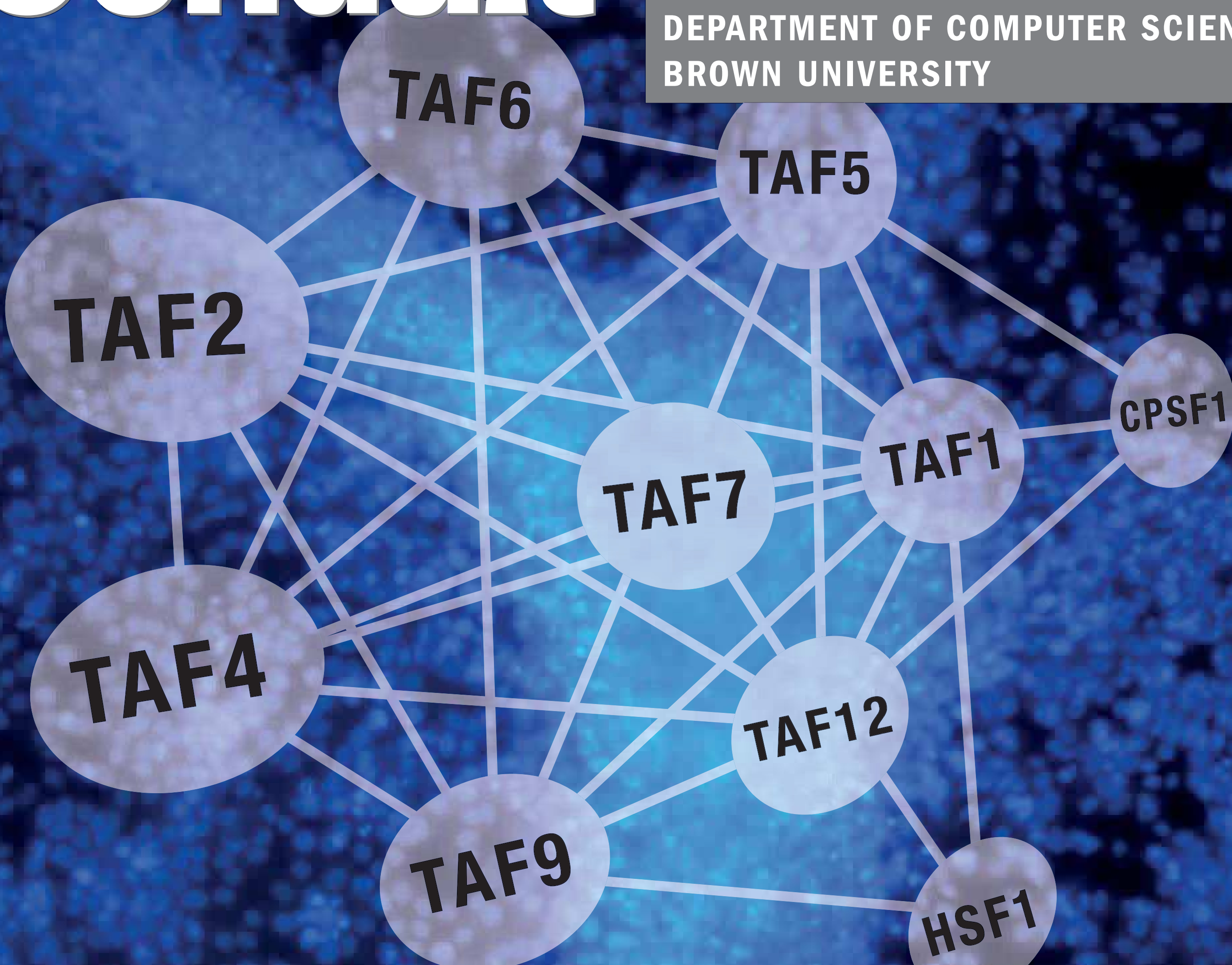
Conduit

A RESEARCH AND
ALUMNI NEWS MAGAZINE
DEPARTMENT OF COMPUTER SCIENCE
BROWN UNIVERSITY



HotNet is the heat-seeker of cancer

ALSO INSIDE:

