally unstable on reverse slope when considering two horizontal dimensions.

Elmer (`www.csc.fi/elmer`) and its glaciological addition, Elmer/Ice (`http://elmerice.elmerfem.org`), were ported to the XE6 system, Lindgren, at PDC. Certain issues with respect to the dynamically linked structure of Elmer and the dependency on external libraries, such as HYPRE, MUMPS and Trilinos, had to be addressed.

The tests performed in the framework of this project on the Greenland geometry indicate an excellent scalability with this new solver up to thousands of partitions/cores. On the other hand, a large part of the CPU hours allocated in the framework of this project was used to perform the grounding line simulations presented in the paper [6]. As an important result for the glaciological community, it is found that – contrary to the recent common understanding based on one-dimensional model findings – grounding lines are not unconditionally unstable on reverse slope if considering the two horizontal dimensions.

# Technical issues

The first task was to compile Elmer and consequently Elmer/Ice on Lindgren. Based on previous experience on CSC's earlier flagship computer, the XT4/5 system louhi, the necessary libraries for Elmer were either available as system libraries or compiled on Lindgren's local klemming-file system. Support was provided by people at PDC (the site of the machine). The following major steps were necessary to obtain reasonable performance for Elmer:

- Switching the default programming environment to GNU with the command: `module swap PrgEnv-pgi/3.1.61 PrgEnv-gnu`, since Elmer proofed to compile best with the GNU set of compilers (mainly due to gfortran).
- Enabling shared objects, setting the environment variable `XTPE_LINK_TYPE=dynamic`. Elmer is built on the concept of dynamically shared objects. For instance, all user functions (and there are a lot of them in glaciological simulations) are loaded at runtime. A static build of Elmer would basically make it impossible for the end user to work with it.
- Identifying the correct external libraries to be linked. Basic Elmer needs LAPACK and BLAS, which by default are provided by the Elmer source code (from `http://www.netlib.org`), but of course not optimized for the Cray. Hence, the version provided by the installed ACML package were used, using a trick to pass the `--blas="-lm" --lapack="-lm"` option flags to the autoconfig process of Elmer. Introducing the system BLAS instead of the one provided by NetLib (and compiled using standard optimization) improved performance by a factor 20!
- Building MUMPS using own version of ScaLAPACK, which in turn also was linked to Cray's libsci.
- Building HYPRE using correct settings for BLAS and LAPACK, like mentioned earlier.
- Identifying the correct linking options for Trilinos (ML was used for solving single blocks of the matrix within the block-preconditioning procedure).

A further issue was encountered because the wall-clock time for a single run on Lindgren was limited to a maximum of 12 hours. Based on tests done on CSC's XT4/5 system louhi, a PBS script was developed that automatically identified the job number of the previous run and then inserted the necessary information to restart from the results and continue the computation.

The compilation script as well as the batch job script for the restarted runs is in the appendices at the end of this paper.

# Block pre-conditioner

The linear systems that need to be solved in Elmer/Ice are of the form:

$$\begin{pmatrix} A & B^T \\ B & C \end{pmatrix} \begin{pmatrix} v \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1}$$

where the v-component of the solution vector represents the velocities, and the p-component the pressure. The matrix A is an elasticity-like operator, C a stabilization term, $B^T$ the negative gradient and B the negative divergence operator. The right hand side contains contributions from gravitational forces and boundary conditions.

While direct solvers are a robust way of solving (1), their computational cost in 3D grows with the square of the number of unknowns, and their memory usage becomes excessive for large 3D applications. Iterative methods are therefore needed.

The general strategy for the iterative solution of is to use a Krylov-subspace method (called the "outer iteration"), combined with a preconditioner P. The approach followed in the newly written module is to use GCR (the generalized conjugate residual method) as outer iteration, and a preconditioner of the block upper triangular form:

$$P = \begin{pmatrix} A & B^T \\ 0 & M \end{pmatrix} \tag{2}$$

Here M denotes the mass matrix, scaled by the element-wise fluidity of the current approximation of the solution. It can be shown that, under certain regularity assumptions, this preconditioner leads to a convergence rate of the outer iteration which is independent of the mesh-size. However, the application of the inverse of P still requires the exact solution of linear systems with A and M. These operations would still be too expensive with direct solvers for large 3D problems, so further approximations are made. Note that the inverse of P has to be applied in each outer iteration, so the approximate solution of systems with A and M should be very inexpensive compared to the exact solution of a system (1).

