Sandia National Laboratories

Exceptional service in the national interest

HPC AND DATABASES REVISITED

Jay Lofstead

Center for Computing Research, Scalable Systems Software

22 April 2024, CHEOPS@EuroSys

SAND2024-04829C

U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SETTING THE STAGE

- NetCDF3 (Classic) introduced in 1990
  - 32-bit until 2005 with version 3.6.0 which introduced 64-bit support
  - Parallel API access support introduced pNetCDF (2005) and NetCDF4 (2008)

- HDF5 introduced late 1990s
  - including the table feature from HDF4
    - B-tree indexing makes it work like an RDBMS

With database technology well established and the need to accelerate and expand data access options, it was the right time to look into an RDBMS

LATE 1990S, EARLY 2000S

ANL (Rob Ross, Rajeev Thakur, et al.), Northwestern (Alok Choudhary), and a few others took up the challenge

Result:

1. Array access critical, but databases largely tables of rows with row-wise access.
2. Databases required software install that was not a good fit for HPC batch systems
3. Overheads for a element per cell far too high
4. 3-D domain storage explodes overheads compared to 2-D
    1. SciDB columnar format doesn't extend well to 3-D data

Very reasonable conclusion: Not a good fit

IO CHALLENGES IN THE 2000S

- Switch to multi-core and growing importance of 3-D domains and 3-D domain decompositions
- Physically contiguous data format requires massive data rearrangement
  - 512 procs can be 90%+ of IO time [Yu, Vetter: 2008]
- Disk still predominant making seek and rotational latency critical considerations

- Tuning MPI-IO a dark art for non-experts
- Tuning PFS configuration rarely done

THE NEXT WAVE

- SQLite released in 2000
  - Moved to binary numerical storage in 2004 replacing all strings
  - Hit 100% branch test coverage in 2009
  - PostgreSQL work-alike
  - Started strong in embedded market including cell phones
- SciDB released in 2008
  - Columnar database
- ADIOS 1.0 released in 2010
  - Log-structured, block-based data organization
- Apache Arrow released in 2016

LATE 2000S INTO 2010S

- ADIOS developed
  - Shifted from single physically contiguous data layout to blocked
  - Used log-structured format
  - Separated API from implementation to enable more optimization
    - (MPI-IO is arguably a similar implementation)
  - Included “indexes” for faster access

Problems not addressed:

1. Attributes stored as a list requiring O(n) search time to find anything
2. “Index” sizes limited to DRAM in a single node

PROBLEMS WITH MAJOR LIBRARIES

- Single physically contiguous linearization of 3-D variable too slow for all but fast dimension
- Linear searching to find items is a terrible choice (EMPRESS 500x faster than HDF5 searching for attributes)
- Memory capacity limitations problematic
- Avoiding data corruption during crashes requires consistency controls (e.g., transactions)

- But real database engines are inadequate….

2020S

- DOE ASCR Data Management funding round
  - How do we manage the data volumes to help application scientists do their job?
  - Coeus
  - RECUP
  - SoMeta/Proactive Data Containers
- ADIOS2
  - Streaming model, but plug in engines for various processing steps
- HDF VOL separating API from implementation
- DAOS phase 2 (the product rather than research prototype)

Lots of work to try to help with data access. Why not leverage the database community work?

2010S TO TODAY

- D^2T – doubly distributed transactions used SQLite for managing state for HPC
- EMPRESS metadata management system developed

- Both demonstrated embedded RDBMSes both work well and were potentially viable
- We could potentially get the Database community data management/searching/indexing advances with minimal recoding for special formats!

- Well, almost

- Stitch-IO leaned in using SQLite as the data format….

NOT PROBLEM FREE

- Stitch-IO chose SQLite to be lazy
  - No need to write a custom data layout
  - Selecting and ordering blocks that intersect a requested sub-region at time evolution is easy to write in SQL
  - Copying into a buffer just takes a one-by-one test and copy

- Problems!
  - SQLite uses POSIX system locks for write locks
    - Lustre does not support proper POSIX locking; GPFS does; others ?
    - Must manually serialize write access limiting scalability (token passing or work queue for a single process)

AND MORE PROBLEMS!