- Robot Lab Unpacks New Partner
- 2011 Artemis Project

Next Generation Database Systems @ Brown CS

By: The Database Management Group [database.cs.brown.edu]

Starting with the requirements of business data processing, the design of database management systems (DBMSs) have been traditionally guided by the need to ensure consistency and persistence of structured, scalar data in the presence of short-lived data processing tasks with hard integrity requirements. While this design has served business applications well for more than two decades, many data-intensive applications that recently emerged have different goals and requirements, and are thus not satisfactorily supported by this prevailing database model.

The past decade has seen various proposals for new DBMSs designed to better meet the demands of various new application domains, such as web databases and scientific computing, while taking advantage of the advances in hardware platforms. The rest of the article discusses primary examples of these new DBMS breeds and how Brown researchers and alumni are contributing to these efforts.

NoSQL Systems

The advent of affordable, shared-nothing computing systems portends a new class of parallel DBMSs for web applications. The difficulty of scaling front-end applications is well known for workload-intensive DBMSs. One approach to this problem employed by many web-based companies is to shard the data and workload across a large number of commodity, shared nothing servers using a cost-effective, parallel DBMS. Many of these companies have adopted various new DBMSs, colloquially referred to as *NoSQL* systems, that give up certain guarantees provided by traditional databases (e.g., atomicity) in favor of availability and scalability. This approach is desirable if the consistency requirements of the data is "soft" (e.g., status updates on a social networking site that do not need to be immediately propagated throughout the application).

One notable NoSQL DBMS is MongoDB from 10Gen, a New York City-based start-up co-founded by Brown alum Eliot Horowitz'03. MongoDB is an open source, document-oriented (i.e., JSON) DBMS that is increasingly popular with both large and small companies, and is noted particularly for its ease of use and scalability. Notable users of MongoDB include Craigslist, The New York Times, FourSquare, and Shutterfly.

The performance of NoSQL systems like MongoDB is predicated on the existence of an optimal database design that is tailored for the unique characteristics of the workloads and cloud computing environments. Such a design defines how an application's data and workload is sharded across nodes in a cluster. The choice of one design over another determines the number of operations that can execute to completion on just a single node without needing to communicate with other nodes, as well as how evenly work is distributed across nodes. Hence, without a proper design, a NoSQL DBMS running on multiple nodes will perform no better than a single-node system.

The Brown Data Management Group are working with 10Gen to research new methods to improve the performance of MongoDB. The Brown alumni love fest includes 10Gen engineers Spencer Brody'10 and Dan Pasette'95 MS'96, while the student group includes graduate students Andrew Pavlo PhD'25 and Christopher Keith MS'13, and undergraduates Andrew Scheff'13 and Emanuel Buzek'13. This team is developing an automatic database design tool that analyzes an existing MySQL or MongoDB application, and then automatically generates an optimal configuration tailored to that application and improves the application's performance on MongoDB.

NewSQL Systems

Many enterprise systems that handle high-profile data (e.g., financial and order processing systems) also need to be able to scale but are unable to use NoSQL solutions because they cannot give up strong transactional and consistency requirements. The only options previously available for these organizations were to either purchase a more powerful single-node machine or develop custom middleware that distributes queries over traditional DBMS nodes. Both approaches are prohibitively expensive and thus are not an option for many.

As an alternative to NoSQL and custom deployments, a new emerging class of parallel DBMSs, called NewSQL, are designed to take advantage of the partitionable workloads of enterprise applications to achieve scalability without sacrificing ACID guarantees. The applications targeted by these *NewSQL* systems are characterized as having a large number of transactions that (i) are short-lived (i.e., no user stalls), (ii) touch a small subset of data using index look-ups (i.e., no full table scans or large distributed joins), and (iii) are repetitive (i.e., executing the same queries with different inputs). Such transactions in enterprise applications are also typically executed as pre-defined transaction templates or stored procedures in order to reduce DBMS overhead.