# Results

In the following we present the findings of the numerical experiments conducted within this application. For the purpose of investigating the scalability of the newly implemented block-preconditioner, tests with different mesh resolutions and partition numbers were undertaken, using the present-date Greenland geometry, for which corresponding solutions obtained with direct solvers existed. The second focus of activities was on a synthetic case investigating the assumption that the grounding line (contact line with the ocean where ice sheet gets afloat) is unconditionally unstable on retrograde slopes in three-dimensions.

## New block preconditioned solver

In collaboration with Jonas Thies (Uppsala University, Sweden), who also presented this method in a course held by the PRACE Advances Training Center [11] in Helsinki, a new solver for the Stokes system has been implemented in Elmer/Ice and tested by performing scalability tests. This solver was inspired by previous works on preconditioned solvers for variable viscosity Stokes flows [1],[3],[5],[8], but includes improvements like carefully considered application of linear system scaling and adaptable ways to improve finite element stability via utilizing high-order bubble polynomials.

Weak (constant load per CPU, Fig. 1) and strong (constant problem size, Fig. 2) scalability tests were performed using the present-date Greenland geometry. Meshes from 708,000 up to 4,580,000 nodes were used to test the scalability of the newly implemented block preconditioned method to solve the Stokes system. The performance of this new solver has been compared with the parallel sparse direct solver MUMPS. The weak scaling experiment uses a constant number of 4200 nodes per partition in combination with an increasing number of partitions from 168 up to 1092. Weak scaling was found to be above 60% even for the largest test case, from the baseline of 168 partitions. An ideal efficiency was obtained with the new block preconditioned method for the strong scaling, whereas for a number of partitions larger than 100, MUMPS was always found to scale poorly. This new solution strategy clearly opens the door to applications an order of magnitude larger than could be performed with the original Elmer/Ice.

