Symmetric Queries as a Building Block for Efficient Parallel Query Evaluation

Yuqing Melanie Wu
Pomona College Indiana University
Claremont, CA Bloomington, IN

Motivation
- What’s unique in data access patterns in data-intensive applications?
- While such unique data access patterns demands special buffer management in traditional DB engines, how about in parallel environment?

Open Questions
- Is there any summary info or techniques that can help speed up queries that usually demands large number of (self) joins?
- For queries whose complexity is between the groups, to what degree are their evaluation parallelizable?
- If some of these queries are not naturally parallelizable, can we find sub-queries that are?

Broader Questions
- What are the language whose queries are naturally parallelizable?
- How can we identify parallelizable components in a generic query?
- How can we evaluate a generic query efficiently in a parallel environment?

Symmetric Queries
- Why symmetric?
- Symmetric queries we identified:
  - QuineCALC
    \[
    \varphi_1 := \Gamma(x, X) \land x = y \land \forall \phi_2, \forall \psi_1, \exists X \phi_1
    \]
  - SyCALC
    \[
    \varphi_1 := \Gamma(x, X) \land x = y \land \forall \phi_2, \forall \psi_1, \exists X \phi_1
    \]
Properties

• QuineCALC
  – A restricted first-order logic
  – A generalization of symmetric n-ary Boolean function

• SyCALC
  – Extension of QuineCALC, accommodating projection and Cartesian product.

Findings

• All QuineCALC queries are counting only.
• For every SyCALC query q, for every natural number n, there exist a symmetric function that is n-equivalent to q.
• For all natural number m,n and for every symmetric relational function on sequence of n sets that return m-ary relations, there exist an equivalent SyCALC query.

Language Classes

Findings (cont.)

• A SyCALC query is counting-only only if it is equivalent to a quantified Boolean combination of QuineCALC query.
• It is undecidable whether a SyCALC query is counting-only.
• It is decidable whether a SyCALC query is counting-only.

Next Steps

• Applications – What’s new/unique about the queries in
  – Graph DB
  – Complex systems
  – Sensor network
  – Bio-info
  – ……
• Implementing QuineCALC and SyCALC
• Query decomposition

Education

This is not a second thought

• The whole industry expect CS majors to be familiar with the parallel computing environment.
• Another push like OO or more?
IU undergrad curriculum

Core:
- intro to CS
- intro to software systems
- discrete structures for CS
- data structure

Specialization:
- foundation
- intelligent systems
- Programming languages Systems

Elective:
- fundamentals of computing theory
- algorithm design and analysis
- programming language
- compiler
- computer structures
- intro to OS
- intro to AI
- intro to network

DB concepts
robotic
intro to verification
digital design
distributed systems
quantum computing
cryptography

Pomona undergrad curriculum

Intro courses:
- intro to CS
- fundamentals of CS
- discrete math
- data structure
- computability and logic

Core:
- programming language
- algorithms
- systems

Elective:
- software engineering
- user interface design
- database
- robotic
- compiler design
- computer architecture
- natural language processing
- operating systems
- computer vision
- computer graphics
- artificial intelligence
- parallel and real-time computing
- computer network
- parallel and real-time computing

Education

- What has been pushed into undergraduate education?
- Are they in the correct courses?
- What are the core concepts of parallelism?

Thank you

Please contact me at Melanie.Wu@pomona.edu