dCSE ICOM

Dr. Xiaohu Guo
ARC Group ,CSE Department, STFC

Science & Technology Facilities Council

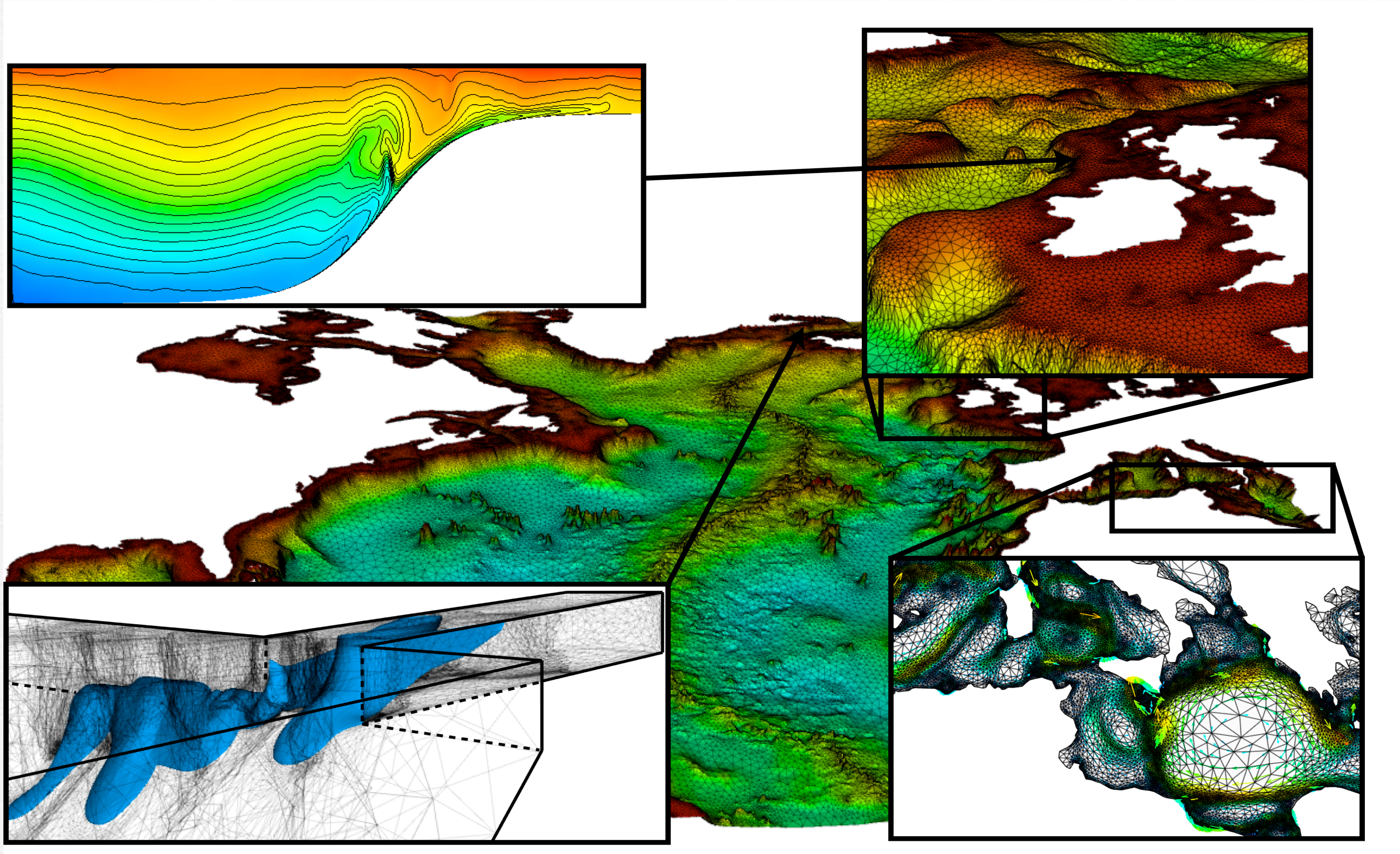
Contents

- □ ICOM Introduction
- □ The Initial Project Plan
- □ Profiling with CrayPAT
- □ PETSc Profiling
- □ The Future Work

