Petsc on HPC

What is Petsc?

PETSc (Portable, Extensible Toolkit for Scientific Computation) is a software library for numerical calculations and scientific computing developed at the University of Tennessee and Oak Ridge National Laboratory. It is an open-source software package that provides a powerful and flexible platform for solving large-scale scientific problems.

PETSc is written in C, but it includes interfaces for several other programming languages, including C++, Fortran, and Python. The library provides a variety of tools for solving linear and nonlinear equations, optimization problems, and eigenvalue problems. It also includes support for parallel computing, allowing users to run computations on clusters or supercomputers.

One of the key features of PETSc is its flexibility. The library provides a wide range of algorithms and solvers that can be customized and combined in various ways to suit the needs of different applications. PETSc is also designed to be portable and easily adaptable to different computing architectures and operating systems.

PETSc has been used in a wide range of scientific and engineering applications, including fluid dynamics, electromagnetics, structural analysis, and quantum mechanics. Its user base includes researchers in academia, government, and industry, as well as students and hobbyists interested in scientific computing.

Links:

Official Website

Manual

Versions Available:

The following versions are available on the cluster:

- `petsc/3.11.2`

- `petsc/3.3-p7`
- `petsc/3.3-p7-gcc-mpich`
- `petsc/3.3-p7-intel-mpich`

How to load Petsc?

To load Petsc, use the following commands:

```
#Load the Petsc module
module load petsc/3.11.2
```

To verify if the module is loaded correctly, use the following command,

```
# List all the module loaded in the environment
module list
```

How to use Petsc?

Users can link their application code with the PETSc libraries using compiler flags such as -I and -L. Users can then call PETSc functions from their code to solve numerical problems related to scientific computing. To use PETSc in parallel, users need to initialize MPI and set up the appropriate data structures and communication patterns. PETSc provides several tools for parallel computing, such as parallel vectors and matrices, distributed arrays, and parallel preconditioners. Users can choose from a variety of parallel options and configure the library to suit their needs. PETSc also provides diagnostic tools for debugging and profiling parallel applications, such as PetscLog and PetscViewer.

Here is a basic demonstration of sample code in C++ that uses the library and compile flag.

```
 #include <petscksp.h>
#include <mpi.h>

int main(int argc, char **argv) {
    // Initialize MPI
    MPI_Init(&argc, &argv);

    // Initialize PETSc
    PetscInitialize(&argc, &argv, PETSC_NULL, PETSC_NULL);

    // Set the size of the linear system
    const PetscInt n = 10;

    // Create a linear system matrix
    Mat A;
    MatCreate(PETSC_COMM_WORLD, &A);
    MatSetSizes(A, PETSC_DECIDE, PETSC_DECIDE, n, n);
    MatSetFromOptions(A);
    MatSetUp(A);

    // Create a right-hand side vector
    Vec b;
    VecCreate(PETSC_COMM_WORLD, &b);
    VecSetSizes(b, PETSC_DECIDE, n);
    VecSetFromOptions(b);

    // Create a solution vector
    Vec x;
    VecDuplicate(b, &x);

    // Fill the matrix and vectors with some values
    PetscInt i_start, i_end;
    MatGetOwnershipRange(A, &i_start, &i_end);
    for (PetscInt i = i_start; i < i_end; ++i) {
        for (PetscInt j = 0; j < n; ++j) {
            MatSetValue(A, i, j, 1.0 / (i + j + 1), INSERT_VALUES);
        }
        VecSetValue(b, i, 1.0, INSERT_VALUES);
    }
    MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY);
    MatAssemblyEnd(A, MAT_FINAL_ASSEMBLY);
    VecAssemblyBegin(b);
    VecAssemblyEnd(b);

    // Create the linear solver
    KSP solver;
    KSPCreate(PETSC_COMM_WORLD, &solver);
    KSPSetOperators(solver, A, A, SAME_NONZERO_PATTERN);
    KSPSetType(solver, KSPCG);
    KSPSetFromOptions(solver);

    // Solve the linear system
    KSPSolve(solver, b, x);

    // Print the solution
    PetscPrintf(PETSC_COMM_WORLD, "Solution:\n");
    VecView(x, PETSC_VIEWER_STDOUT_WORLD);

    // Clean up
    KSPDestroy(&solver);
    VecDestroy(&x);
    VecDestroy(&b);
    MatDestroy(&A);
    PetscFinalize();
    MPI_Finalize();

    return 0;
}
```

This code is a sample program that demonstrates how to use the PETSc library with MPI to solve a linear system of equations. Specifically, the program creates a square matrix A, a right-hand side vector b, and a solution vector x, fills them with values, and then solves the linear system Ax=b using the conjugate gradient method provided by the PETSc KSP (Krylov Subspace Methods) library.

To compile it, use the following command linking against petsc library,

```
mpicxx -o example sample.cpp -lpetsc
```

To execute it, use the following command,

```
mpiexec -np 4 ./example
```

Here is a slurm script, that wraps all the process described above,

```
#!/bin/bash
#SBATCH --job-name=example
#SBATCH --ntasks=4
#SBATCH --time=00:10:00
#SBATCH --output=example.out
#SBATCH --error=example.err
#SBATCH --p main
#SBATCH --qos  main

# Load necessary modules
module load petsc/3.3-p7-gcc-mpich

# Compile the code
mpicxx -o example sample.cpp -lpetsc
 # Run the code
mpiexec -n 4 ./example
```

Where to find help?

If you are confused or need help at any point, please contact OIT at the following address.

https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp