CCIT Report #589 May 2006

Efficient Dynamic Aggregation

Yitzhak Birk, Idit Keidar, Liran Liss, and Assaf Schuster
Technion – Israel Institute of Technology
{birk@ee,idish@ee,liranl@tx,assaf@cs}.technion.ac.il

June 24, 2006

Abstract

We consider the problem of *dynamic aggregation* of inputs over a large graph. A dynamic aggregation algorithm must continuously compute the result of a given aggregation function over a dynamically changing set of inputs. To be efficient, such an algorithm should refrain from sending messages when the inputs do not change, and should perform *local* communication whenever possible.

We show an instance-based lower bound on the efficiency of such algorithms, and provide two algorithms matching this bound. The first, MultI-LEAG, re-samples the inputs at intervals that are proportional to the graph size, and is extremely message efficient. The second, DynI-LEAG, more closely monitors the aggregate value by sampling it more frequently, at the cost of slightly higher message complexity.

1 Introduction

We consider the problem of continuous monitoring of an aggregation function over a set of dynamically changing inputs on a large graph. We term this problem *dynamic aggregation*. For example, the inputs may reflect sensor readings of temperature or seismic activity, or load reported by computers in a computational grid. The aggregation function may compute the average temperature, or whether the percentage of sensors that detect an earthquake exceeds a certain threshold, or the maximum computer load. It is desirable to seek *local* solutions to this problem, where input values and changes thereof do not need to be communicated over the entire graph.

Since virtually every interesting aggregation function has some input instances on which it cannot be computed without global communication, a priori, it is not clear whether it is possible to do better. Nevertheless, in a recent work [1], we have shown that when computing an aggregation function on a large graph for fixed (in time) inputs, it is in many cases possible to reach the correct result without global communication. Specifically, while *some* problem instances trivially require global communication, many instances can be computed locally, i.e., in a number of steps that is independent of the graph size. We have introduced