J. Parallel Distrib. Comput. 74 (2014) 2423–2432

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Static load-balanced routing for slimmed fat-trees

Xin Yuan [a,*], Santosh Mahapatra [a], Michael Lang [b], Scott Pakin [b]

[a] CSD, Florida State University, Tallahassee, FL 32306, United States
[b] Los Alamos National Laboratory, Los Alamos, NM, United States

HIGHLIGHTS

- A new load balanced routing scheme, *RRR*, for slimmed fat-trees.
- A combined routing scheme that integrates *RRR*, D-mod-k, and S-mod-k.
- Performance study.

ARTICLE INFO

Article history:
Received 29 August 2013
Received in revised form
27 January 2014
Accepted 2 February 2014
Available online 15 February 2014

Keywords:
Fat-tree
Static routing
Interconnect
Single-path routing

ABSTRACT

Slimmed fat-trees have recently been proposed and deployed to reduce costs in High Performance Computing (HPC) clusters. While existing static routing schemes such as destination-mod-k (D-mod-k) routing are load-balanced and effective for full bisection bandwidth fat-trees, they incur significant load imbalance in many slimmed fat-trees. In this work, we propose a static load balanced routing scheme, called Round-Robin Routing (*RRR*), for 2- and 3-level extended generalized fat-trees (XGFTs), which represent many fat-tree variations including slimmed fat-trees. *RRR* achieves near perfect load-balancing for any such XGFT in that links at the same level of a tree carry traffic from almost the same number of source–destination pairs. Our evaluation results indicate that on many slimmed fat-trees, *RRR* is significantly better than D-mod-k for dense traffic patterns due to its better load-balancing property, but performs worse for sparse patterns. We develop a combined routing scheme that enjoys the strengths of both *RRR* and D-mod-k by using *RRR* in conjunction with D-mod-k. The combined routing is a robust load-balanced routing scheme for slimmed fat-trees: it performs similar to D-mod-k for sparse traffic patterns and to *RRR* for dense patterns.

1. Introduction

Recent studies [1,6] have suggested that full bisection-bandwidth fat-trees are under-utilized for HPC applications and that slimmed fat-trees where the upper level links are over-subscribed can be used to reduce the cost without significantly sacrificing the communication performance. Slimmed fat-trees have been deployed in many recent large scale HPC systems including some of the current largest supercomputers in the world such as the Stampede supercomputer with a 5:4 over-subscribed slimmed fat-tree [16], and Tianhe-1A with a 2:1 over-subscribed slimmed fat-tree [17].

Full bisection-bandwidth fat-trees and slimmed fat-trees can both be described by the extended generalized fat-trees (XGFTs) [13]. This paper considers deterministic traffic-oblivious load balanced single-path routing for XGFTs that assumes no knowledge of traffic patterns. Without knowing the traffic pattern, the main objective of such a scheme is to route the traffic such that each link is used by a similar number of source–destination (SD) pairs.

Existing deterministic single-path routing schemes for XGFTs mainly include destination-mod-k (D-mod-k) routing [3,4,9,15,20] and source-mod-k (S-mod-k) routing [7,13,15]. Analyses performed on D-mod-k have concluded that it is better than random routing [5] and some adaptive routing [3,15]. Since D-mod-k and S-mod-k have similar properties and performance [15], we will focus on D-mod-k. D-mod-k is effective for a wide collection of communication patterns on full bisection-bandwidth fat-trees [9,20]. From the load balancing perspective, D-mod-k achieves perfect load balance for such fat-trees: links at the same level carry traffic from exactly the same number of SD pairs. For most slimmed fat-trees, however, D-mod-k is not load balanced. Fig. 1 in Section 3

* Corresponding author.
E-mail addresses: xyuan@cs.fsu.edu (X. Yuan), mahapatr@cs.fsu.edu (S. Mahapatra), mlang@lanl.gov (M. Lang), pakin@lanl.gov (S. Pakin).

http://dx.doi.org/10.1016/j.jpdc.2014.02.001
0743-7315/

shows an example where some link carries traffic from twice as many SD pairs as other links using D-mod-k. Such load imbalance degrades the effectiveness of D-mod-k on slimmed fat-trees.

We propose a load balanced routing scheme, Round-Robin Routing (*RRR*), as an alternate to D-mod-k for 2- and 3-level XGFTs. *RRR* achieves near perfect load balancing for any such XGFT: links at the same level carrying traffic from a similar number of SD pairs. Thus, *RRR* has a better load-balancing property than D-mod-k on slimmed fat-trees where D-mod-k incurs load imbalance. Note that while the techniques used in these routing schemes can potentially be extended to develop load balanced routing schemes for larger trees such as 4-level trees, directly applying the algorithms for 2- and 3-level trees does not yield a load balanced routing for 4-level trees.

D-mod-k has a nice property that we call *traffic concentration property*: all traffic that must go through top level switches to a destination uses the same top level switch. Due to the traffic concentration property, network contention for uplinks and downlinks is correlated with D-mod-k: the routes for two different SD pairs do not have contention in downlinks if the routes do not have contention in uplinks. This property results in improved performance as shown in [15]. While *RRR* achieves near perfect load-balancing, it does not have the traffic concentration property. Our evaluation indicates that *RRR* achieves significant better performance than D-mod-k for dense traffic patterns where load balancing is the dominating factor. However, for sparse permutation-like traffic patterns, *RRR* performs worse than D-mod-k, which reflects the interplay of the load balancing and traffic concentration properties.

To overcome the limitations of both *RRR* and D-mod-k, we develop a combined routing scheme that uses *RRR* in conjunction with D-mod-k and S-mod-k. The combined routing scheme schedules most of SD pairs using D-mod-k, some SD pairs using S-mod-k, and the remaining SD pairs using *RRR*. It has the advantages of both D-mod-k and *RRR*: it still enjoys near perfect load balancing while concentrating most of the traffic since most of SD pairs are scheduled using D-mod-k. Our evaluation demonstrates that the combined scheme is a robust static load-balanced routing scheme for slimmed fat-trees. On various slimmed fat-trees, it performs like D-mod-k for sparse patterns and *RRR* for dense patterns.

It must be noted that our proposed schemes use both source and destination identifiers to compute the route for each SD pair. The schemes are not destination based routing schemes. Hence, although the route calculation complexity in our scheme is similar to that in the D-mod-k routing, for the current Infiniband networks that use destination based routing, our scheme can only be realized with the destination renaming technique [10,12] that introduces significantly higher resource requirement that the D-mod-k routing.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 illustrates XGFTs and the load imbalance of D-mod-k on slimmed fat-trees. Section 4 details the proposed *RRR*. Section 5 describes the combined routing scheme. Section 6 reports the results of our performance study. Finally, Section 7 concludes the paper.

2. Related work

Since the inception of fat-trees as general purpose interconnection networks [8], many variants of fat-trees have been proposed and studied including the *m*-port-*n*-trees [9], *k*-ary-*n*-trees [14], generalized fat-trees (GFTs) and extended generalized fat-trees (XGFTs) [13]. Existing single-path routing schemes for XGFTs mainly include random routing where a random path is selected for each SD pair [2,4], the S-mod-k routing [7,13,15], and the D-mod-k routing [3,4,9,15,20]. The S-mod-k and the D-mod-k routing have been shown to have negligible difference in performance [15].

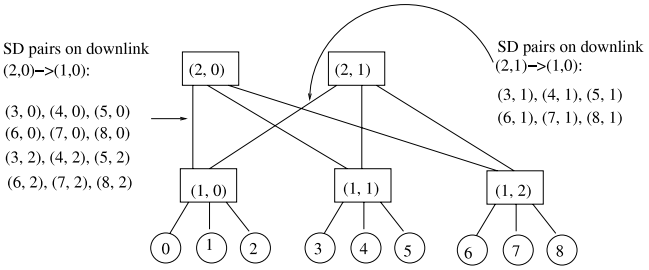

Fig. 1. Load imbalanced on XGFT(2; 3, 3; 1, 2) with D-mod-k.

