Dixin Tang

Address	5730 S Ellis Ave, JCL 299 Chicago, IL 60637	**Homepage** **Email**	`people.cs.uchicago.edu/~totemtang` totemtang@uchicago.edu

Research Areas

Query Processing/Optimization, Cloud Database, Transaction Processing

Education

2015-present	Ph.D. Candidate in Computer Science - The University of Chicago
	Advisor: Aaron Elmore
2011-2014	M.S. in Computer Science - Institute of Computing Technology, Chinese Academy of Sciences
	Advisor: Wei Li
2007-2011	B.S. in Software Engineering - Huazhong University of Science & Technology

Publications at UChicago

- CrocodileDB in Action: Resource-Efficient Query Execution by Exploiting Time Slackness
  Dixin Tang, Zechao Shang, Aaron J. Elmore, Sanjay Krishnan, Michael J. Franklin
  VLDB 2020, Demo

- Thrifty Query Execution via Incrementability
  Dixin Tang, Zechao Shang, Aaron J. Elmore, Sanjay Krishnan, Michael J. Franklin
  SIGMOD 2020

- CrocodileDB: Efficient Database Execution through Intelligent Deferment
  Zechao Shang, Xi Liang, **Dixin Tang**, Cong Ding, Aaron J. Elmore, Sanjay Krishnan, Michael J. Franklin
  CIDR 2020

- Intermittent Query Processing
  Dixin Tang, Zechao Shang, Aaron J. Elmore, Sanjay Krishnan, Michael J. Franklin
  VLDB 2019

- Socrates: The New SQL Server in the Cloud
  Panagiotis Antonopoulos, Alex Budovski, Cristian Diaconu, Alejandro Hernandez Saenz, Jack Hu, Hanuma Kodavalla, Donald Kossmann, Umar Farooq Minhas, Naveen Prakash, Hugh Qu, Chaitanya Sreenivas Ravella, Krystyna Reisteter, Sheetal Shroti, **Dixin Tang**, Vikram Wakade
  SIGMOD 2019

- Toward Coordination-free and Reconfigurable Mixed Concurrency Control
  Dixin Tang, Aaron J. Elmore
  USENIX'ATC 2018

- Adaptive Concurrency Control: Despite the Looking Glass, One Concurrency Control Does Not Fit All
  Dixin Tang, Hao Jiang, Aaron J. Elmore
  CIDR 2017

Earlier Publications

- A Case Study of Optimizing Big Data Analytical Stacks Using Structured Data Shuffling
  Dixin Tang, Taoying Liu, Rubao Lee, Hong Liu, Wei Li
  BigData Congress 2016

- SparkArray: An Array-Based Scientific Data Management System Built on Apache Spark
  Wenjuan Wang, Taoying Liu, **Dixin Tang**, Hong Liu, Wei Li, Rubao Lee
  NAS 2016

- A Case Study of Optimizing Big Data Analytical Stacks Using Structured Data Shuffling
  Dixin Tang, Taoying Liu, Rubao Lee, Hong Liu, Wei Li
  CLUSTER 2015, Short Paper

- RHJoin: A Fast and Space-efficient Join Method for Log Processing in MapReduce
  Dixin Tang, Taoying Liu, Hong Liu, Wei Li
  ICPADS 2014

- Optimizing the Join Operation on Hive to Accelerate Cross-Matching in Astronomy
  Liang Li, **Dixin Tang**, Taoying Liu, Hong Liu, Wei Li, Chenzhou Cui
  IPDPS Workshops 2014

Industrial Experience

- **Internship at Microsoft Research** June 2018-Sep. 2018
  Project: Benchmarking Socrates Mentor: Umar Farooq Minhas

  Socrates is a new cloud-native database that decouples computation from storage. My internship job is to test the new database architecture of Socrates in the industrial setting, understand its performance bottlenecks, and propose optimization opportunities.

Research Projects at UChicago

- **CrocodileDB: Resource-efficient Database Execution** July 2019-Present

  CrocodileDB is a new database for processing dynamic datasets (e.g. streaming tuples). It exploits intelligent deferment to achieve low resource consumption and high query performance at the same time. In CrocodileDB, users can trade-off between resource consumption and query performance by specifying a maximally allowed time slackness between the data is ready and the corresponding query result is returned. Our system integrates the slackness constraints along with information about new data and query structures into the underlying query optimizer to unlock new resource-efficient query execution plans.

- **Incrementability-aware Query Processing** Mar. 2019-Present

  This project studies how to efficiently maintain non-positive standing queries. While incremental executions can reduce query latency, it increases total query work (i.e. CPU cycles) when we eagerly maintain non-positive queries because output tuples in earlier executions are repeatedly removed by later executions. To reduce the total work of incremental executions without sacrificing query latency, we selectively defer some parts of a query that are less amenable to incremental execution. To enable this optimization, we define a metric, *incrementability* that quantifies the cost-effectiveness of incremental executions. Our system can leverage this metric to decide how eagerly and lazily we should execute different parts of a query.

- **Intermittent Query Processing** Dec. 2017-Sep. 2019

  This project studies how to maintain a standing query over an incomplete data set with the remaining data arriving in an intermittent yet predictable way. To process this data arrival pattern, we propose *Intermittent Query Processing*(IQP) that does not keep the query active all the time but releases some memory resources when there is no data to process. By exploiting the knowledge of incoming data, IQP selectively keeps a subset of intermediate states when the query is inactive. Therefore, with limited memory consumption IQP can achieve low latency of updating query results for new data.

- **Adaptive Concurrency Control for Main-memory Database** Sep. 2015-Nov. 2017

  We build a main-memory database that supports adaptively mixing multiple forms of concurrency control with minimal overhead. Our system can decompose the workload into partitions and selects a concurrency control protocol for each partition of workload that the protocol is optimized for, and during workload changes adaptively reconfigure the protocols online.

Earlier Projects

- **Structured Data Shuffling for Big Data Analytical Stacks** Nov. 2013-Jan. 2015

  We build a structured data shuffling procedure that can leverage the semantics of SQL queries to apply efficient compression algorithms and discard unnecessary data during data shuffling.

- **A Fast and Space-efficient Join Method for Log Processing in MapReduce** Sep. 2012-Nov. 2013

  We design a join method that achieves high query performance with a small extra storage cost for log processing. It shuffles the log table to avoid huge storage consumption and optimizes the shuffle procedure to achieve high query performance.

System Building Experience

- **Multi-Query Optimization in SparkSQL**

  We extend SparkSQL to support multi-query optimization for all TPC-H queries. Our framework takes Dataframe queries as input and generates a query plan that shares the execution of all submitted queries. We enhance each intermediate tuple during query execution to additionally include the query IDs that the tuple is valid to and modify the SparkSQL operator to leverage this information to support shared query execution.

- **Complex Materialized View Maintenance in SparkSQL**

  We extend SparkSQL to support maintaining materialized views with deletes and updates. Our system supports complex materialized views that are beyond select-project-join-aggregate queries. Specifically, the operators our system supports include select, project, aggregate, sort, materialize, distinct, and all kinds of joins (inner, outer, and anti join) and there is no ordering limit about these operators.

- **Intermediate State Management in PostgreSQL**

  We extend PostgreSQL to support materialized view maintenance with intermediate state management. Our modified PostgreSQL can selectively discard some intermediate states to save memory resources. When a discarded state is required to process new data, we can rebuild this state from saved intermediate states to quickly update the materialized view.

- **Adaptive Concurrency Control in Main-Memory Database**

  We develop a key-value main-memory database which hosts several concurrency control protocols including partition-based concurrency control, two-phase locking, and optimistic concurrency control protocols. Our database uses a machine learning model to adaptively apply a protocol to a partition of the database and reconfigures protocols in response to workload changes.

Honors & Awards

2018	USENIX ATC'18 Student Travel Grant
2016	University Unrestricted (UU) Fellowship - The University of Chicago
2016	CERES 1st year Graduate Research Award - The University of Chicago

Teaching Assistant

Fall 2015	MPCS 51040 - C programming
Spring 2016	MPCS 52040 - Distributed Systems
Winter 2017	CMSC 23500 - Introduction to Database
Winter 2018	CMSC 23500 - Introduction to Database
Winter 2019	CMSC 23500 - Introduction to Database
Winter 2020	CMSC 23500 - Introduction to Database

Referees

Name	Aaron Elmore
Affiliate	University of Chicago
Position	Assistant Professor
Contact	aelmore@cs.uchicago.edu

Name	Michael Franklin
Affiliate	University of Chicago
Position	Liew Family Chairman of Computer Science
Contact	mjfranklin@uchicago.edu

Name	Sanjay Krishnan
Affiliate	University of Chicago
Position	Assistant Professor
Contact	skr@cs.uchicago.edu

Name	Umar Farooq Minhas
Affiliate	Microsoft Research
Position	Principle Researcher
Contact	ufminhas@microsoft.com

Name	Wei Li
Affiliate	Institute of Computing Technology
Position	Associate Professor
Contact	liwei@ict.ac.cn