SciDB for Metagenome Analysis

Katherine Sittig-Boyd[1,3], Amrita Pati[2], Kristin Tennessen[2], Yushu Yao[3]
[1]Simmons College, [2]Joint Genome Institute, [3]National Energy Research Scientific Computing Center

> Abstract

Metagenomics is the study of genetic material gleaned from a community of organisms. The Joint Genome Institute [1] is constantly collecting and sequencing genomic data. Implementing SciDB, an open-source DBMS designed around multi-dimensional arrays, may be more optimal for large-scale data analysis. This research compares the performance of SciDB and the SQLite file system on metagenome data querying.

> Background and Purpose

Implementing SciDB [3] may allow for more optimal query time when applied to large datasets.

Research Goals

- Compare SciDB and SQLite, the JGI's current file system
- Determine comparative querying speed

> Data Set

- Researchers at the JGI are interested in how genes map to protein families ("pfams")
- 297M genes in 1,500 metagenomes

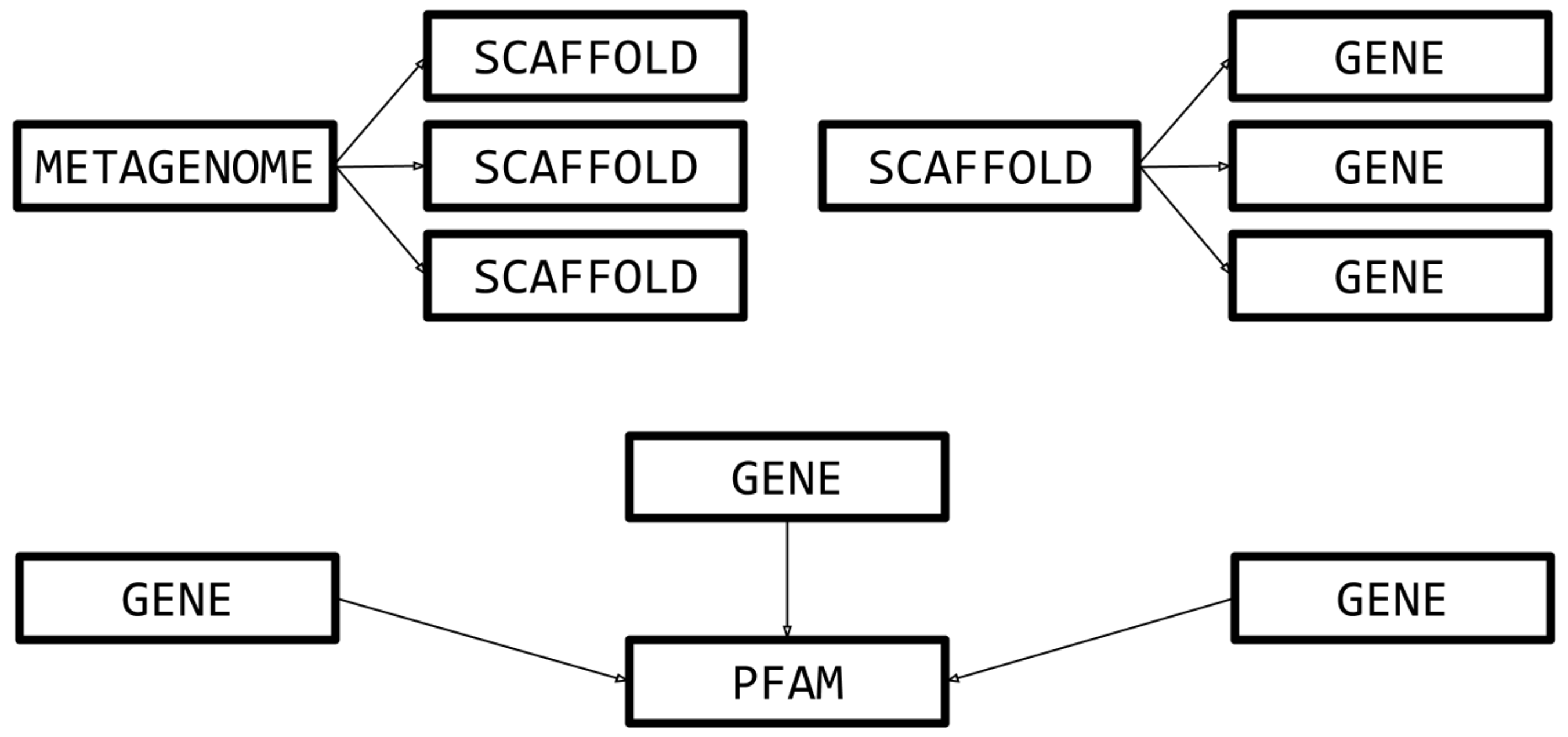

Fig. 1 The relationship between a metagenome, scaffold, gene and pfam

> SciDB Implementation

The SciDB query times were tested using NERSC's [2] Jesup testbed. This version of SciDB is 13.12

- 16-node testbed cluster

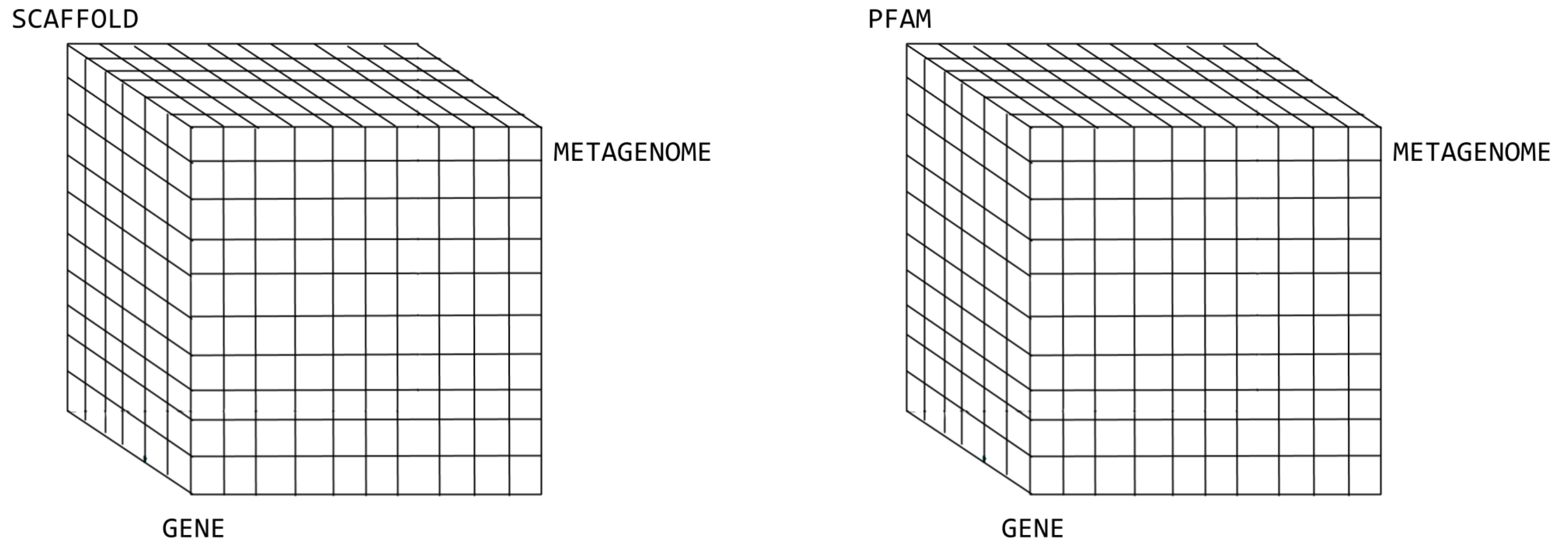

Fig. 2: Visualizations of the 3D arrays in SciDB

> SQLite Implementation

The SQLite file system is implemented using sqlite3 [4]. SQLite queries were executed using the Carver supercomputer's serial queue.

