Exploration and Visualization of Oil Reservoir Simulation Data

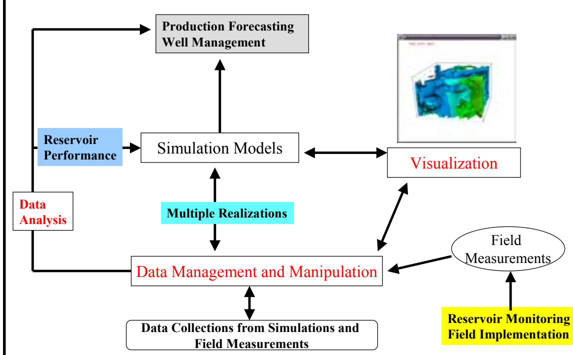
Joel Saltz, Umit Catalyurek, Tahsin Kurc
Biomedical Informatics Department
The Ohio State University
http://medicine.osu.edu/informatics

Mary Wheeler, Steven Bryant, Malgorzata Peszynska, Ryan Martino
Center for Subsurface Modeling
University of Texas at Austin
http://www.ticam.utexas.edu/CSM

Alan Sussman, Michael Beynon
Department of Computer Science
University of Maryland
http://www.cs.umd.edu/projects/adr

Don Stredney, Dennis Sessanna
Interface Laboratory
The Ohio Supercomputer Center
http://www.osc.edu

Economic Modeling and Well Management



Motivation and Challenges

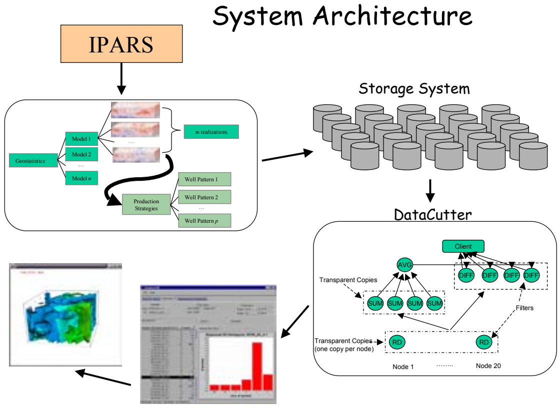
- Implementing effective oil and gas production
  - Optimizing well placement
  - Efficient exploration of possible production strategies
- Challenges: Geologic uncertainty, operational flexibility, and large, detailed flow models
- Simula… geosta… …gies
- Evalua… strate…
- Enable… …on of multiple scenarios
  - Integration of a robust, Grid-based computational and data handling infrastructure

Small scale can be done
Large scale exploration is not trivial
Remote exploration capability
Large storage space

Approach

- Combine leading-edge computational tools
  - **IPARS** for reservoir simulation
  - **DataCutter/ADR** for terascale data management/interrogation
  - **DISCOVER** for collaborative/interactive simulation
- Evaluate geologic uncertainty and production strategies simultaneously
  - Multiple realizations of multiple geostatistical models
  - Multiple production strategies (number, location of wells)

System Architecture



IPARS: Integrated Parallel Accurate Reservoir Simulator

- 8 individual physical models / algorithms for multiphase flow and transport
- Implemented in a common framework providing
  - memory management for general geometry grids
  - linear solvers with state-of-the-art preconditioners
  - portable parallel communication
  - keyword input and output with visualization
  - "hooks" for well management and other reservoir processes
- Code is portable across several serial and parallel platforms including Linux (clusters), SGI, RS6000, T3E, Windows (DOS)

System Support for Exploration of Large Datasets

DataCutter: A suite of Middleware for subsetting and filtering multi-dimensional datasets stored in a distributed environment

- Indexing Service
  - Multilevel hierarchical indexes based on spatial indexing methods – e.g., R-trees
- Filtering Service
  - Distributed C++ component framework
  - Specialized components for processing data
  - filters – logical unit of computation, high level tasks,
    - **init**,**process**,**finalize** interface
  - streams – how filters communicate
    - unidirectional buffer pipes
    - uses fixed size buffers (min, good)
  - manually specify filter connectivity and filter-level characteristics
- Active Data Repository (ADR): C++ class library and runtime system for building parallel databases of multi-dimensional datasets