Science & Technology
Facilities Council

Imperial College Ocean Modeling (ICOM)

Science & Technology Facilities Council



Science & Technology Facilities Council

Computational Characteristic

- Unstructured FEM Code
- Adaptive Mesh, solving from large scales to small scales
- Most time solving Ax=b, where A is Sparse Matrix
  - FEM Matrix assembling
  - Using PETSc's preconditioner and Solver
  - Most Computing time is spent here
- Fortran, C++/C, Python
- MPI Based, well tested on processes from 1 to 64.
  - dynamic load-balanced domain decomposition methods

Computational Characteristic

Indirect Addressing,

- □ Unstructured FEM Code
- □ Adaptive Mesh, solving from large scales to small scales
- □ Most time solving Ax=b, where A is Sparse Matrix
  - □ FEM Matrix assembling
  - □ Using PETSc's preconditioner and Solver
  - □ Most Computing time is spent here
- □ Fortran, C++/C, Python
- □ MPI Based, well tested on processes from 1 to 64.
  - □ dynamic load-balanced domain decomposition methods

Science & Technology Facilities Council

Required Packages

- □ VTK
- □ CGNS
- □ BLAS
- □ LAPACK
- □ XML2
- □ MPI
- □ PETSc
- □ ParMetis
- □ APPACK
- □ NetCDF
- □ UDUnits
- □ Python Development Environments
- □ Trang
- □ Spatial-Index
- □ Fortran 90 Compilers
- □ C++
- □ Subvision(SVN)

Collaborations

- Applied Modelling and Computation Group, Imperial College, London (AMCG, http://amcg.ese.ic.ac.uk/)
- ARC,The Computational Science & Engineering Department(CSED),STFC (http://www.cse.clrc.ac.uk/)
- Proudman Oceanographic Laboratory, Liverpool (POL, http://www.pol.ac.uk/)

Science & Technology Facilities Council

The Initial Project Plan

- □ Installing code on Hector.
  - □ Python environment
  - □ GNU Installation
  - □ PGI Installation
- □ Profiling and Optimization
  - □ Hardware Counters for individual processor.
  - □ Parallel Efficiency, Load balancing, Communication Overheads
- □ Parallel I/O

Science & Technology Facilities Council

Installing ICOM on Hector

---issues and experiences

- □ Python Environments for ICOM
  - □ python-CNL
- □ GNU Installation
  - □ PETSc Module, NETCDF/4.0 Module,
- □ PGI Bugs