Design

- One database implemented for each metagenome consisting of two tables, GENE and GENE_PFAM

> The Query

1. Generate the list of all metagenomes that contain six given pfams on a single scaffold in individual metagenomes
2. Use the results of 1) and identify sets of genes annotated with pfams in the pfam group located consecutively on a metagenome scaffold

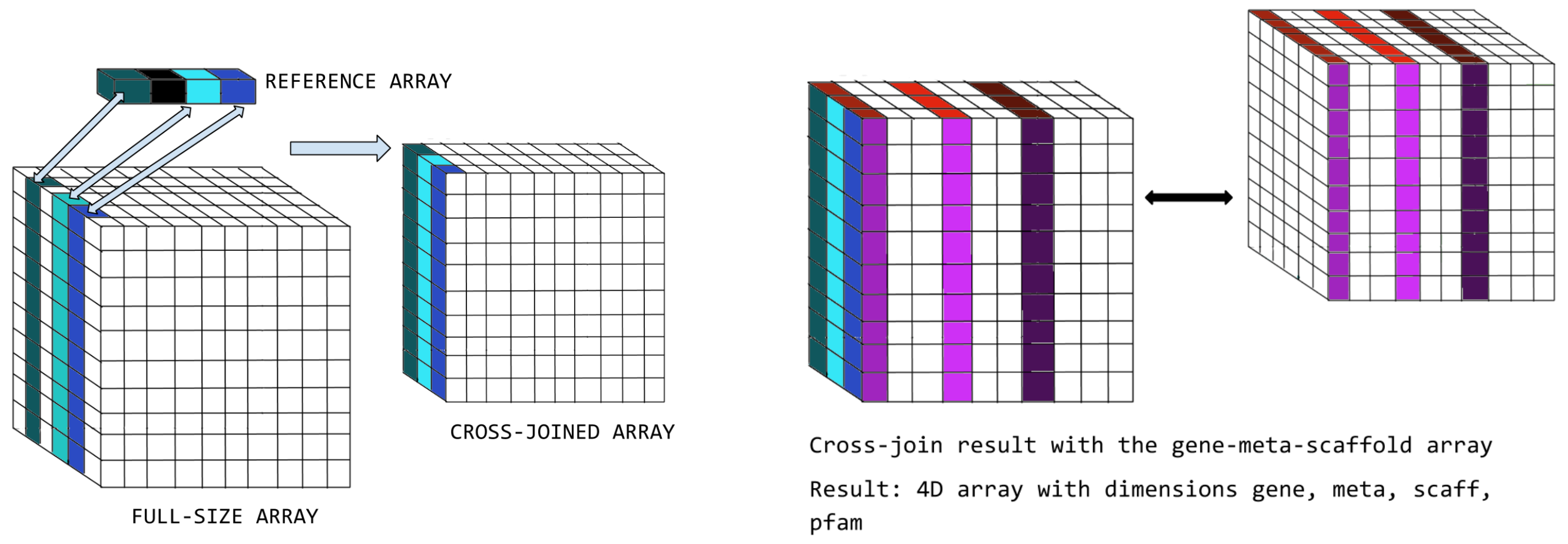
REFERENCE ARRAY
CROSS-JOINED ARRAY
FULL-SIZE ARRAY
Cross-join result with the gene-meta-scaffold array
Result: 4D array with dimensions gene, meta, scaff, pfam

Fig. 3: The cross-joining process

> Results

SciDB

- Ran query 5 times
- Fastest time: **8.60 seconds**

SQLite

- Ran query 5 times
- Fastest time*: **12.90 seconds**

*The SQLite runtime is theoretical to account for the difference in threading (1 thread for SQLite, 32 threads for SciDB).

> Discussion

Assuming both methods are optimized:

- SciDB query time: **30% faster on average**
- Significant optimization effort to make SQLite query run with 32 threads in parallel, I/O contention

> Bibliography

References

[1] Joint Genome Institute website. http://jgi.doe.gov/
[2] NERSC website. http://www.nersc.gov
[3] SciDB website. http://www.scidb.org
[4] SQLite website. http://www.sqlite.org/

> Acknowledgments

This work was supported in part by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422), with additional assistance from the LBL and DOE.