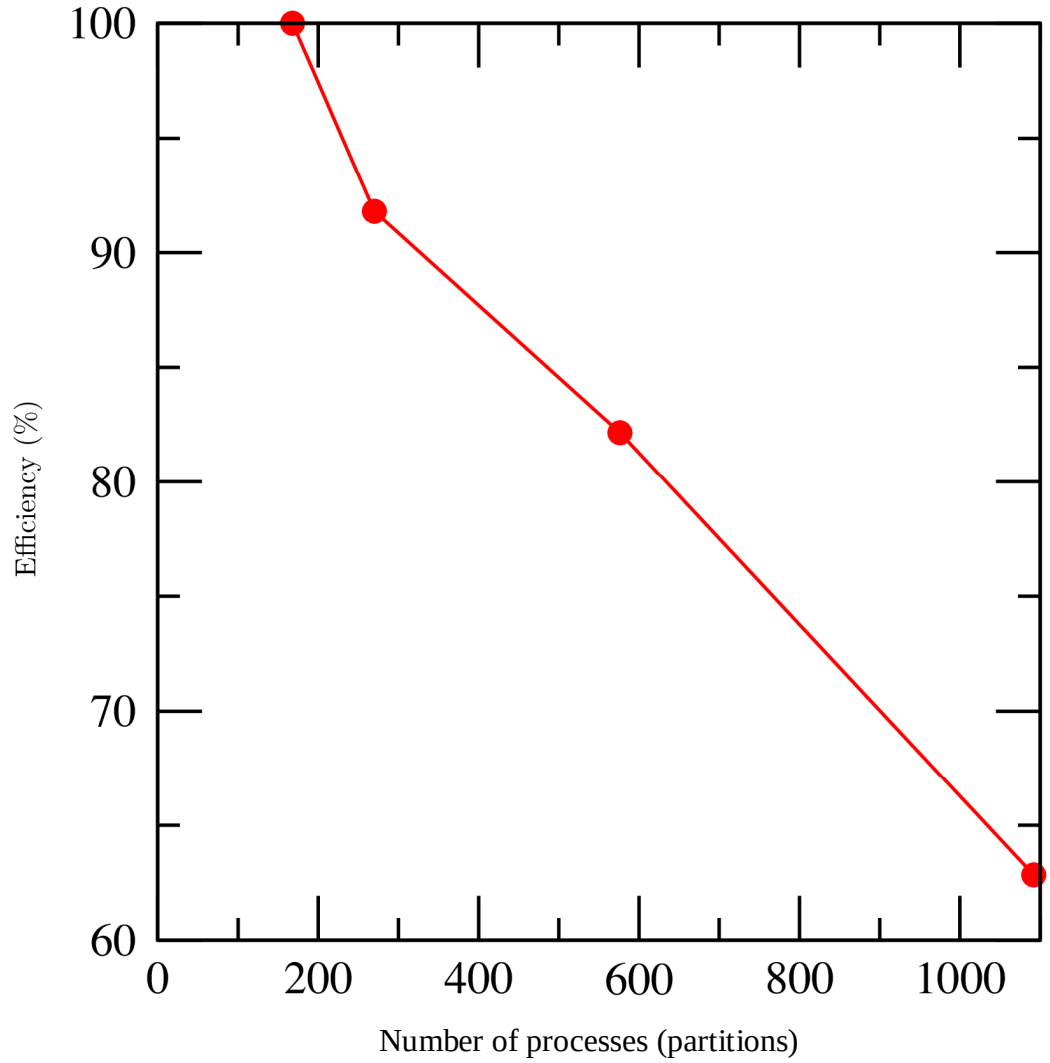


Fig. 1. Efficiency for a weak scaling experiment using the new block preconditioned solver for an approximate number of 4,200 nodes in all partitions and meshes from 0.708x106 nodes up to 4.58x106 nodes.

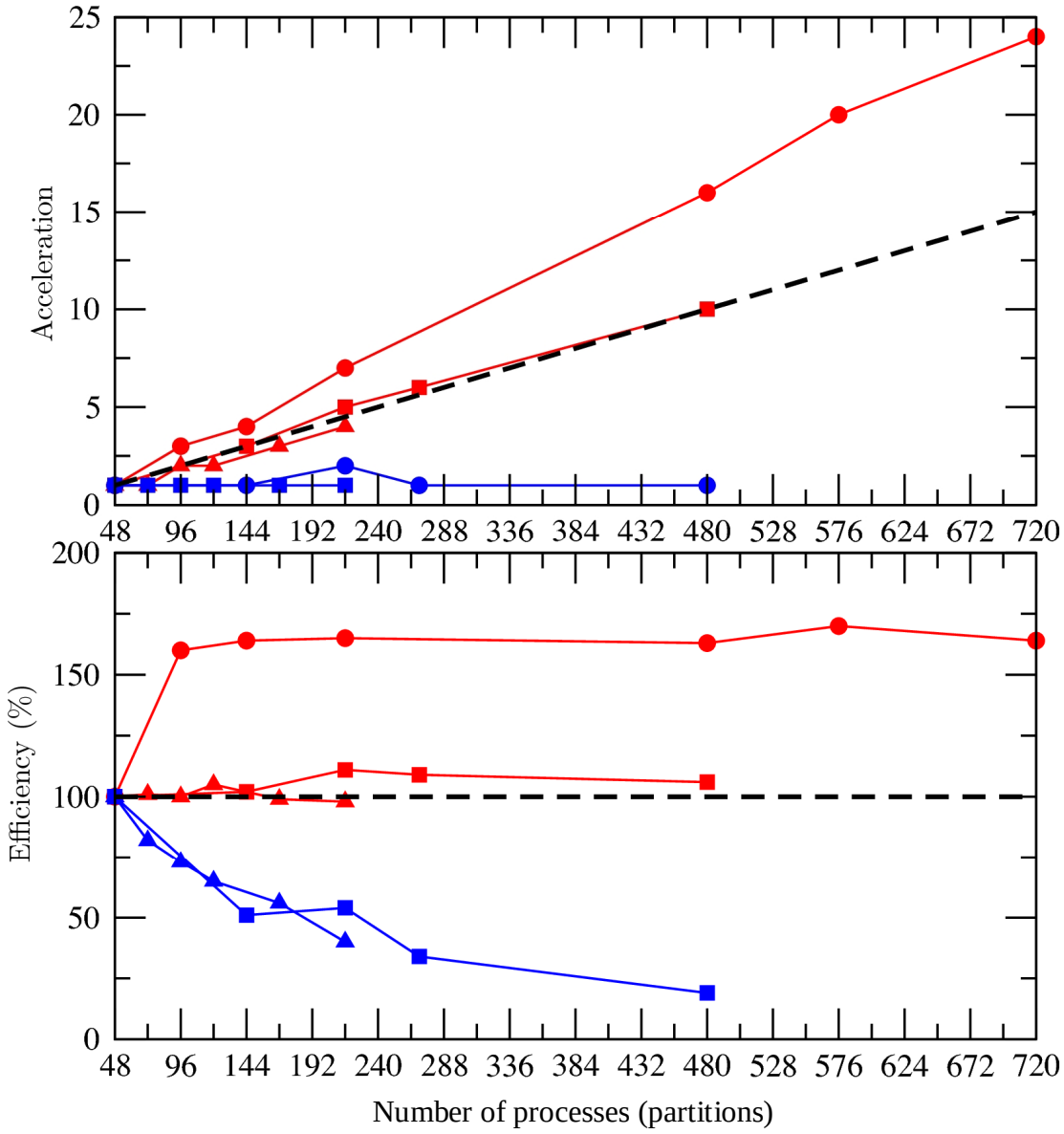


Fig. 2. Speedup (up) and efficiency (bottom) for strong scaling experiments using the new block preconditioned solver for meshes with 2.400x106 nodes (red bullets), 1.142x106 nodes (red squares) and 0.708x106 nodes (red triangles), and the MUMPS solver for meshes with 1.142x106 nodes (blue squares) and 0.708x106 nodes (blue triangles). The dashed line indicates a theoretical efficiency of 100%.

## Stability of grounding line on reverse slope

In [6], the stability of the grounding line (the line defining the border between grounded ice and emerged floating ice in water) on reverse slopes is questioned for three-dimensional geometries (see Fig. 3). As an important result, it was found that the grounding line is not unconditionally unstable on reverse slopes when considering the two horizontal dimensions. Retrograde bed slopes at the grounding lines of marine ice sheets, such as the West Antarctic Ice Sheet (WAIS), do not per se imply instability, nor do they imply that these regions are close to a threshold of instability. This result clearly questions those estimates of the potential near-future contribution of WAIS to global sea level change based solely on the assumption that WAIS, with the major part of the grounding line resting on retrograde slopes, must be inherently unstable.