Science & Technology Facilities Council

Installing ICOM on Hector

---issues and experiences

- □ Python Environments for ICOM
  - □ python-CNL
- □ GNU Installation
  - □ PETSc Module, NETCDF/4.0 Module,

```
Lowering Error: bad ast optype in expression [ast=1339,asttype=12,datatype=0]
Lowering Error: bad ast optype in expression [ast=1339,asttype=12,datatype=0]
PGF90-F-0000-Internal compiler error. Errors in Lowering       2 (Halos_Numbering.F90:
```

```
do i = 1, nprocs
      local_nodes(halo_receives(halo, i)) = .false.
end do


function halo_receive_count(halo, process)
    !!< Retrieve the number of receive nodes in the supplied halo
    !!< for the supplied process

    type(halo_type), intent(in) :: halo
    integer, intent(in) :: process

    integer :: halo_receive_count

    assert(process > 0)
    assert(process <= halo_proc_count(halo))

    assert(associated(halo%receives))
    assert(associated(halo%receives(process)%ptr))

    halo_receive_count = size(halo%receives(process)%ptr)

  end function halo_receive_count
```

```
do i = 1, nprocs
   local_nodes(halo_receives(halo, i)) = .false.
end do


function halo_receive_count(halo, process)
  !!< Retrieve the number of receive nodes in the supplied halo
  !!< for the supplied process

  type(halo_type), intent(in) :: halo
  integer, intent(in) :: process

  integer :: halo_receive_count

  assert(process > 0)
  assert(process <= halo_proc_count(halo))

  assert(associated(halo%receives))
  assert(associated(halo%receives(process)%ptr))

  halo_receive_count = size(halo%receives(process)%ptr)

 end function halo_receive_count
```

```
integer, dimension(:), pointer :: nnntmp

do i = 1, nprocs
   nnntmp => halo_receives(halo, i)
   local_nodes(nnntmp) = .false.
end do

function halo_receive_count(halo, process)
  !!< Retrieve the number of receive nodes in
  !!< the supplied halo for the supplied process

  type(halo_type), intent(in) :: halo
  integer, intent(in) :: process

  integer :: halo_receive_count

  assert(process > 0)
  assert(process <= halo_proc_count(halo))

  assert(associated(halo%receives))
  assert(associated(halo%receives(process)%ptr))

  halo_receive_count = size(halo%receives(process)%ptr)

 end function halo_receive_count
```

```
do i = 1, nprocs
      local_nodes(halo_receives(halo, i)) = .false.
end do


function halo_receive_count(halo, process)
    !!< Retrieve the number of receive nodes in the supplied halo
    !!< for the supplied process

    type(halo_type), intent(in) :: halo
    integer, intent(in) :: process

    integer :: halo_receive_count

    assert(process > 0)
    assert(process <= halo_proc_count(halo))

    assert(associated(halo%receives))
    assert(associated(halo%receives(process)%ptr))

    halo_receive_count = size(halo%receives(process)%ptr)

  end function halo_receive_count
```

```
integer, dimension(:), pointer :: nnntmp

do i = 1, nprocs
      nnntmp => halo_receives(halo, i)
      local_nodes(nnntmp) = .false.
end do

function halo_receive_count(halo, process)
    !!< Retrieve the number of receive nodes in
    !!< the supplied halo for the supplied process

    type(halo_type), intent(in) :: halo
    integer, intent(in) :: process

    integer :: halo_receive_count

    assert(process > 0)
    assert(process <= halo_proc_count(halo))

    assert(associated(halo%receives))
    assert(associated(halo%receives(process)%ptr))

    halo_receive_count = size(halo%receives(process)%ptr)

  end function halo_receive_count
```

BUGGY ! BUGGY ! BUGGY!

Science & Technology
Facilities Council

Profiling with CrayPAT
-----Issues and Results

Science & Technology
Facilities Council

Issues with CrayPAT

Sampling or Tracing ?

- □ pat_build options for large applications
  - □ Automatic Program Analysis should be used to sampling for large Applications
  - □ Direct using "-w", "-u", may generate large data file
- □ Profiling with PETSc
- □ Data reporting format ?

Science & Technology Facilities Council

Hardware Counters for Individual Processor
------- Some Results from CrayPAT

Science & Technology
Facilities Council

Instruction Metrics

```
========================================================================
Totals for program
------------------------------------------------------------------------
  Time%                                  100.0%
  Time                               853.138193 secs
  Imb.Time                                   -- secs
  Imb.Time%                                  --
  Calls                  0.123M/sec    102466624.5 calls
  PAPI_L1_DCM           20.523M/sec    17054085234 misses
  PAPI_TOT_INS        2348.976M/sec  1951908350570 instr
  PAPI_L1_DCA          976.804M/sec   811686199323 refs
  PAPI_FP_OPS          148.933M/sec   123757242660 ops
  User time (approx)     830.961 secs  1911211054688 cycles  97.4%Time
  Average Time per Call                     0.000008 sec
  CrayPat Overhead : Time   16.1%
  HW FP Ops / User time   148.933M/sec   123757242660 ops  1.6%peak(DP)
  HW FP Ops / WCT        145.061M/sec
  HW FP Ops / Inst                         6.3%
  Computational intensity    0.06 ops/cycle      0.15 ops/ref
  Instr per cycle                        1.02 inst/cycle
  MIPS              300668.95M/sec
  MFLOPS (aggregate)     19063.37M/sec
  Instructions per LD & ST  41.6% refs        2.40 inst/ref
  D1 cache hit,miss ratios  97.9% hits        2.1% misses
  D1 cache utilization (M)  47.59 refs/miss     5.949 avg uses
========================================================================
```