Dataset

- Data size = ~1.5TB
- 207 simulations, selected from
  - 18 Geostatistics Models (GM)
  - 10 Realizations of each model (R)
  - 4 Well Patterns (WP)
- Each simulation is ~6.9GB
  - 10,000 time steps
  - 9,000 grid elements
  - 8 scalars + 3 vectors = 17 variables
- Stored on UMD Storage Cluster
  - 9TB disks on 50 nodes: PIII-650, 128MB, Switched Ethernet

Economic Model

- Economic assessment
  - Net Present Value (NPV)
  - Return on Investment (ROI)
  - Sweep Efficiency (SE)
- Queries
  - return R-WP for given GM that has NPV > avg
  - return R-WP for all GM which has max NPV
  - ….

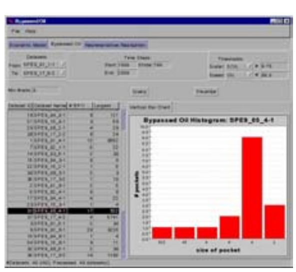
Economic model uses
- well rates (time series data)
- cost and price (e.g., oil) parameters

Results

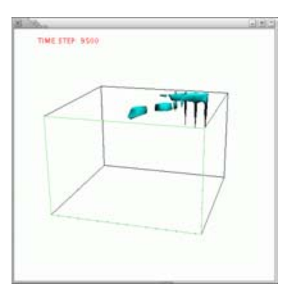
- Economic model shows range of winners and losers
- We want to also understand the physics behind this
- An example is looking at bypassed oil
  - Turns out to be strongly correlated to economics

Bypassed Oil (BPO)

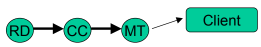
- User selects
  - a subset of datasets (D) and time steps (TS),
  - thresholds for oil saturation value (T_{os}) and oil velocity (T_{ov})
  - minimum number of connected grid cells (T_{cc}).
- Query: Find all the datasets in D that have bypassed oil pockets with at least T_{cc} grid cells.
- A cell (C) is a potential bypassed oil cell if $C_{os} > T_{os}$ and $C_{ov} < T_{ov}$.
- Algorithm for bypassed oil
  1. Find all potential bypassed oil cells in a dataset at time step T_i
  2. Run connected components analysis; Discard pockets with fewer cells than T_{cc}; Mark a cell if in a bypassed oil pocket, 0 otherwise.
  3. Perform an AND operation on all cells over all time steps in TS.
  4. Perform the operations in Step 2 on the result, and report back to client.



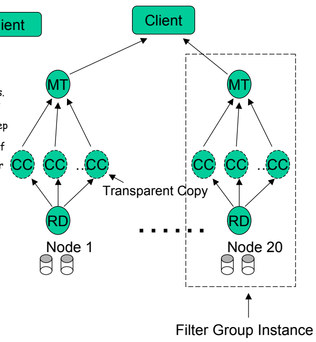
Bypassed Oil



BPO using DataCutter



- RD -- Read data filter. Accesses data sets. Each time step is a data buffer, which contains oil velocity and oil saturation values.
- CC -- Connected component filter. Performs steps 1 and 2 of the bypassed oil algorithm.
  - find bypassed oil pockets at a time step on data buffer received from RD.
  - send a byte array to MT. Each entry of the byte array denotes a grid cell and stores if the cell is bypassed oil cell or not.
- MT – Merge over time. Carries out steps 3 and 4 of the bypassed oil algorithm.
  - AND the data buffers received from CC, and
  - find bypassed oil pockets and send results to the Client.

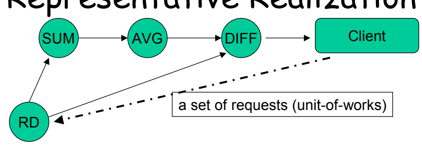


Representative Realization

- Select the simulation/realization that has values closest to a user-defined criteria.
  - analyze that simulation or use its initial conditions for further simulation studies.
- Find the dataset among a set of datasets
  - values of oil concentration, water pressure, and gas pressure are closest to the average of these values across the set of datasets
- User selects
  - A set of datasets (D) and a set of time steps ($T_1, T_2, \ldots, T_N$).
- Query: Find the dataset that is closest to the average.

