発表概要

トポロジを考慮しソース選択を行うデータ転送スケジューラ

高橋 慧[†1] 田浦 健次朗[†1] 近山 隆[†1]

本発表では，トポロジ情報を用いて転送時のリンクの共有を回避することで，各ノードに到着するデータのスループットを最大化するデータ転送スケジューラを提案する．自然言語処理等のデータ・インテンシブなアプリケーションでは，全体の実行時間に占めるデータ転送時間が大きいので，効率的なデータ転送が重要になる．たとえば N 本のデータ転送が 1 本のリンクを共有すると，それぞれの転送の速度は最悪で 1/N と大幅に低下してしまう．もしデータのソースが複数あり，トポロジ情報を用いてこれらを適切に選択すれば，このリンクの共有による転送速度の低下を回避できる．しかし既存のデータ転送スケジューラは，トポロジを考慮せずノード間で一定なバンド幅を仮定しており，このリンクの共有を検知することができない．そこで我々はトポロジ情報を用い，あるノードが必要とするデータが複数のソースにある場合は，最も大きなバンド幅で転送できるソースを選択する．複数のノードが同一のソースを用いる場合には，それらのノードを 1 列に並べてパイプライン転送を行う．また，あるリンクを複数の転送が共有しているとき，多くのノードが必要としているデータの転送は重要であると考えられるので，より多くのバンド幅を与えるような最適化を線形計画法を用いて行う．これらの工夫によって，全体のファイル転送のスループットが向上する．この転送計画アルゴリズムを実装し，実験を行った．

A Data Transfer Scheduler Selecting Sources by Using Topology Information

Kei Takahashi,[†1] Kenjiro Taura[†1] and Takashi Chikayama[†1]

We propose a data transfer scheduler avoiding link sharing among transfers and trying to maximize throughput arriving at each node. Data transfer scheduling is especially important in the execution of data intensive applications, such as natural language processing, since data transfer takes much of their execution time. For instance, when N data transfers share one link, each transfer speed decays down to 1/N. If data have multiple replicas and if each node selects data source from them by using network topology information, this decay may be avoided. However, existing data transfer schedulers do not consider network topology, and only assume static bandwidth among nodes. Thus, the decay of transfer speed cannot be detected. We plan efficient data transfer scheduling by using network topology and bandwidth information. When data have multiple replicas, a replica reached with the maximum bandwidth is selected. When multiple nodes need data located at one source, the data are multicasted in a pipeline fashion. In addition, we actively control each transfer bandwidth: when multiple multicast transfers share a link, more bandwidth is given to a multicast transfer with more nodes. Consequently, the total throughput is improved. We implemented and evaluated this transfer algorithm.

（平成 19 年 8 月 1 日発表）

†1 東京大学
The University of Tokyo