Higher or Lower ?

Science & Technology Facilities Council

Floating point operations mix

```
==========================================================================
Totals for program
--------------------------------------------------------------------
  Time%                                    100.0%
  Time                              848.475046 secs
  Imb.Time                                    -- secs
  Imb.Time%                                   --
  Calls                 0.124M/sec    102466624.5 calls
  RETIRED_MMX_AND_FP_INSTRUCTIONS:
    PACKED_SSE_AND_SSE2  246.314M/sec   203225296909 instr
  PAPI_FML_INS            77.419M/sec    63876051299 ops
  PAPI_FAD_INS            72.577M/sec    59881191368 ops
  PAPI_FDV_INS             0.741M/sec      611424499 ops
  User time (approx)     825.066 secs  1897652554688 cycles  97.2%Time
  Average Time per Call                   0.000008 sec
  CrayPat Overhead : Time  16.6%
  HW FP Ops / Cycles                        0.07 ops/cycle
  HW FP Ops / User time  149.997M/sec   123757242668 ops  1.6%peak(DP)
  HW FP Ops / WCT        145.858M/sec
  FP Multiply / FP Ops                      51.6%
  FP Add / FP Ops                         48.4%
  MFLOPS (aggregate)    19199.58M/sec
==========================================================================
```

cache (L1 and L2 metrics)

```
========================================================================
Totals for program
------------------------------------------------------------------------
 Time%                                          100.0%
 Time                                   819.006943 secs
 Imb.Time                                      -- secs
 Imb.Time%                                          --
 Calls                  0.128M/sec    102466624.5 calls
 REQUESTS_TO_L2:DATA         32.920M/sec    26327242071 req
 DATA_CACHE_REFILLS:
  L2_MODIFIED:L2_OWNED:
  L2_EXCLUSIVE:L2_SHARED     19.521M/sec     15612010608 fills
 DATA_CACHE_REFILLS_FROM_SYSTEM:
  ALL                   12.722M/sec     10174567842 fills
 PAPI_L1_DCA              955.468M/sec   764127945223 refs
 User time (approx)        799.742 secs  1839406671875 cycles  97.6%Time
 Average Time per Call                        0.000008 sec
 CrayPat Overhead : Time       16.7%
 D1 cache hit,miss ratio (R)  96.6% hits          3.4% misses
 D1 cache utilization          29.63 refs/refill   3.704 avg uses
 D2 cache hit,miss ratio        61.4% hits          38.6% misses
 D1+D2 cache hit,miss ratio   98.7% hits          1.3% misses
 D1+D2 cache utilization       75.10 refs/miss     9.388 avg uses
 System to D1 refill          12.722M/sec     10174567842 lines
 System to D1 bandwidth       776.508MB/sec  651172341865 bytes
 L2 to Dcache bandwidth      1191.486MB/sec  999168678900 bytes
========================================================================
```

