The Scalable Data Management, Analysis, and Visualization (SDAV) Institute

Key Technical Accomplishments

(Also, check out 8 posters)  

http://sdav-scidac.org/
SDAV Institute

Arie Shoshani (PI)

Co-Principal Investigators from:

Laboratories
- Argonne NL
- Lawrence Berkeley NL
- Lawrence Livermore NL
- Oak Ridge NL
- Los Alamos NL
- Sandia NL
- Kitware (Industry)

Universities
- Georgia Inst of Technology
- North Carolina State U
- Northwestern U
- Ohio State U
- U of California, Davis
- Rutgers U
- U of Utah

24 people attending

SciDAC PI meeting 2015
SDAV Approach to Productivity and Relevance

- Main Emphasis over the last year
  - Parallel processing on multi/many cores
  - *In situ* processing on leadership-class facilities (LCFs)
  - Support ever growing volume of data (TBs – PDs)
  - Collaborate with application scientists to enhance their data understanding and insight
  - Pro-active installation and support of software on LCFs

- Libraries, tools, and Frameworks
  - Libraries – software package that can be invoked (or embedded) by other programs through APIs
    - e.g. in-situ data movement, code-coupling, indexing, rendering, compression
  - Tools – stand-alone software package – usually support user interfaces
    - e.g. topology-based analysis, feature tracking, flow analysis over space/time, I/O monitoring
  - Frameworks – software that can imbed multiple libraries
    - SDAV has mainly I/O and visualization frameworks
**SDAV Portfolio**
(http://sdav-scidac.org/toolkit.html)

---

### Data Management tools

<table>
<thead>
<tr>
<th>I/O Frameworks</th>
<th>In Situ Processing</th>
<th>Indexing / Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADIOS</td>
<td>GLEAN</td>
<td>FastBit</td>
</tr>
<tr>
<td>Darshan</td>
<td>DIY</td>
<td>ISABELA</td>
</tr>
<tr>
<td>Parallel netCDF</td>
<td>DataSpaces</td>
<td></td>
</tr>
<tr>
<td>ROMIO</td>
<td>EvPath</td>
<td></td>
</tr>
<tr>
<td>ViSUS/IDX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
SDAV Portfolio
(http://sdav-scidac.org/toolkit.html)

Analysis and Visualization tools

- **Analysis and Visualization Frameworks**
  - Visit
  - ParaView

- **Analysis and Visualization Tools and Libraries**
  - ExMage
  - Ultravis-P
  - IceT
  - VTK
  - TALASS
  - MSCEER
  - NDDAV

- **Multi-/Many-core Visualization Libraries**
  - Dax
  - EAVL
  - PISTON
# Deployment of SDAV Software

<table>
<thead>
<tr>
<th>Software</th>
<th>ALCF</th>
<th>NERSC</th>
<th>OLCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADIOS</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>CEDMAV</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>DataSpaces</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>Darshan</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>DIY</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>GLEAN</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>Fastbit</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>FastQuery</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>Flexpath</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>IceT</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software</th>
<th>ALCF</th>
<th>NERSC</th>
<th>OLCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSUFlow</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>ParaView</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>ParallelNetCDF</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>PIDX</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>UCDVis</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>VisIt</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>VisUS</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>VTK</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>VTK-m</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
<tr>
<td>Warp+VisIt</td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
<td><img src="image" alt="Status" /></td>
</tr>
</tbody>
</table>

Legend:
- **Red**: Installed
- **Green**: Installed in other tools
- **Blue**: Local, unsupported

---

Note: The status icons are placeholders and should be replaced with actual icons representing the status of each software on each site.
Visualization
Frameworks and tools
VTK-m: Accelerating the Visualization Toolkit for Multi-core and Many-core Architectures

VTK-m goals

- A single place for the visualization community to collaborate, contribute, and leverage massively threaded algorithms.
- Reduce the challenges of writing highly concurrent algorithms by using data parallel algorithms.

- **VTK** (Kitware) is a serial, single-threaded class library (data structures and algorithms) used as the basis for important applications (VisIt, PV).
- **EAVL** (ORNL) emphasizes the development of a new data model,
- **Dax** (Sandia) emphasizes the development of a new execution model,
- **PISTON** (LANL) emphasizes portability and parallel algorithm development.

