PQL: A Purely-Declarative Java Extension for Parallel Programming

Peter Pilgerstorfer

PQL

• Declarative query language for parallel programs
  • First order logic
• Java Extension
• Design goals
  • Easy to write parallel code
  • Focus on performance
• Not Turing complete
Overview

• Queries
• Implementation
• Evaluation
Quantifying Queries

• Boolean expression

• Syntax

\[
\langle QUANT \rangle\langle ID \rangle : \langle QUERY \rangle
\]

• Examples

\[
\text{forall int } x : x == x
\]
\[
\text{exists int } x : \text{myArray}[x] == x
\]
Container Queries

- Create a Map, Set or Array
- Syntax
  \[ \text{query}((\text{MATCH})):\langle\text{QUERY}\rangle \]
- Examples
  \[ \text{query}(\text{Set.contains}(x)) : \text{range}(1,10).\text{contains}(x) \]
  \[ \text{query}(\text{Map.get}(x) == y) :
    \text{range}(1,10).\text{contains}(x) \&\& y == 2*x \]
Reduction Queries

• Compute reduction

• Syntax
  \[ \text{reduce}(\langle OP \rangle) \langle ID \rangle [\text{over} \langle IDs \rangle] : \langle QUERY \rangle \]

• Examples
  \[ \text{reduce}(\text{sumInt}) \text{ int } x : \text{mySet}.\text{contains}(x) \]
  \[ \text{reduce}(\text{sumInt}) \text{ int } x \text{ over } y : \text{myMap}.\text{get}(y) == x \]

• (restricted) user defined operations possible
Implementation

• Translate PQL into intermediate language PQIL
  • Only reduce operations and joined expressions
  • Java expressions are seen as constants
• Translate PQIL to nested loops before compilation
Implementation - Access Path Selection

\[ \text{query} (\text{Set.contains}(y)) : \]
\[ \text{func}(x) == y \land \text{arr}[x] == y \]

• Relation size
• Parallelizability
  • In practice outer loop parallel
  • Loop has to be large enough
Optimization

• Eliminate redundant joins
• Merge nested queries
  • Original:
    \[
    \text{query(Map.get(key) == newset):}
    \]
    \[
    \text{  newset == query(Set.contains(value)): array[value] == key;}
    \]
  • Merged:
    \[
    \text{query(Map.get(key) == Set.contains(value)):}
    \]
    \[
    \text{  array[value] == key;}
    \]
Evaluation – Lines of Java code

Manual | Manual-Parallel | Hadoop | SQL | PQL
--- | --- | --- | --- | ---
| bonus | threegrep | webgraph | wordcount |

Legend:
- bonus
- threegrep
- webgraph
- wordcount
Evaluation - Speed
Evaluation - Speed

[Graph showing average execution time vs. number of Java threads for pqm, para-manual, and manual modes. The pqm mode shows a consistently lower execution time across all thread counts.]
Discussion

• Declarative language
• Works with Java data structures
• Little computational overhead compared to manual parallelization