TLB

```
========================================================================
Totals for program
------------------------------------------------------------------------
  Time%                                    100.0%
  Time                              801.013149 secs
  Imb.Time                                      -- secs
  Imb.Time%                                      --
  Calls                  0.125M/sec     98035814.1 calls
  PAPI_L1_DCM             21.003M/sec    16436513322 misses
  PAPI_TLB_DM              0.356M/sec      278678034 misses
  PAPI_L1_DCA            953.479M/sec   746188830992 refs
  PAPI_FP_OPS            158.137M/sec   123757242646 ops
  User time (approx)      782.596 secs  1799969937500 cycles  97.7%Time
  Average Time per Call                       0.000008 sec
  CrayPat Overhead : Time   16.3%
  HW FP Ops / User time   158.137M/sec   123757242646 ops  1.7%peak(DP)
  HW FP Ops / WCT        154.501M/sec
  Computational intensity    0.07 ops/cycle      0.17 ops/ref
  MFLOPS (aggregate)     20241.52M/sec
  TLB utilization        2677.60 refs/miss     5.230 avg uses
  D1 cache hit,miss ratios  97.8% hits          2.2% misses
  D1 cache utilization (M)  45.40 refs/miss     5.675 avg uses
========================================================================
```

Vectorization

```
=================================================================
Totals for program
-------------------------------------------------------------------------
 Time%                                        100.0%
 Time                                  817.257585 secs
 Imb.Time                                          -- secs
 Imb.Time%                                          --
 Calls                    0.129M/sec    102466624.5 calls
 RETIRED_SSE_OPERATIONS:
   SINGLE_ADD_SUB_OPS:
   SINGLE_MUL_OPS               10 /sec           7747 ops
 RETIRED_SSE_OPERATIONS:
   DOUBLE_ADD_SUB_OPS:
   DOUBLE_MUL_OPS         153.748M/sec   122294107689 ops
 RETIRED_SSE_OPERATIONS:
   SINGLE_ADD_SUB_OPS:
   SINGLE_MUL_OPS:OP_TYPE       10 /sec           7747 ops
 RETIRED_SSE_OPERATIONS:
   DOUBLE_ADD_SUB_OPS:
   DOUBLE_MUL_OPS:OP_TYPE  155.588M/sec   123757234945 ops
 User time (approx)        795.419 secs  1829462945312 cycles  97.3%Time
 Average Time per Call                 0.000008 sec
 CrayPat Overhead : Time      16.5%
=================================================================
```

Bandwidth

```
========================================================================
Totals for program
------------------------------------------------------------------------
 Time%                                      100.0%
 Time                                 806.077027 secs
 Imb.Time                                       -- secs
 Imb.Time%                                        --
 Calls                     0.144M/sec     102466624.5 calls
 QUADWORDS_WRITTEN_TO_SYSTEM:
  ALL                      42.360M/sec     30124086020 ops
 DATA_CACHE_REFILLS:
  L2_MODIFIED:L2_OWNED:
  L2_EXCLUSIVE:L2_SHARED      22.001M/sec     15645972281 fills
 DATA_CACHE_REFILLS_FROM_SYSTEM:
  ALL                      14.308M/sec     10174900337 fills
 DATA_CACHE_LINES_EVICTED:ALL  50.686M/sec     36045196387 ops
 User time (approx)          711.153 secs   1635651072613 cycles  88.2%Time
 Average Time per Call                        0.000008 sec
 CrayPat Overhead : Time        16.9%
 System to D1 refill          14.308M/sec     10174900337 lines
 System to D1 bandwidth       873.268MB/sec   651193621587 bytes
 L2 to Dcache bandwidth      1342.826MB/sec  1001342226008 bytes
 L2 to System BW per core     323.178MB/sec   240992688159 bytes
========================================================================
```