- Database page size defaults to 4KB and number of pages are limited to 2^{31}
  - Working on a problem with a 6 TB base 3-D domain size blew out the default settings
  - Bonus! These can be changed by adjusting the database settings
- Indexes must be properly created or performance is terrible
  - 26 minutes to read for the first timestep after initialization for 128 processes
  - Fixed the index and dropped a join to reduce to < 1 second
- Scale out with shared nothing, but then querying is harder (and more data management)

NOSQL TO THE RESCUE?

- Columnar databases, maybe
  - Fall 2018 project tried to install Cassandra on restricted network
  - Fixed columns are the only ones with an index
  - Searching for attributes by name probably not efficient still
  - SciDB not a great fit for 3-D data

- Key-value stores
  - EMPRESS investigation showed can only be efficiently accessed by designed key-structure

- Document stores (MongoDB)
  - We have mixed data types, sizes, and densities

RDBMS DEVELOPMENTS REDUX

- SQLite developed in early 2000s. SQLite3 is AMAZING!
  - Most used and most widely ported piece of software on the planet
  - Public domain code (more permissive than open source as there is no copyright)
  - Extensive regression testing infrastructure
  - Support for in memory only, on storage only, or write-back to storage usage
  - 300K of C code (1 .h and 1 .c file)
- SQLite is problematic
  - Designed for single user, embedded. Decisions were made
    - E.g., Locking could be more advanced (row level locking like a traditional, modern RDBMS)
  - No support for data partitioning

RDBMS DEVELOPMENTS

- DuckDB
  - The current database community darling
  - Inspired from SQLite, but focused on more advanced features/performance
    - E.g., Embeddable, simple API, good performance

- Container-based RDBMS deployments
  - PostgreSQL, MariaDB, and others

Progress, but still not quite the right fit. Big need:

Massive N-to-1 write to a “single” table

- Do single process write (create) and then fully parallel update?

WHAT DO WE NEED TO DO?

- Database/Big Data community handles massive ingest
  - E.g., Book: Big Data Application Architecture Q&A: A Problem – Solution Approach
  - One of the “V”s is velocity

- Row-level locking a feature since early 1990s (Ingress had it) at least
- Shift from per-element to blob-based storage
  - ADIOS demonstrated viability and performance
  - Stitch-IO shows can work at small-ish scale (up to 1K procs good, untested beyond that)

COMMUNITY ENGAGEMENT

- How do we get database community interested?
- How do we get past the “it doesn’t work for us” mindset?

- Goal: We want true best in class data management

QUESTIONS?

- Jay Lofstead
- gflofst@sandia.gov

REFERENCES

- SciDB: Relational daddy answers Google, Hadoop, NoSQL • The Register
  - https://www.theregister.com/2010/09/13/michael_stonebraker_interview?page=1
- Various Wikipedia articles (HDF, NetCDF, Apache Arrow, SciDB)
- UCAR website for NetCDF information
- HDF Group website for HDF information
- ADIOS information provided by the author
- Choudhary A, Kandemir M, No J, Memik G, Shen X, Liao WK, Nagesh H, More S, Taylor V, Thakur R, Stevens R. Data management for large-scale scientific computations in high performance distributed systems. Cluster Computing. 2000 Jul;3:45-60.
- Ross R, Thakur R, Choudhary A. Achievements and challenges for I/O in computational science. InJournal of Physics: Conference Series 2005 (Vol. 16, No. 1, p. 501). IOP Publishing.
- Yu W, Vetter J. Parcoll: Partitioned collective i/o on the cray xt. In2008 37th International Conference on Parallel Processing 2008 Sep 9 (pp. 562-569). IEEE.
- ADIOS - Accelerating HPC (adios-io.org)