**VTK-m constituent technologies**

- **VisIt, ParaView**
- **VTK-m**
- **Phi, Tesla, x86**

**VTK-M**
VTK-m Progress and Results

Features
- Device Interface (Serial, CUDA, TBB; OpenMP in progress)
- Architecture allows hardware-agnostic implementations
- Field and Topology Worklet and Dispatcher implemented

Filters
- Isosurface for structured grids
- Statistical filters (histograms, moments, etc.)
- In development: stream lines, stream surfaces, tetrahedralization

Example: ray casting with DAX and VTK-m
- Implementation of both ray-casting and cell projection volume rendering algorithms using Dax, one of VTK-m’s constituent projects
- Complied for CUDA, OpenMP, and Intel’s Thread Building Blocks
- Performance study on NVIDIA Titan X GPU, Intel Xeon, and Intel Xeon Phi
- Cell projection implementation using data parallel primitives renders image in near sub-second times.

Volume rendering of type Ia supernova simulation data set using ray-casting
Continued collaboration and use of visualization tools with specific science code

Example: \textit{in situ} Visualization and Analysis of Particle Accelerator Simulations using WarpVisIt

components
- Warp framework: PIC modeling of beams, accelerators, plasmas, …
- VisIt: parallel visualization tool

Problem
- Warp worked with the popular OpenDX (data explorer)
- $10^7$ to $10^9$ particles are required for simulation but often only a small fraction form features of interest

Solution
- Couple state-of-the-art \textit{in situ} visualization tool VisIt with Warp
- Integrate \textit{in situ} query capabilities with Warp
- Integrate high-performance I/O with Warp
- Provides dynamic feature exploration while simulation is in progress
in situ Visualization and Analysis of Particle Accelerator Simulations using WarpVisIt

**Impact**

- Enable flexible *in situ* analysis of particle features
- Reduce cost for visualization and I/O
- Enable analysis and collection of particle subsets of interest at higher temporal frequency

**Enable Filtered Species**

- Easy definition and implementation of derived particle species while exposing to the analysis the same species interface as Warp

---

**a)** Without halo filter

**b)** With halo filter
Data Analysis Tools
Analysis Tools Applied to Various Applications

- **Feature-driven analysis**
  - *Dynamic tracking graphs and multi-core exploration, e.g.*, Localized thresholds for vortex detection

- **Importance driven analysis**
  - *In-situ data triage/reduction, e.g.*, Explorable images

- **Statistical analysis and data mining**
  - *Feature-based algorithms in MPI, e.g.*, Memory efficient graph analytics

- **Topological techniques**
  - *Integrate higher order feature descriptors, e.g.*, Abstract representation of material structure to investigate Lithium diffusion

- **Vector field analysis**
  - *Incorporate advanced statistical methods, e.g.*, Flowline indexing via geometric signatures
Problem: Lithium Diffusion in Carbon Nanospheres

- understanding their cycling and capacity to store lithium is not possible with classical MD simulations
- Instead, models for lithium diffusion are used where the valence of a carbon ring determines whether or not lithium is able to pass through

Tool used for modeling structure

- MSCEER (Morse-Smale Complex Exploration, Evaluation, and Reasoning) is a set of tools for delivering state-of-the-art topology-based analytics to the scientific community.

New algorithm developed

- A new algorithm in MSCEER based on discrete Morse theory operates on a Delaunay triangulation, explicitly identifying ring structures and their valence

Result

- Allow interactive visualization of blocking and non-blocking structures, for the first time showing the planes along which lithium can diffuse. Penetration into the nanosphere is dominated by large-scale dislocations.
Analyzing and Tracking Features in Large-Scale Turbulent Combustion at Scale (using TALASS)

Understanding Flame Stability in Turbulent Non-Premixed Combustion
• Find when and where a flame becomes unstable and undergo significant extinction and re-ignition decreasing efficiency and increasing emissions.

Challenges
• Complex fast moving geometry
• Extinction threshold is uncertain and unstable

Tool used to enhance understanding
• Topological Analysis of Large Scale Simulations (TALASS)
• Generate graphs that track flame progression over time

Results
• Modified graph using adaptive threshold to produce cleaner tracks, easier to understand and analyze.
Data and Storage Management
The Traditional Post-Processing I/O Stack

high-level I/O interfaces: HDF5, Parallel netCDF

MPI-IO implementation: ROMIO is the basis for all platforms today.

production software: GPFS (IBM), Lustre (Intel/Whamcloud)

Application

High-Level I/O Libraries

I/O Middleware

I/O Forwarding / Routing

Parallel File System

I/O Hardware

Monitoring application I/O behavior at scale: Darshan running on production platforms, with negligible overhead.
SDAV Data management frameworks provide a vehicle for deployment of *in situ* data analysis/transformations.

