GEN: A Database Interface Generator for HPC Programs

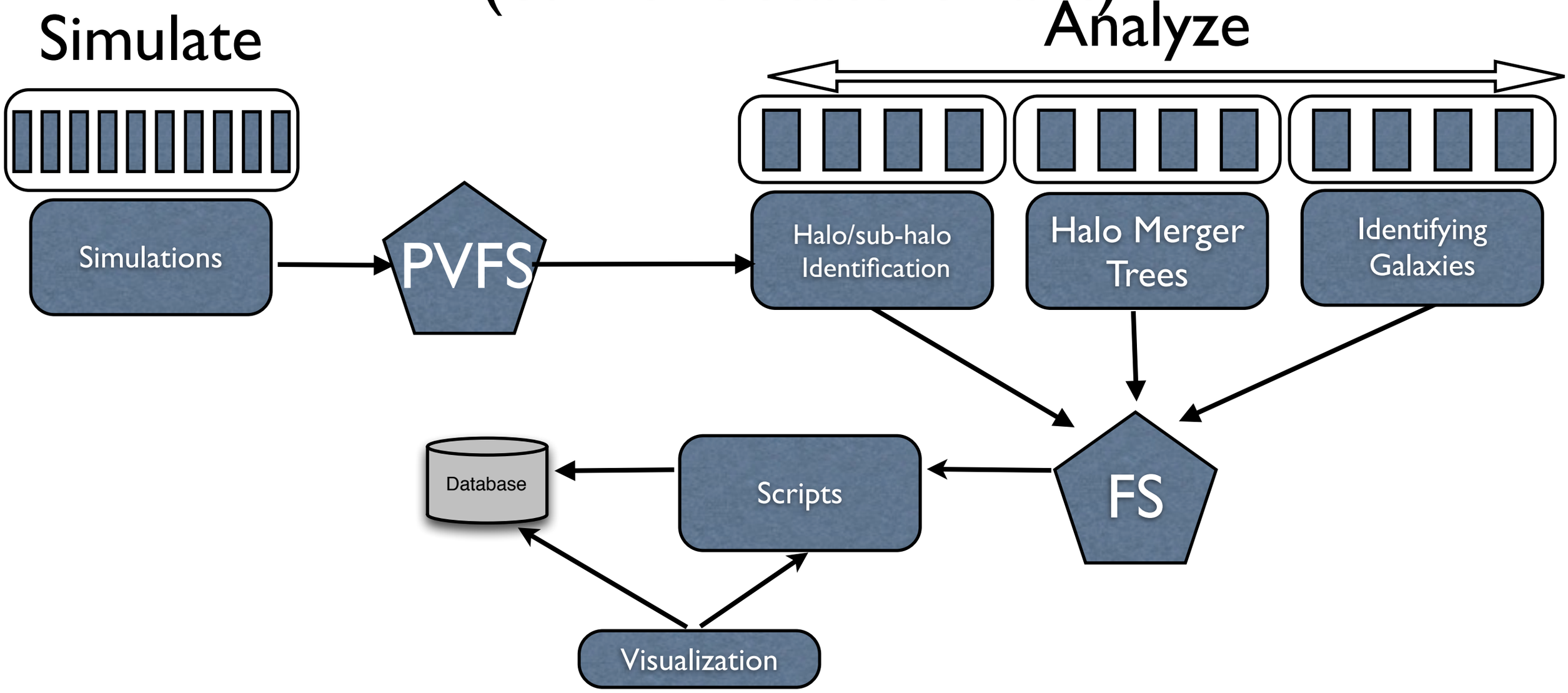
Quan Pham, Tanu Malik
Computation Institute
University of Chicago and Argonne National Laboratory
tanum@uchicago.edu

Computational Science Challenges

- Science is iterative and data-intensive
  - Simulate-Analyze-Simulate
- Supercomputing time is precious
  - So reduce time for data analysis
- Challenges
  - Slow IO and network bandwidth
- How to remove redundant and IO-intensive steps?

Computational Cosmology

(current state-of-art)

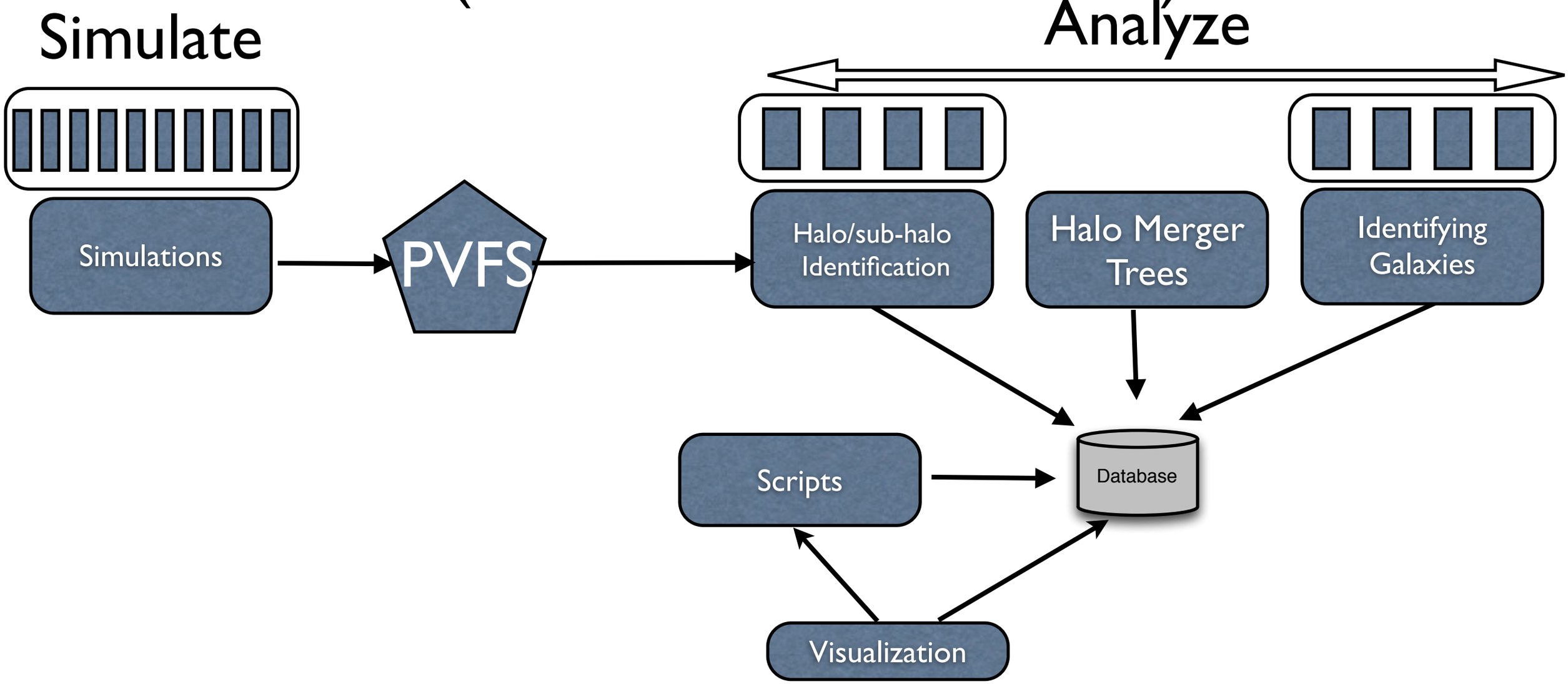


- Data sizes reduce as analysis proceeds
- Simulation size is few Petabytes
- Analysis data may range from TBs to 10s of GBs

Madduri, Malik, Habib, et. al. PDACS: A Portal for Data Analysis Services for Cosmological Simulations. In: ACM Computing In Science and Engineering, 2015

Computational Cosmology

(envisioned state-of-art)



- Database-friendly analysis within the DB
- Mechanisms for direct IO to databases

Related Work

- The SciHadoop System
  - Move data between file-systems (from PVFS to HDFS)
  - Simple sub-selection and aggregation queries
- Querying using HPC libraries
  - Transform sub-selection and aggregation SQL queries into parallel IO operations using parallel-NetCDF
  - Requires user to perform parallel data management
- Database-as-a-service
  - RESTful service approach good for small amount of data

GEN: A Database Interface Generator

- Takes user-supplied C declarations and provides an interface to load into and access data from common scientific array databases

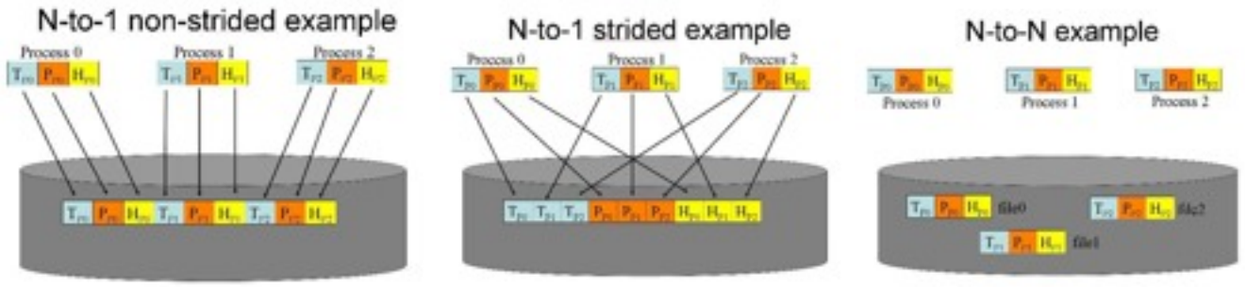
GEN Overview

```
MPI_Init ( &argc, &argv );
MPI_Comm_rank ( MPI_COMM_WORLD, &id );
MPI_Comm_size ( MPI_COMM_WORLD, &p );

if ( rank == 0 )
{
   x_file = fopen ( "x_data.txt", "w" );
for ( j = 0; j < n; j++ )
   {
      for ( i = 0; i < m; i++ )
        fprintf ( output, "  %24.16g", table[i+j*m] );

      fprintf ( output, "\n" );
   }
    fclose ( x_file );
}
```

```
MPI_Init ( &argc, &argv );
MPI_Comm_rank ( MPI_COMM_WORLD, &id );
MPI_Comm_size ( MPI_COMM_WORLD, &p );

if ( rank == 0 )
{
     CREATE TABLE "x_data.txt"
for ( j = 0; j < n; j++ )
{
  for ( i = 0; i < m; i++ )
    INSERT into "x_data.txt" VALUES (table[i+j*m])
  fprintf ( output, "\n" );
}
fclose ( x_file );
}
```

Issues

- IO patterns



  - N-1 non-strided or N-1 strided or N-N
- Scattered IO calls
- No direct mapping between POSIX calls and DB statements
- Direct C POSIX calls to DB statements
- Constraints:
  - No source code changes

GEN

- Assumptions
  - Serial IO for now
  - IO done contiguously within the context of a single function

GEN-1

- Map system calls to DDL statements without changing source code
  - LD_PRELOAD: load a shared library
  - Use LD_PRELOAD to force load a library that has *GEN's* implementation of fopen(), fwrite(), fread()

fopen("x_data.txt", w")

Create <Database Object> x_data.txt.tmp <structure>

GEN-11

- Schema-later; no structure initially, reorganize later

```
fopen("x_data.txt", w")

Create Table x_data.txt.tmp <i int; x double>;
Create Array x_data.txt.tmp <x:double> [i=0:*]
```

