Data Analysis, Visualization, and Storage, Part 1

Presented by Tom Peterka

Additional support from Rob Ross, Rob Latham, Mike Papka, Jon Woodring, Jim Ahrens, Venkat Vishwanath, Mark Hereld, Joe Insley, Tim Tautges, Vijay Mahadevan

Tom Peterka
tpeterka@mcs.anl.gov
Mathematics and Computer Science Division

CESAR All hands meeting
1/19/12

High-Level Goals

- Data models
  - Coupled, uniform data model, while retaining individual code data models
- Storage performance
  - Improve parallel storage performance at high concurrency, staging, topology-awareness
- Run-time analysis
  - In situ, coprocessing, feature extraction, native data structures
- Postprocessing analysis
  - Needs derived from visualization workflows and research contributed back to tools like ParaView, VisIt

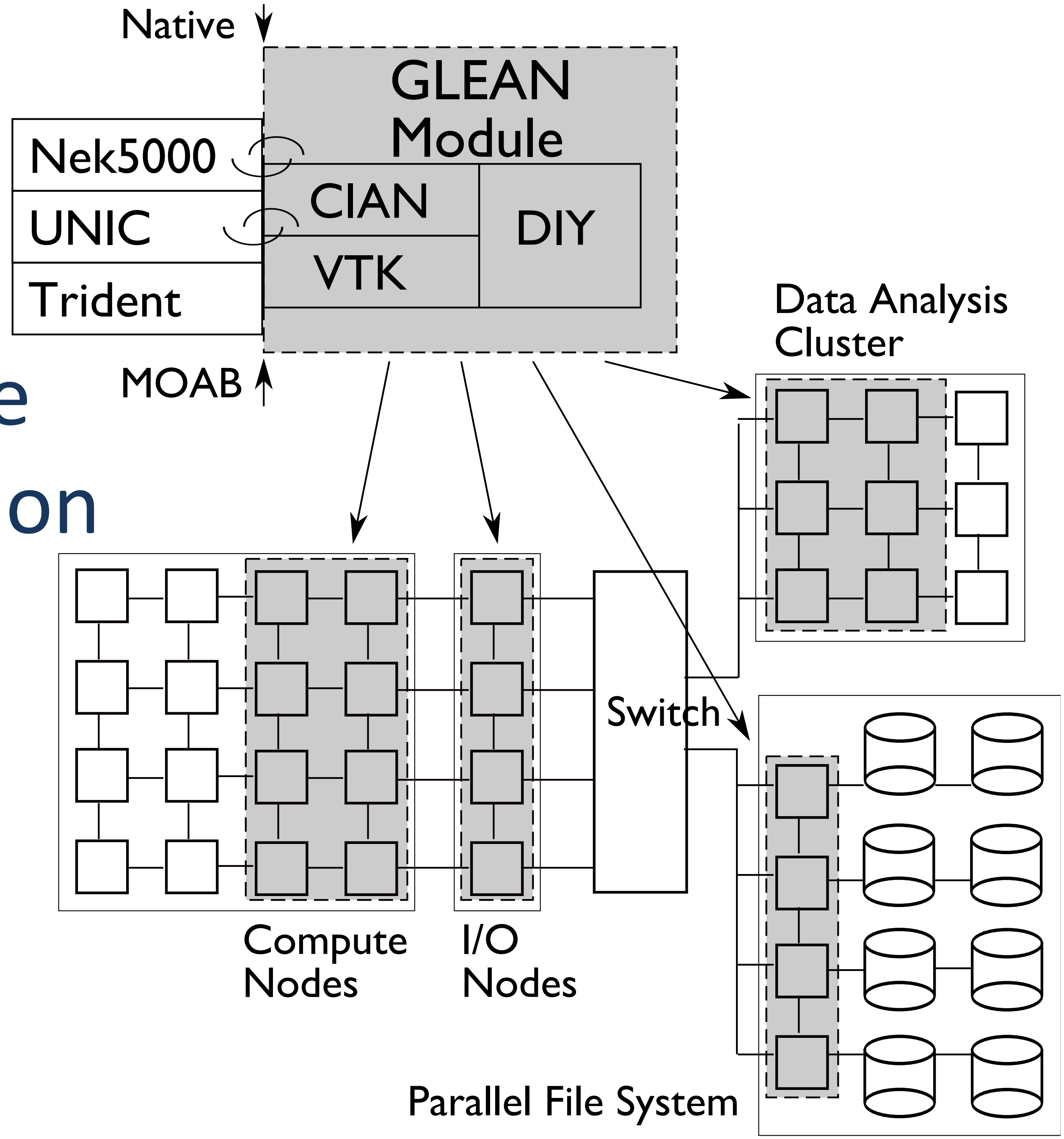
Elements of Codesign Solutions

- Meshing and coupling data models (Tim Tautges)
  - MOAB
- Storage performance of systems and libraries (Rob Latham, Venkat Vishwanath)
  - HDF5, Damsel, PVFS, ROMIO, MPI-IO, Darshan, GLEAN
- Postprocessing parallel analysis (Joe Insley, Venkat Vishwanath)
  - ParaView, VisIt
- Run-time parallel analysis (Tom Peterka, Venkat Vishwanath, Joe Insley, Jon Woodring)
  - DIY, VTK, GLEAN, MOAB

CESAR Datavis Organization

- Native interfaces
  - This talk
  - Roughly corresponds to work package #6: Peterka/Ross/Papka/Tautges/Ahrens: custom visualizations to read NEK/UNIC output (Lusk)
- MOAB interface
  - Tomorrow