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Management Framework</td>
</tr>
<tr>
<td>Data Model Interface(s)</td>
</tr>
<tr>
<td>(e.g., adaptive mesh)</td>
</tr>
<tr>
<td>Data Management Services</td>
</tr>
<tr>
<td>(e.g., buffering, scheduling, code coupling)</td>
</tr>
<tr>
<td>Data Transformations</td>
</tr>
<tr>
<td>(e.g., indexing, compression, <em>in situ</em> analysis)</td>
</tr>
<tr>
<td>Parallel File System</td>
</tr>
</tbody>
</table>
ADIOS (Adaptable I/O System)

- Abstracts Data-at-Rest to Data-in-Motion (*in situ*) for HPC
  - Provides portable, fast, scalable, easy-to-use, metadata rich output
  - Dynamically allows users to change the method during an experiment/simulation
  - Provides solutions for a large number of the applications
  - Applied to extremely large data volumes

- Astrophysics
- Climate
- Combustion
- CFD
- Environmental Science
- Fusion
- Geoscience
- Materials Science
- Medical:
  - Pathology
  - Neutron Science
  - Nuclear Science
  - Quantum Turbulence
  - Relativity
  - Seismology
  - Sub-surface modeling
  - Weather
I/O Performance Improvement for Multiple Applications
(mentioned in last PI meeting)

Optimizations use:

- Synchronous I/O on leading HPC systems (Titan, Mira, Edison, Tiahne-1A, Garnet, …)
- Techniques are to optimize in-memory data copy, movement, and file system access
- Optimizations for GPFS, Lustre
I/O Frameworks Facilitate Embedding *in situ* Data Services

Example: In-Memory Data Management for Coupled Simulation Workflows using Staging with DataSpaces and ADIOS

The need – enable *in situ* processing and analysis

- Staging as a persistent service – applications can dynamically connect to use staging / staging hosted services
- Enable more complex and dynamic workflows
- Many use cases: I/O acceleration, data staging, coupling, data reorganization/manipulation, indexing/querying, data persistence, etc.

Code Coupling: get, put, publish, subscribe service

In transit data movement and workflow management
In-Memory Data Management for Coupled Simulation Workflows using Staging with DataSpaces and ADIOS

**Fusion Application** (C.S. Chang)
- XGC1 and XGCa executed sequentially and share data
- In each iteration, the workflow first executes XGC1 for $n$ time steps to compute turbulence data and particle state, and then executes XGCa for $m$ time steps to evolve the state of plasma.
- XGC1 and XGCa use node-local shared memory segments as part of an in-memory data staging service

**Impact of using DataSpaces**
- Enables tightly coupled simulations at very large scales use node-local shared memory segments as part of an in-memory data staging service
- Results in huge performance improvement over traditional file-based approaches

Particle data read: reduced by $\sim 98\%$ compared to file-based
GLEAN: Topology-Aware Parallel I/O

- I/O performance is highly dependent on configuration settings
  - Need to perform the right computation at the right place and time taking into account the characteristics of the simulation, resources and analysis
  - e.g. HACC code often uses traditional I/O stack for checkpointing
- Using **GLEAN** Small code tweaks plus configuration adjustments led to **15x** performance improvement for checkpoint operations.
- Achieved **160 GB/s** for HACC production simulations on Mira BG/Q system
Large-scale Parallel Analysis of Applications with the DIY Library

DIY – Do It Yourself

- Large-scale parallel analysis (visual and numerical) on HPC machines
- For scientists, visualization researchers, tool builders
- *In situ*, co-processing, post-processing
- Data-parallel problem decomposition
- MPI + threads hybrid parallelism
- Scalable data movement algorithms
- Runs on Unix-like platforms, from laptop to supercomputer (including all IBM and Cray HPC leadership machines)
Examples of using DIY

**IMAGE SEGMENTATION IN POROUS MEDIA**

Left: 3D image of a granular material (flexible sandstone) acquired at ALS (by Michael Manga and Dula Parkinson). Data: 8.4 billion points: $2560 \times 2560 \times 1276$.

LBNL (Dmitriy Morozov and Patrick O’Neil) developed tools for segmentation and connectivity analysis of granular and porous media using DIY.

Right: segmentation of the material identifies individual grains.

**TESSELLATION-BASED DENSITY ESTIMATION COSMOLOGY AND ASTROPHYSICS**

Three representations of the same halo. From left to right: original raw particle data, Voronoi tessellation, and regular grid density sampling.