Analyses also conclude that D-mod-k performs better than random routing [5] and some adaptive routing [3,15]. In addition, it can also support all global shift communication patterns without contention [19]. Other single-path routing schemes [11,18] are only effective for fat-trees of certain shapes. None of the existing deterministic routing schemes achieves load balancing on all slimmed fat-trees. In this work, we develop a deterministic routing scheme that achieves near optimal load balancing for any 2- and 3-level XGFT.

3. Background

We will describe XGFT [13] and introduce the terminologies used later in the paper. An $XGFT(h; m_0, m_1, \ldots, m_{h-1}; w_0, w_1, \ldots, w_{h-1})$ has $h+1$ levels of nodes. Each level i node, $0 \le i \le h-1$, has w_i parents; and each level i node, $1 \le i \le h$, has m_{i-1} children. For each level i, $1 \le i \le h-1$, the ratio of the number of links going down and the number of links going up, that is, $m_{i-1} : w_i$ is called the *over-subscription ratio*. For full bisection bandwidth fat-trees, the ratio is 1 at all levels. For slimmed fat-trees, the ratio is larger than 1 at some levels. Level 0 nodes are *processing nodes* while nodes in other levels are *switches*. In general, $XGFT(h; m_0, m_1, \ldots, m_{h-1}; w_0, w_1, \ldots, w_{h-1})$ has $(\prod_{i=k+1}^{h} m_{i-1}) \times (\prod_{i=1}^{k} w_{i-1})$ switches at level k, $1 \le k \le h$, and $(\prod_{i=1}^{h} m_{i-1})$ processing nodes at level $k = 0$. Note that our XGFT notation is slightly different from that in [13]: our index starts from 0 while their index starts from 1.

The nodes in an XGFT can be numbered by the tuple $(l, a_h, a_{h-1}, \ldots, a_1)$ where l is the level; and for all i, $l+1 \le i \le h$, $0 \le a_i \le m_{i-1}$; and for all i, $1 \le i \le l$, $0 \le a_i \le w_{i-1}$. A node $(l, a_h, a_{h-1}, \ldots, a_{l+1}, a_l, a_{l-1}, \ldots, a_1)$ connects to nodes at level $l+1$ $(l+1, a_h, a_{h-1}, \ldots, X, a_l, a_{l-1}, \ldots, a_1)$ **using port** X, $0 \le X < w_l$, and nodes at level $l-1$ $(l-1, a_h, a_{h-1}, \ldots, a_{l+1}, Y, a_{l-1}, \ldots, a_1)$, $0 \le Y < m_{l-1}$. We will use slight variations of this numbering scheme in this paper.

In an XGFT, the path from source s to destination d always consists of an upward path to one of the nearest common ancestors (NCAs) of s and d, and a unique downward path from the NCA to d. Since the downward path is unique, routing for XGFTs selects the upward path. Before reaching the NCA, at each level, D-mod-k chooses a parent node connected to the port identified by $\lfloor \frac{d}{\prod_{i=0}^{k-1} w_i} \rfloor \bmod w_k$.

D-mod-k is able to achieve load balancing when for every level i, $1 \le i \le h-1$, m_{i-1} is a multiple of w_i. This includes full bisection bandwidth fat-trees and some special slimmed fat-trees. For other types of trees, however, D-mod-k is not load balanced. Fig. 1 shows such a case on $XGFT(2; 3, 3; 1, 2)$. With D-mod-k, switch $(2, 0)$ carries traffic to destinations 0, 2, 4, 6, 8 while switch $(2, 1)$ carries traffic to destinations 1, 3, 5, 7. As can be seen from the figure, the downlinks are not balanced: link $(2, 0) \rightarrow (1, 0)$ carries traffic from 12 SD pairs while link $(2, 1) \rightarrow (1, 0)$ carries traffic from only 6 SD pairs. The proposed *RRR* overcomes this problem for any 2- or 3-level XGFT.