  - Roughly corresponds to work package #7: Tautges/Peterka: MOAB interface for coupling NEK.UNIC at scale (Siegel)
- Internally, we combine our work into one datavis group
  - Tim T., Tom P., Rob R., Rob L., Jon W. Jim A., Venkat V., Joe I., Mark H., Vijay M.
  - There will be some crossover and overlap between talks

Software Organization



Application Code Changes for In Situ Analysis in Nek5000

```
subroutine userchk
include 'SIZE'
include 'TOTAL'
common /exacu/ ue(lx1,ly1,lz1,lelt),ve
      (lx1,ly1,lz1,lelt)
common /exacd/ ud(lx1,ly1,lz1,lelt),vd
      (lx1,ly1,lz1,lelt)


call cian


    …
```

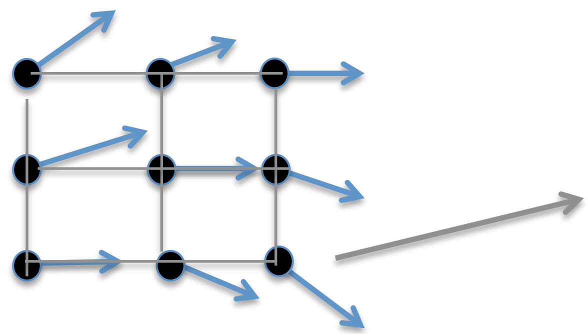
```
subroutine cian

integer first

        ! --- first time step ---
save first
data first /1/
if (first.eq.1) then
  first = 0
  call CIAN_begin(nelv, 1)
   call CIAN_irg_geom(xm1, ym1, zm1, nx1 *
      ny1 * nz1 * nelv, 0)
endif

      ! --- every time step ---
call CIAN_lambda2(nx1 * ny1 * nz1 * nelv,
      -1.0, 0.0)

      ! --- last time step ---
if (lastep.eq.1) then
   call CIAN_end()
endif

return
end
```

CIAN: Cesar Exascale Analysis API

```
void CIAN_begin (int *nblocks);

void CIAN_scalar_data(double *vals, int *npts, int *ofst);

void CIAN_vector_data(double *vx, double *vy, double *vz,

    int *npts, int *ofst);

void CIAN_geom(double *x, double *y, double *z, int *npts, int *ofst);

void CIAN_lambda2(int *nvals, double *min_val, double *max_val);

#ifdef MOAB

void CIAN_mesh(iMesh_Instance *mesh);

#endif
```

