Automatic Parallelization of Graph Queries with MapReduce

Tao Zan¹  Yu Liu¹  Zhenjiang Hu²
¹The Graduate University for Advanced Studies  ²National Institute of Informatics

Problem

♦ Graphs in real world can be quite huge with millions of nodes and edges.

♦ How to parallelize graph queries efficiently is an important issue.

Goal

We aim at a new way of processing graph queries with MapReduce by translating UnQL query to structural recursion, then parallelizing structural recursion. When achieving this goal, thus can we use UnQL for large scale data processing by MapReduce.

Background

♦ UnQL is a language that was designed for querying unstructured data.

An example:

Q1 = select t
where *.a.t in $db

Methods, Results and Future Work

♦ Methods

● Properties for Parallelization

All join-free queries can be highly parallelized in MapReduce based on three decomposable properties for structural recursion.

A graph can be expressed as:

\[ g = \&x \circ \text{cycle}(g_1 \circ g_2 \circ ... \circ g_n) \]

By induction:

\[ \text{rec}(e)(g) = \text{rec}(e)(\&x) \circ \text{cycle}(g_1 \circ g_2 \circ ... \circ g_n) \]

\[ = \text{rec}(e)(\&x) \circ \text{rec}(e)(g_1) \circ \text{rec}(e)(g_2) \circ ... \circ \text{rec}(e)(g_n) \]

Graph Data

Graph Queries on Hadoop

Preprocessing: Bulk computing
Local \(\varepsilon\)-edge elimination

Iterative-Map:
Edge reversing

Iterative-Reduce:
1. Eliminate \(\varepsilon\)-edges
2. Remove unreachable part

Iterate over the results

Result Graph

Future work

♦ Parallelization of graph query on cyclic graphs.

A combinational approach for parallelizing graph query UnQL.