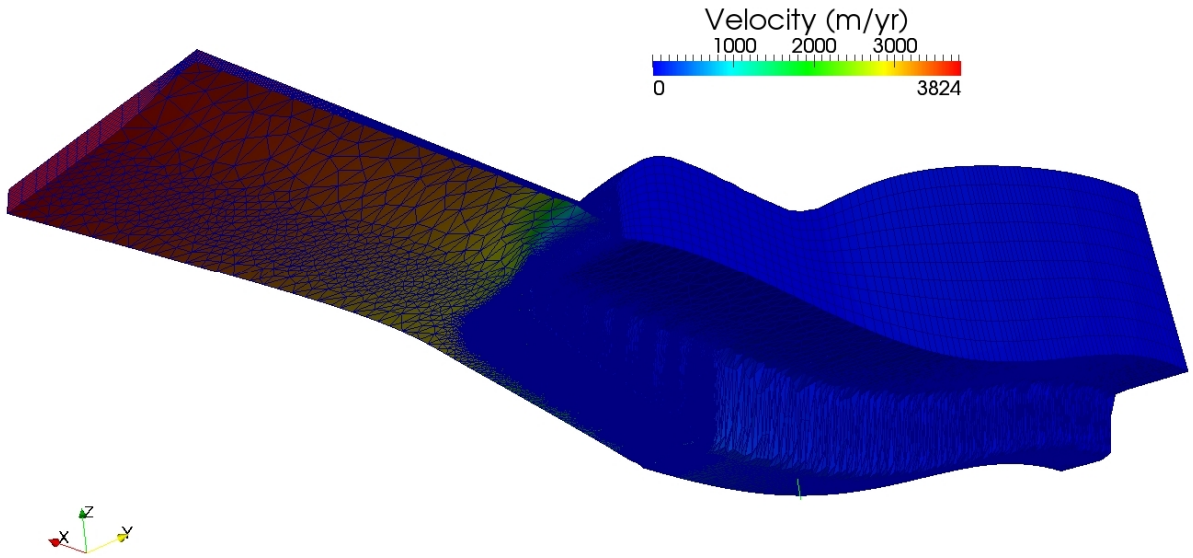


Fig. 3. A perspective plot of the steady-state geometry obtained with the numerical code Elmer/Ice. Colour scale indicates the ice flow velocity and the mesh refinement is shown.

# Use and benefits of the PRACE infrastructure

The newly developed block preconditioned solver will clearly benefit the whole community by enabling solutions for large Stokes problems. Primarily, it will benefit the Elmer community, but this might also be implemented relatively quickly with other ice sheet models that solve the full-Stokes system, and then benefit the whole glaciological community. It might also benefit other communities solving flow problems on materials with non-linear viscosity.

The result presented in [6] will clearly have a broad impact in the sea level community as a large number of sea level rise predictions were based upon the assumption of marine ice-sheet instability. In this paper the authors demonstrate that this is not unconditionally true.

# Future work involving PRACE

A paper currently submitted and under review will focus on the “Capabilities and performances of the new generation ice-sheet model Elmer/Ice” and presents results obtained with this new block preconditioned method [2]. This paper was submitted at the beginning of 2013 to Geoscientific Model Development, an interactive open access journal of the European Geosciences Union.

Currently, a new setup for the whole Antarctic ice sheet is being tested. With the developed block-preconditioner method, we hope to soon be able to present a high resolution inversion of the bedrock sliding coefficients for the largest ice mass on the planet, similar to the work that has been done on Greenland [4].

**References**

[1] C. Burstedde et al., “Parallel scalable adjoint-based adaptive solution of variable-viscosity Stokes flow problems,” Comput. Method. Appl. M. **198**, 1691 (2009); doi:10.1016/j.cma.2008.12.015

[2] O. Gagliardini, T. Zwinger, and 12 others, “Capabilities and performances of the new generation ice-sheet model Elmer/Ice,” submitted to Geosci. Model Dev. (2013)

[3] T. Geenen et al., “Scalable robust solvers for unstructured FE geodynamic modeling applications: Solving the Stokes equation for models with large localized viscosity contrasts,” Geochem. Geophy. Geosy. **10** (2009); doi:10.1029/2009GC002526
[4] F. Gillet-Chaulet, O. Gagliardini, H. Seddik, M. Nodet, G. Durand, C. Ritz, T. Zwinger, R. Greve and D.G. Vaughan, “Greenland ice sheet contribution to sea-level rise from a new-generation ice-sheet model,” The Cryosphere, **6**, 1561 (2012); doi:10.5194/tc-6-1561-2012
[5] P.P. Grinevich and M.A. Olshanskii, “An iterative method for the Stokes-type problem with variable viscosity,” SIAM J. Sci. Comput. **31**, 3959 (2009); doi:10.1137/08744803
[6] G.H. Gudmundsson, J. Krug, G. Durand, L. Favier, and O. Gagliardini, “The stability of grounding lines on retrograde slopes,” The Cryosphere, **6**, 1497 (2012); doi:10.5194/tc-6-1497-2012
[7] S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor, and H.L. Miller (eds.) “Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change” (Cambridge University Press, Cambridge/New York, 2007)
[8] M. ur Rehman et al., “On iterative methods for the incompressible Stokes problem,” Int. J. Numer. Meth. Fl. **65**, 1180 (2011); doi:10.1002/fld.2235
[9] www.ice2sea.eu
[10] www-lgge.obs.ujf-grenoble.fr/pdr/ADAGe
[11] events.prace-ri.eu/conferenceDisplay.py?confId=29