Left and center: Cross sections of CIC (cloud-in-cell) and much more accurate TESS-based density estimators. Right: Strong scaling for grid sizes ranging from $1024^3$ to $8192^3$ (0.5 trillion).
Science Problem often Require Specialized Data Management Techniques

Problem: detection of ice-calving events

- Large-scale calving events (free-floating icebergs and ice fracture) are of scientific interest to study global climate change
- Typical iceberg is 6x the size of Manhattan
- Scientific simulations are developed as part of the BISICLES partnership (Dan Martin)
- Simulation is AMR-based ice sheet modeling using Chombo

Goal

- Identify events during the simulation runs -> in situ/real-time for every time step

Technical approach

- Develop Parallel Connected Component Labeling (PCCL) for in situ AMR data
- Approach

Result

- Great speedup (up to 6x) at larger per-process data sizes relative to previously used multi-phase CCL (repeated scans)
SDAV Future Outlook

- **Experience with *in situ* processing has been very successful**
  - Typically less than 1% of simulation time used for *in situ* feature extraction
  - In-memory code-coupling, in-memory workflow management, and *in situ* indexing and search capabilities help minimize data movement
  - *In situ* visualization and analysis methods allow dynamic exploration of data and avoid writing full-resolution data
  - **We expect to continue and benefit from this approach**

- **Scalability has been successful**
  - Many SDAV tools scaled to hundreds of thousands of cores
  - Many SDAV tools took advantage of GPUs when available, especially Vis tools
  - Many tools managed to reduce their memory use
  - **We see a clear path for taking advantage of future architectural characteristics**

- **Burst buffers may be useful for**
  - Checkpoint restart
  - Fast read - staging of data
  - Extension of memory
  - Extension of storage
  - **We expect to benefit greatly from burst buffers in the SciDAC3 time frame**
SDAV Future Outlook

- **Hardware Gap**: generally high concurrency platforms are mostly designed with other target applications (e.g. simulation-> high flops, low memory)

- **Algorithmic Gap**: generally data analysis algorithms have been traditionally executed in post processing and therefore less research has been focused on extreme parallelism

- New algorithmic research needed:
  - New data structures that reduce memory usage
  - New algorithmic structure that better exploits GPUs
  - New algorithmic models that minimize communications
  - How to exploit opportunities given by deep memory hierarchies?
  - How can we better coordinate with I/O infrastructures?
Extra slides
Data Management Tools

- **I/O Frameworks**
  - **ADIOS**: provides a simple, flexible way for scientists to describe the data in their code, and based on that to provide efficient I/O, and in situ data processing
  - **Darshan**: captures an accurate picture of application I/O behavior
  - **Parallel netCDF**: a library providing high-performance I/O while still maintaining file-format compatibility with Unidata's NetCDF
  - **ROMIO**: is a high-performance, portable implementation of MPI-IO
  - **ViSUS/IDX**: Provides data streaming techniques for progressive processing and visualization of large surface and volume meshes

- **In Situ Processing**
  - **GLEAN**: is an extensible framework that takes system characteristics into account in order to facilitate simulation-time data analysis and I/O acceleration
  - **DIY**: Provides scalable building blocks for data movement tailored to the needs of large-scale parallel analysis workloads
  - **DataSpaces**: facilitates in situ code coupling using a shared-space abstraction
  - **EvPath**: is an event transport middleware layer providing processing over virtual data paths

- **Indexing / Compression**
  - **FastBit**: A very fast indexing method based on compressed bitmap representation specially suitable for scientific data
  - **ISABELA**: a tools for lossy but highly accurate (> .99 correlation) compression of spatio-temporal scientific data
Analysis and Visualization tools

• Analysis and Visualization Frameworks
  • VisIt: is an Open Source, interactive, scalable, visualization, animation and analysis tool.
  • ParaView: is an open-source, multi-platform data analysis and visualization application.

• Analysis and Visualization Libraries
  • ExMage: provides in situ pathtube generation and visualization.
  • TALASS: is a collection of routines for parallel and distributed processing of particle data.
  • Ultravis-P: is a collection of routines for parallel and distributed processing of particle data.
  • MSCEER: is a set of tools and libraries for feature extraction and exploration in scalar fields.
  • IceT: is a high-performance sort-last parallel rendering library that provides the unique ability to generate images for tiled displays.
  • NDDAV: is an interactive analysis framework for high-dimensional data.
  • VTK: is an open-source system for 3D computer graphics, image processing and visualization.
Analysis and Visualization tools

- **Multi-/Many-core Visualization Libraries**
  - **Dax**: The Dax Toolkit supports the fine-grained concurrency for data analysis and visualization algorithms required to drive exascale computing.
  - **EAVL**: is the Extreme-scale Analysis and Visualization Library that expands traditional data models to support current and forthcoming scientific data sets.
  - **PISTON**: is a cross-platform software library providing frequently used operations for scientific visualization and analysis.

- **Statistics and Data Mining**
  - **NU-Minebench**: is a data mining benchmark suite containing a mix of several representative data mining applications from different application domains.
  - **Importance-Driven Analysis**: is a tool that uses a newly-designed spatial data structure, named parallel distance tree, to enable highly scalable parallel distance field computing.