Abstract

This thesis presents research which spans three conference papers and one manuscript which has not yet been submitted for peer review.

The topic of 1 is the inherent complexity of maintaining perfect height in B-trees. We consider the setting in which a B-tree of optimal height contains $n = (1 - \epsilon)N$ elements where N is the number of elements in full B-tree of the same height (the capacity of the tree). We show that the rebalancing cost when updating the tree—while maintaining optimal height—depends on ϵ. Specifically, our analysis gives a lower bound for the rebalancing cost of $\Omega(1/(\epsilon B))$. We then describe a rebalancing algorithm which has an amortized rebalancing cost with an almost matching upper bound of $\mathcal{O}(1/(\epsilon B) \cdot \log^2(\min\{1/\epsilon, B\}))$. We additionally describe a scheme utilizing this algorithm which, given a rebalancing budget $f(n)$, maintains optimal height for decreasing ϵ until the cost exceeds the budget at which time it maintains optimal height plus one. Given a rebalancing budget of $\Theta(\log n)$, this scheme maintains optimal height for all but a vanishing fraction of sizes in the intervals between tree capacities.

Manuscript 2 presents empirical analysis of practical randomized external-memory algorithms for computing the connected components of graphs. The best known theoretical results for this problem are essentially all derived from results for minimum spanning tree algorithms. In the realm of randomized external-memory MST algorithms, the best asymptotic result has I/O-complexity $\mathcal{O}(\text{sort}(|E|))$ in expectation while an empirically studied practical algorithm has a bound of $\mathcal{O}(\text{sort}(|E|) \cdot \log(|V|/M))$. We implement and evaluate an algorithm for connected components with expected I/O-complexity $\mathcal{O}(\text{sort}(|E|))$—a simplification of the MST algorithm with this asymptotic cost, we show that this approach may also yield good results in practice.

In paper 3, we present a novel approach to simulating large-scale population protocol models. Naive simulation of N interactions of a population protocol with n agents and m states requires $\Theta(n \log m)$ bits of memory and $\Theta(N)$ time. For very large n, this is prohibitive both in memory consumption and time, as interesting protocols will typically require $N > n$ interactions for convergence. We describe a histogram-based simulation framework which requires $\Theta(m \log n)$ bits of memory instead—an improvement as it is typically the case that $n \gg m$. We analyze, implement, and compare a number of different data structures to perform correct agent sampling in this regime. For this purpose, we develop *dynamic alias tables* which allow sampling an interaction in expected amortized constant time. We then show how to use sampling techniques to process agent interactions in batches, giving a simulation approach which uses subconstant time per interaction under reasonable assumptions.

With paper 4, we introduce the new model of *fragile complexity* for comparison-based algorithms. Within this model, we analyze classical comparison-based problems such as finding the minimum value of a set, selection (or finding the median), and sorting. We prove a number of lower and upper bounds and in particular, we give a number of randomized results which describe trade-offs not achievable by deterministic algorithms.