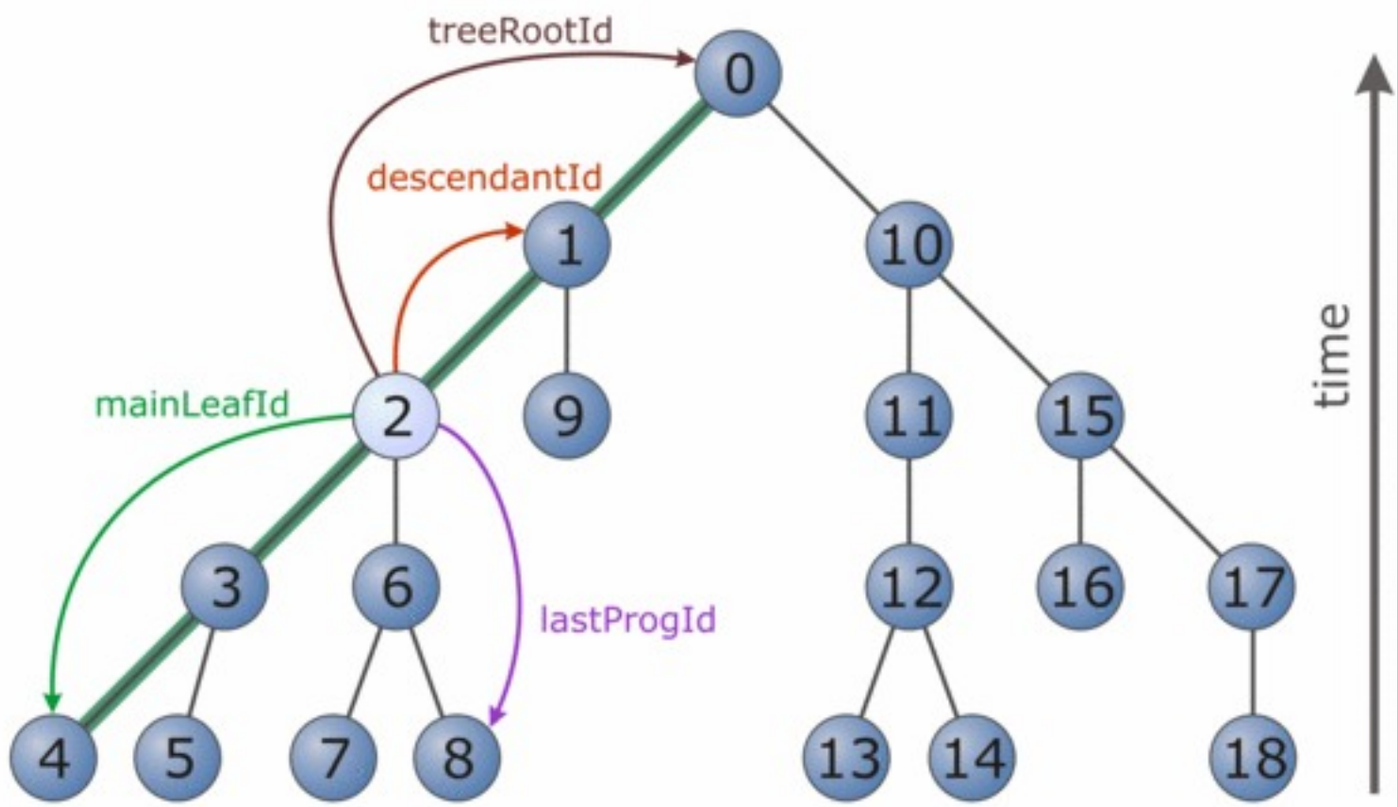
- OS-DB impedance mismatch
  - Not every fwrite() translates to INSERT statement
  - fwrites are buffered before being sent to filesystem
  - A single statement when data is being flushed

Knowing the Structure

- User-defined
- Header files
- Allow some system calls to create sample files

Experiments on Merger Trees

- Building merger-trees in DB system
  - Built from halo particles i.e. particles in a given halo
  - Several time-steps
  - Perform a particle merge-join between halos of timestep t_{i-1} and t_i
  - Lineage queries to answer ancestors and progenitors



- MPI program
  - Streams data to a single node, which streams to GEN
  - Postgres and SciDB interfaces

