Scalable Splitting of Massive Data Streams

Erik Zeitler, Tore Risch
Department of Information Technology
Uppsala University
Sweden

Abstract. Scalable execution of continuous queries over massive data streams often requires splitting input streams into parallel sub-streams over which query operators are executed in parallel. Automatic stream splitting is in general very difficult, as the optimal parallelization may depend on application semantics. To enable application specific stream splitting, we introduce splitstream functions where the user specifies non-procedural stream partitioning and replication. For high-volume streams, the stream splitting itself becomes a performance bottleneck. A cost model is introduced that estimates the performance of splitstream functions with respect to throughput and CPU usage. We implement parallel splitstream functions, and relate experimental results to cost model estimates. Based on the results, a splitstream function called autosplit is proposed, which scales well for high degrees of parallelism, and is robust for varying proportions of stream partitioning and replication.

We show how user defined parallelization using autosplit provides substantially improved scalability (L = 64) over previously published results for the Linear Road Benchmark. The benchmark was run under Linux on a cluster of compute nodes featuring two quad-core Intel® Xeon® E5430 CPUs @ 2.66GHz and 6144 KB L2 cache. Six such compute nodes (48 cores in total) were available for the experiments. For inter-node communication, TCP/IP was used over gigabit Ethernet. Intra-node communication used TCP/IP over the loopback interface.

Keywords: Distributed stream systems, parallelization, query optimization.

Accepted for publication in *Proc. 15th Conf. on Database Systems for Advanced Application, DASFAA 2010,* http://dasfaa2010.cs.tsukuba.ac.jp/index.html, Tokyo, Japan, 1-4 April, 2010.