The first NewSQL system, called H-Store[hstore.cs.brown.edu], is being developed as a joint project between the Brown Data Management Group, MIT, and Yale. Under the direction of the Prof. Stan Zdonik and Andrew Pavlo PhD'25, Brown has been leading the development of the H-Store project for the last three years. The H-Store design is also being commercialized into the open-source database VoltDB.

Unlike a traditional database, such as MySQL or Oracle, H-Store stores all of its data in main memory across multiple machines, allowing it to achieve very high-throughput for transactional workloads. But exploiting this novel operating paradigm is non-trivial for many applications; certain information must be known before the DBMS begins executing some piece of work. Hence, the Brown group is developing new algorithms and techniques to overcome this barrier and make it easier for administrators to use NewSQL systems. Like a pack of dogs fighting over the contents of the dumpster behind Gordito Burrito, Zdonik and Pavlo's research has been relentless at improving H-Store's performance on cloud-based systems, such as Amazon EC2, through intelligent data partitioning and placement. With the help of undergraduate Charles Lee'12, the duo are also working optimizing H-Store to execute on a next generation "many-core" processor prototype provided to Brown from Intel.

Scientific Databases

Research and innovation in many parts of science have become a data-driven endeavour. As scientific data have gone from scarce to superabundant with data generation rates doubling yearly, scientists now spend increasingly more time and effort managing their data than doing real science.

A major part of the problem is the lack of re-usable, effective tools for data storage and processing. Scientific data management has traditionally been performed using low-level file-based solutions, at best using files structured according to a low-level data format. Higher-level data management infrastructures developed so far have been mostly task-specific and thus non-reusable in different domains, resulting in significant duplicated implementation effort by scientists.

In order to improve this poor state of affairs, several scientific DBMSs projects are currently underway to address the inherent limitations of conventional relational DBMSs when applied to scientific data and applications. One major effort is SciDB#, an open-source DBMS designed and optimized for scientific applications. The overarching goal of SciDB is to do for science what relational databases did for the business world, namely to provide a high performance, commercial quality and scalable DBMS appropriate for many science domains.

In contrast to a conventional DBMS that uses relations to represent its data, SciDB is based on a multidimensional array data model. In addition, SciDB includes multiple features specific to and critical for science: these include (i) automatic provenance tracking and querying, (ii) support for uncertain data storage and processing, (iii) automatic versioning, (iv) native support for science-specific operations (such as matrix multiplication), and (v) in situ data processing. Currently, no existing system offers these features in a single, highly scalable engine, requiring scientists to spend a lot of time and effort rolling their own solutions through low-level custom coding.

The Brown Data Management Group has been helping make SciDB a reality, in collaboration with colleagues from MIT, Portland State, University of Washington, and University of Wisconsin. This extended group has been working together for the last three years and recently received a large NSF grant to fund their research on scientific data management.

The Brown team's current focus is on providing support for interactive data analysis and exploration, especially in the presence of very large underlying data sets. To this end, the team has been working on a number of complementary research threads to simplify and speed up data-driven exploration, while enhancing the overall user experience through personalization and automation.

One such thread involves user and workload modeling: PhD student Justin DeBrabant (PhD'16) has been applying learning and mining algorithms to build predictive user-database interaction models that facilitate query recommendations and optimizations such as pre-fetching and pre-computation. Alex Kalinin PhD'16, another graduate student, has been working on goal-oriented data exploration interfaces to allow users to characterize the type of answer they are looking for without precise query specifications. On a related effort, undergraduate Joe Shapiro'13 spent last summer developing a visual front-end for SciDB, which highlighted a number of open research challenges regarding the visual display and manipulation of large data sets. On the automation front, PhD student Jennie Duggan PhD'12 has been developing advanced techniques for for automatically finding the best data representation and compression formats for large evolving scientific data sets.