An Experimental Study of k-Splittable Scheduling for DNS-Based Traffic Allocation

Abstract:
The Internet domain name service (DNS) uses rotation of address lists to perform load distribution among replicated servers. We model this kind of load balancing mechanism in form of a set of request streams with different rates that should be mapped to a set of servers. Rotating a list of length $k$ corresponds to breaking streams into $k$ equally sized pieces. We compare this and other strategies of how to break the streams into a bounded number of pieces and how to map these pieces to the servers. One of the strategies that we study computes an \textit{optimal k-splittable allocation} using a scheduling algorithm that breaks streams into at most $k \ge 2$ pieces of possibly different size and maps these pieces to the servers in such a way that the maximum load over all servers is minimized. For this purpose we use a recently introduced algorithm for the k-splittable machine scheduling problem. The running time of this algorithm is only linear in the number of streams but exponential in the number of machines. We suggest an improvement to this algorithm for the case of identical machines (servers), and prove that our modification reduces the running time significantly. This enables us to compute optimal allocations for several hundreds of servers, although this problem is known to be $\text{np-complete}$-hard. Our experimental study is done using the network simulator SSFNet. We study the average and maximum delay experienced by HTTP requests for various traffic allocation policies and
traffic patterns. Our simulations show that splitting data streams can reduce the maximum as well as the average latency of HTTP requests significantly. This improvement can be observed even if streams are simply broken into $k$ equally sized pieces that are mapped randomly to the servers. Using allocations computed by machine scheduling algorithms, we observe further significant improvements.

**Stichworte:**
scheduling; splittability; traffic allocation problem; server farm; load balancing; internet; web server; optimal solution

**Jahr:**
2003

**Jahr / Monat:**
2003-04-01 00:00:00

**Seiten:**
19

**Occurences:**
- Einrichtungen > Fakultäten > Fakultät für Informatik > Technische Berichte > 2003

**Entries:**