Some Scalability Results

Science & Technology Facilities Council

TTable 1: Profile by Function Group and Function

```
 Time % |      Time |  Imb. Time |  Imb. |     Calls |Experiment=1
        |           |            | Time % |          |Group
        |           |            |        |          | Function
        |           |            |        |          |  PE='HIDE'

 100.0% | 394.147622 |         -- |     -- | 15360515.5 |Total
|-------------------------------------------------------------------
|  42.7% | 168.138632 |         -- |     -- |    16155.0 |MPI_SYNC
||------------------------------------------------------------------
||  18.7% |  73.769241 |  10.789873 |  12.8% |      813.0 |mpi_allreduce_(sync)
||  16.7% |  65.834826 |  20.223770 |  23.5% |    13636.0 |MPI_Allreduce(sync)
||   2.4% |   9.587557 |   0.133895 |   1.4% |     1251.0 |MPI_Allgather(sync)
||   1.7% |   6.804828 |   1.433497 |  17.4% |      169.0 |mpi_barrier_(sync)
||   1.0% |   3.877275 |   0.630532 |  14.0% |        6.0 |MPI_Bcast(sync)
||==================================================================
|  35.3% | 139.269664 |         -- |     -- |   726334.8 |USER
||------------------------------------------------------------------
||  30.2% | 119.119262 | 113.910394 |  48.9% |        1.0 |main
||   5.1% |  20.150291 |   6.065052 |  23.2% |   726332.8 |#1.construct_mom_ele_cg
||==================================================================
|  22.0% |  86.739285 |         -- |     -- | 14617931.8 |MPI
||------------------------------------------------------------------
||  15.9% |  62.665570 |  29.365896 |  31.9% |  1430640.4 |MPI_Waitany
||   2.2% |   8.834029 |  21.319478 |  70.8% |  1427807.8 |MPI_Start
||   1.4% |   5.326372 |   0.792849 |  13.0% |    13636.0 |MPI_Allreduce
|===================================================================
```

