Differential dataflow
by F. McSherry, D. G. Murray, R. Isaacs, and M. Isard

1 Summary

In this paper authors introduced differential computation, which is a generalization of current techniques of incremental computations. They also introduced a definition of differential data, which shows practical application of differential computation in parallel settings. The motivation presented by authors emphasize performance of the new approach, which is very high especially for dynamically changing data.

Authors introduced and implemented in LINQ\(^1\) new query operators, which find a fixed point of a process with and without prioritization, e.g., finding connected components of a graph. Such operators are then used to modify basic database operations (e.g. join, group-by), in order to allow them working under differential and incremental computations.

The paper describes also a sketch of the implementation of the prototype Naiad system, which was used for experiments. In the appendix authors described a demo of the Naiad, and key pieces of code implementing an algorithm of finding connected components in incremental computation settings.

2 Strong Points

- The paper presents a nice idea to store differences \((A_t - \sum_{i \leq t} \delta A_i)\) not the reported values.
- Authors presents also current solutions, including efficient ones: Stateless, Incremental, and Prioritized.
- In the paper, authors introduce two new operations that extends LINQ as well as different implementations for a few settings, e.g., data-parallel operation

3 Weak Points

- The paper is well written, but sometime authors uses new notions before defining them, e.g., version.
- In the Section 3 authors talk about operators. Their arguments bases on a silent assumption that the operator is linear, which not always have to be truth. In there settings it works well, because count, which is used in the connected component problem, is linear, but what if the function is e.g. quadratic?
- Authors often refer to the running example, and present other example implementations of their general approach. However, they do not deeply analyze general properties of differential data.
- The paper presents very few experiments.

4 Questions and Discussion Points

It would be very interesting to deeply analyze differential data and differential computation in general, and not only using defined examples. Such analyze would help to apply presented approach in new domains and different types of problems. In addition, such analysis would define domains and problems, which benefit from differential computation. It would also help to define features of problems and algorithms that will not perform any better when implemented in differential computation style.

---

\(^1\)An extension of .NET, which allows to define declarative queries, e.g., for database, as programming language instructions.