An Example Problem: Lambda-2 Vorticity

A 2D example: Given a velocity vector field ***u***:



1. Compute a tensor A at each grid point

$$A = \begin{bmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{yy} \end{bmatrix}$$

Where $a_{ij} = [1/2(u_{ij} + u_{ji})]^2 + [1/2(u_{ij} - u_{ji})]^2$

and u_{ij} is the 2nd partial derivative of ***u*** Eg.

$$u_{x,y} = \frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right) = \frac{\partial^2 u}{\partial y \partial x}$$

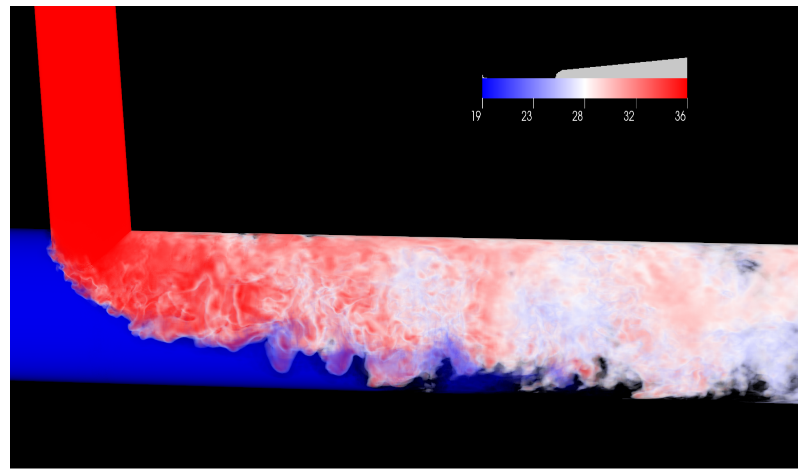
2. Compute the eigenvalues of A by solving the characteristic equation of A, which has three real roots.

3. Compute a derived scalar field of the second eigenvalue, λ_2

4. Keep those points where min < λ_2 < 0

Early Successes

- Demonstrated successfully running MOAB-enabled Nek
- Demonstrated getting data out of native Nek
- Demonstrated calling back into native Nek and using its functions
- Demonstrated computing lambda-2 vorticity and filtering it to a small subset of values
- Compiled and ran UNIC test problem



• Captured the current Visit visualization workflow for NEK

Ongoing

- Getting data and geometry out of MOAB-enabled Nek
- Sending filtered lambda-2 field to coprocessing or postprocessing visualization and comparing lambda-2 field with external analysis in VTK's lambda-2 function
- Understanding and capture of the current data model of NEK and UNIC as well as future extensions

Ongoing

- Data staging for improving the I/O performance of NEK, UNIC and MOAB
- Developing a scalable reader for Nek for visualization including interpolation in spectral space for improved accuracy
- Run-time analysis of Nek (Wall Shear Stress, Vorticity, Histograms, etc.) and understand the right place and time to perform simulation-time analysis and visualization
- Porting and tuning GLEAN and DIY for BG/Q

Next Steps: Direct Nek5000 In Situ with ParaView

Parallel work on VTK no-copy memory interface to allow ParaView/VTK filters to run on native Nek as well as MOAB meshes

- Direct access to Nek memory through ParaView In Situ memory adapter
- Evaluate In Situ with Nek
- Deep copy Nek data to VTK at first, no-copy memory adapter later
- In situ Nek + In Situ ParaView demo to be completed, to be compared with Nek + MOAB + In Situ ParaView demo

Next Steps: What We Can Learn at This Meeting

- Continued and better understanding of applications analysis / storage / visualization (data) needs
- Look beyond today's data needs
- Learn how the applications could use analysis fed back to it at run time
- Understand the role of the coupled Trident code, so we can begin to predict its data needs
- Sample datasets, large and small, and run configurations
- Connection with other work packages and groups