Science & Technology
Facilities Council

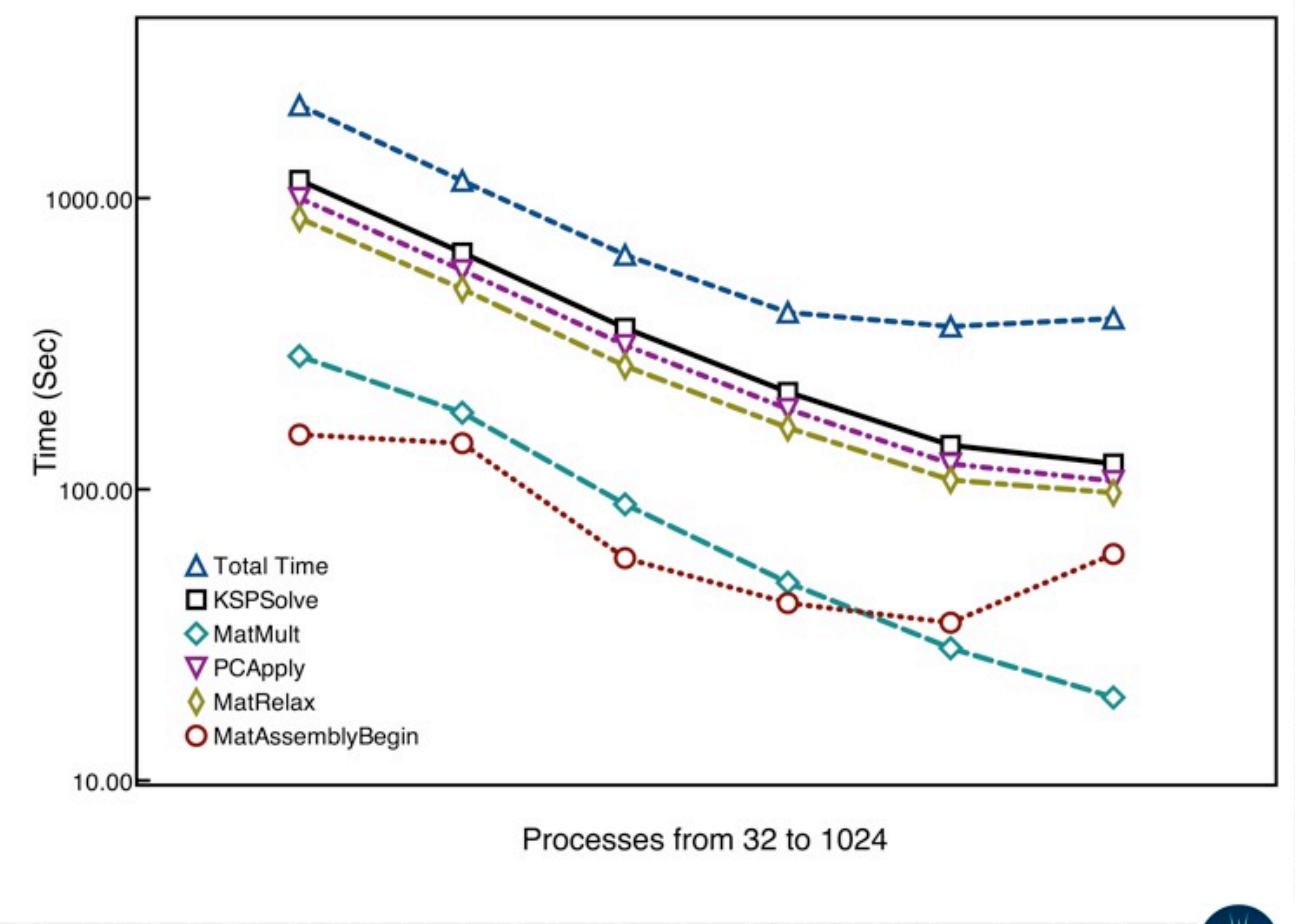
TTable 3: Load Balance with MPI Message Stats

```
 Time % |      Time | MPI Msg |    MPI Msg |Avg MPI |Experiment=1
        |           |   Count |      Bytes |    Msg |Group
        |           |         |            |   Size | PE[mmm]

 100.0% | 415.113351 | 76689.7 | 16669148.0 | 217.36 |Total
|---------------------------------------------------------------
|  40.5% | 168.160689 |      -- |         -- |     -- |MPI_SYNC
||--------------------------------------------------------------
||   0.0% | 195.963445 |      -- |         -- |     -- |pe.211
||   0.0% | 167.655654 |      -- |         -- |     -- |pe.250
||   0.0% |  43.578608 |      -- |         -- |     -- |pe.0
||==============================================================
|  33.8% | 140.261359 |      -- |         -- |     -- |USER
||--------------------------------------------------------------
||   0.1% | 260.182527 |      -- |         -- |     -- |pe.0
||   0.0% | 140.918415 |      -- |         -- |     -- |pe.383
||   0.0% | 104.965772 |      -- |         -- |     -- |pe.211
||==============================================================
|  25.7% | 106.691133 | 76689.7 | 16669148.0 | 217.36 |MPI
||--------------------------------------------------------------
||   0.0% | 118.929751 | 76986.0 | 14962336.0 | 194.35 |pe.118
||   0.0% | 107.291485 | 73851.0 | 14093920.0 | 190.84 |pe.297
||   0.0% |  94.469296 | 79494.0 | 18782760.0 | 236.28 |pe.577
|================================================================ext
```

Science & Technology
Facilities Council

PETSc Own Profiling Tools



Science & Technology
Facilities Council

Future Work --- Parallel I/O

- Current I/O
  - each process writes out its own solution domain to disk
  - File format, vtu
- Parallel I/o
  - a subset of the processors will perform the reading and writing, with the number of I/O processors tuned to match the number of Object Storage Targets (OST) in the Lustre filesystem.
  - File format: CGNS (build upon HDF5)

Summary

- Ported code to Hector
    - The code required lots of third party packages
    - Exposed bugs in PGI compiler
- Initial CrayPAT Analysis, Sampling v.s. Tracing
- Initial optimization will concentrate on parallel I/O
- Most PETSC functions are scaling quite well, except MatAssemblyBegin

Science & Technology Facilities Council

Acknowledgment

- Thanks to Hector Help Desk for assistance.
- Thanks to ARC Group for advice and suggestions

Thanks !

Science & Technology Facilities Council