**Acknowledgements**

We thank Nicole Audiffren (CINES, France) for the coordination within DECI, Gaël Durand, Olivier Gagliardini, Fabien Gillet-Chaulet, Lionel Favier, Basile de Fleurian and Jean Krug (LGGE, France) for the work on the scientific applications and Jonas Thies (Uppsala University, Sweden) for cooperation in developing the block preconditioned method as well as the interface of Elmer to Trilinos. We also want to express our gratitude to the support from PDC, in particular Lilit Axner and Jonathan Vincent as well as Petri Nikunen (CSC, Finland) for reading and David Silverstein (PDC, Sweden) for reviewing this paper and making helpful comments. This work was supported through the Distributed European Computing Initiative (DECI) of the PRACE-2IP project funded in part by the European Commission Framework Programme (FP7) under grant agreement RI-283493.

# Appendices

## Elmer compilation script

```
#!/bin/bash
#####################################################################
# Compilation script for Elmer on PDC Cray XE6 Lindgren
#
#####################################################################
umask 0002
     cd trunk

# compiler wrappers
      export CC=cc
      export CXX=CC
      export FC=ftn
      export F77=ftn

      export COMP_DEFS="-D__x86_64__ -D__CRAYXE -D__CRAYXT_COMPUTE_LINUX_TARGET -
D__TARGET_LINUX__"

      # the performance flags
               export OPTFLAGS="-fPIC -O3 -msse2 -ftree-vectorize -funroll-loops\
            -ffast-math -march=opteron -mtune=opteron"
      export CFLAGS="$OPTFLAGS $CFLAGS"
      export CXXFLAGS="$OPTFLAGS $CXXFLAGS"
      export FFLAGS="$OPTFLAGS $FFLAGS"
      export FCFLAGS="$OPTFLAGS $FCFLAGS"


      # All the nice little paths and files needed:
      export UGNI_DIR="/opt/cray/ugni/2.1-1.0301.2967.10.23.gem/lib64"
      export UDREG_DIR="/opt/cray/udreg/2.2-1.0301.2966.16.2.gem/lib64"
```

```
      export DMAPP_DIR="/opt/cray/dmapp/3.0-1.0301.2968.22.24.gem/lib64"
      export XPMEM_DIR="/opt/cray/xpmem/0.1-2.0301.25333.20.2.gem/lib64"
      export PMI_DIR="/opt/cray/pmi/2.1.2-1.0000.8396.13.5.gem/lib64"
      export MPI_LIBDIR="/opt/cray/mpt/5.3.0/xt/gemini/mpich2-gnu/lib/"
      export ALPS_DIR="/usr/lib/alps/"
      export GCC_DIR="/opt/gcc/4.5.3/snos/lib64"
      export MPI="/opt/cray/mpt/5.3.0/xt/gemini/mpich2-gnu/"
      export MPI_INC="-I/opt/cray/mpt/5.3.0/xt/gemini/mpich2-gnu/include/45\
             -I/opt/cray/mpt/5.3.0/xt/gemini/mpich2-gnu/include -I/usr/include/alps"
      export MPI_LIB="$MPI_LIBDIR/libmpichf90.a $MPI_LIBDIR/libmpich.so $MPI_LIBDIR/libmpl.a\
             $XPMEM_DIR/libxpmem.a $DMAPP_DIR/libdmapp.a $UGNI_DIR/libugni.a
$PMI_DIR/libpmi.a\
             $UDREG_DIR/libudreg.a  $ALPS_DIR/libalpslli.a $ALPS_DIR/libalpsutil.a  -lpthread"

      # linked-in libraries
      export OWN_LIBS="/path/to/self/compiled/libraries"
      #HYPRE
      export HYPRE="$OWN_LIBS"
      #MUMPS
      export MUMPS_DIR="$OWN_LIBS"
      export FCFLAGS="$FCFLAGS -I$MUMPS_DIR/include"
      export FCPPFLAGS="$FCPPFLAGS -I$MUMPS_DIR/include -DHAVE_MUMPS"
      export SCALAPACK="-L$OWN_LIBS/lib  -lscalapack"
      export LDFLAGS="-L$MUMPS_DIR/lib -ldmumps -lmumps_common -lpord -lpthread $SCALAPACK"
      # Trilinos (from Cray)
      export CXXFLAGS="\$CXXFLAGS -DHAVE\_TRILINOS \
          -I/opt/cray/trilinos/default/GNU/47/x86_64/include"
      export FCPPFLAGS="\$FCPPFLAGS -DHAVE\_TRILINOS"
      export LDFLAGS="$LDFLAGS -L/opt/cray/trilinos/default/GNU/47/x86_64/lib\
            -lbelostpetra_gnu -lbelosepetra_gnu -lbelos_gnu -lml_gnu -lifpack_gnu\
            -lamesos_gnu -lgaleri_gnu -lisorropia_gnu -lepetraext_gnu\
            -ltpetrainout_gnu -ltpetra_gnu -ltriutils_gnu -lzoltan_gnu -lepetra_gnu\
            -lkokkoslinalg_gnu -lkokkosnodeapi_gnu -lkokkos_gnu -lteuchos_gnu"
      # the modules to be installed
      modules="matc umfpack mathlibs elmergrid meshgen2d eio hutiter fem"

      # well, then, fingers crossed:
      for m in $modules; do
        cd $m
          ./configure --prefix="/path/to/elmer/installtion"\
             --with-mpi=yes --with-hypre="-I$HYPRE/include -L$HYPRE/lib -lHYPRE"\
             --with-blas="-lm" --with-lapack="-lm"
        make -j6
        make install
        cd ..
      done
      cd ..
echo "All Done"
```