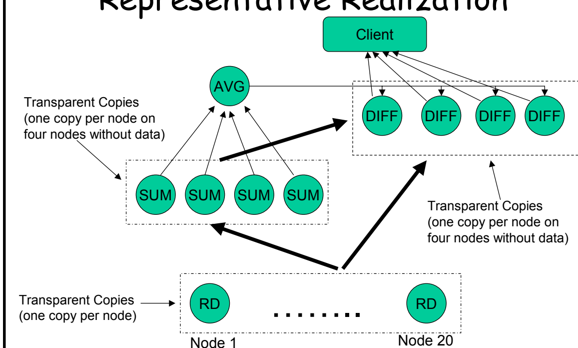
$$\min \sum_{(\text{all grid points})} | O_c - O_{cavg} | + | W_p - W_{pavg} | + | G_p - G_{pavg} |$$

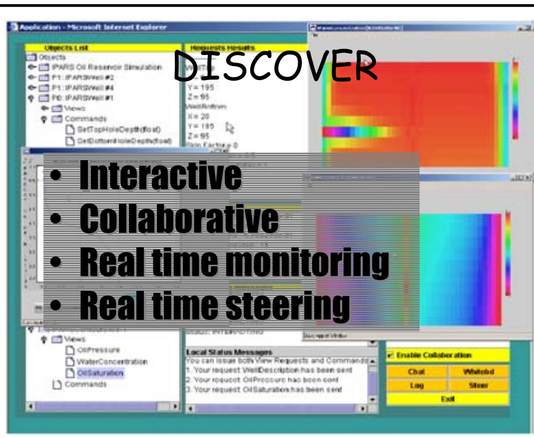
Representative Realization



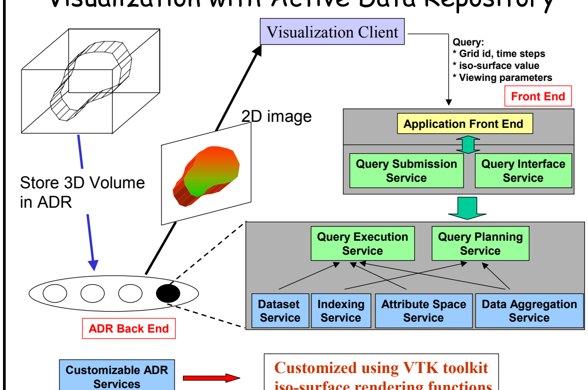
- RD – Read filter. Accesses data sets. A data buffer is one time step. Read filter sends data from each dataset to SUM and DIFF.
- SUM – Sum filter. Performs sum of C_o, W_p, and G_p at each grid point across datasets.
- AVG – Average filter. Carries out average operation on C_o, W_p, and G_p values. AVG and SUM together execute step 1 of the average algorithm.
- DIFF – Difference filter. Finds the sum of differences between grid values and average values for each dataset (Step 2). Sends the difference to the Client.
- Client – Keeps track of differences for each time step, carries out average over all time steps for each dataset (Step 3). Note this could be another filter.

Representative Realization

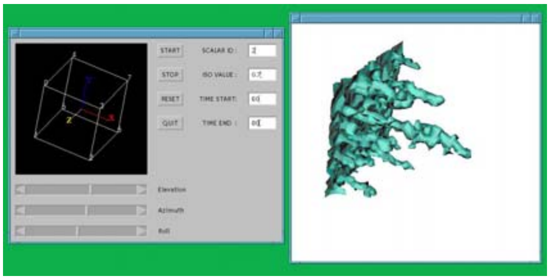




Visualization with Active Data Repository



Visualization with Active Data Repository



Iso-surface rendering of output (e.g., oil saturation)

Experimental Results

Query Type	# of Realizations	Time Steps (start:end:stride)	Execution Time (secs)
ECO	206	0:10000:1	136.97
BPO	206	1000:9000:1000	1.07
RR	206	1000:9000:1000	12.37

Economic model (ECO) uses ASCII text files, whereas for BPO and RR input files are binary files. For ECO most of the time is spent in parsing text files