Experimental Setup

- Overhead of redirection
  - Dynamically-linked library may get off-loaded
    - 0.04 ms ± 0.02ms
- Time for reorganizing the data
  - Reorganizing is cheaper in the DB than outside

Table 1: Reorganization times

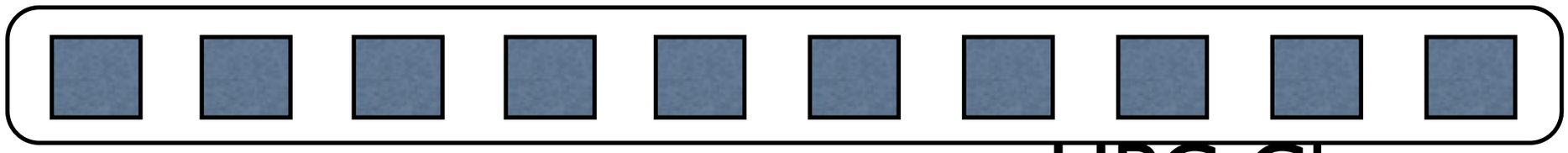
Data Size (in GB)	Postgres		SciDB	
	Load from Files	Reorganize within DB	Load from Files	Reorganize within DB
1	2.7	1.2	8.4	3.2
5	6.8	2.5	14.3	7.6
10	10.56	4.7	22.6	8.9

Conclusions and Current Work

- A database generator library for data-intensive scientific applications that requires no source code changes, which loads to be reorganized later.
- Relax the assumptions
- Loading is still an issue
- Source-code release

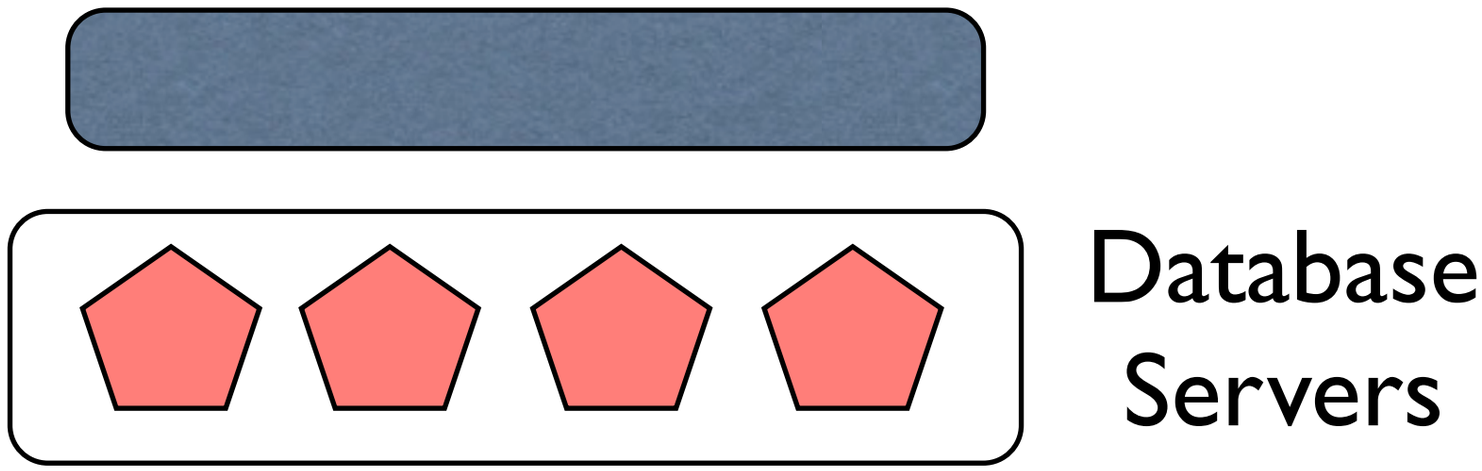
- Thank You!
- Acknowledgements
  - Salman Habib, Katrin Heitmann, Steve Rangel, Hal Finkel, Ravi Madduri

A vision



HPC Cluster

High-bandwidth Network



Database Servers