## PBS runscript

```
#!/bin/bash
basename="run"
for count in {25..27}
do
  (( prev = count - 1 ))  # sets the previous file to restart
  ############## the Elmer SIF part
  echo "\$namerun = \"${basename}${count}\""  > ${basename}${count}.sif
  echo "\$namerunRS = \"${basename}${prev}\""    >> ${basename}${count}.sif
  cat "./main.sif" >> ${basename}${count}.sif
  echo ${basename}${count}.sif > ELMERSOLVER_STARTINFO${count}

  ############## Create submit file
  echo "#PBS -N t_${basename}_${count}" > ${basename}_${count}.pbs
  echo "#PBS -o t_${basename}_${count}.log" >> ${basename}_${count}.pbs
  echo "#PBS -j oe" >> ${basename}_${count}.pbs
  echo "#PBS -l mppwidth=24" >> ${basename}_${count}.pbs
  echo "#PBS -l walltime=24:00:00" >> ${basename}_${count}.pbs
  if [ $count -gt 0 ];
  then
      echo "#PBS -W depend=afterany:'$jobid'" >> ${basename}_${count}.pbs
  fi
  echo "cat \$PBS_NODEFILE" >> ${basename}_${count}.pbs
  echo "cd \$PBS_O_WORKDIR" >> ${basename}_${count}.pbs
  echo "echo \$PBS_JOBID" >> ${basename}_${count}.pbs
  echo ". /opt/modules/default/etc/modules.sh" >> ${basename}_${count}.pbs
```

```
  echo "module swap PrgEnv-pgi/3.1.61 PrgEnv-gnu" >> ${basename}_${count}.pbs
  echo "cp ELMERSOLVER_STARTINFO${count} ELMERSOLVER_STARTINFO" >> ${basename}_${count}.pbs
  echo "aprun -n 24 ElmerSolver_mpi > elmertest${count}.out" >> ${basename}_${count}.pbs

  ############## submit the job
  jobid=`qsub ${basename}_${count}.pbs`
  echo ${jobid}
done
```