TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute for Computer Engineering, Computer Architecture

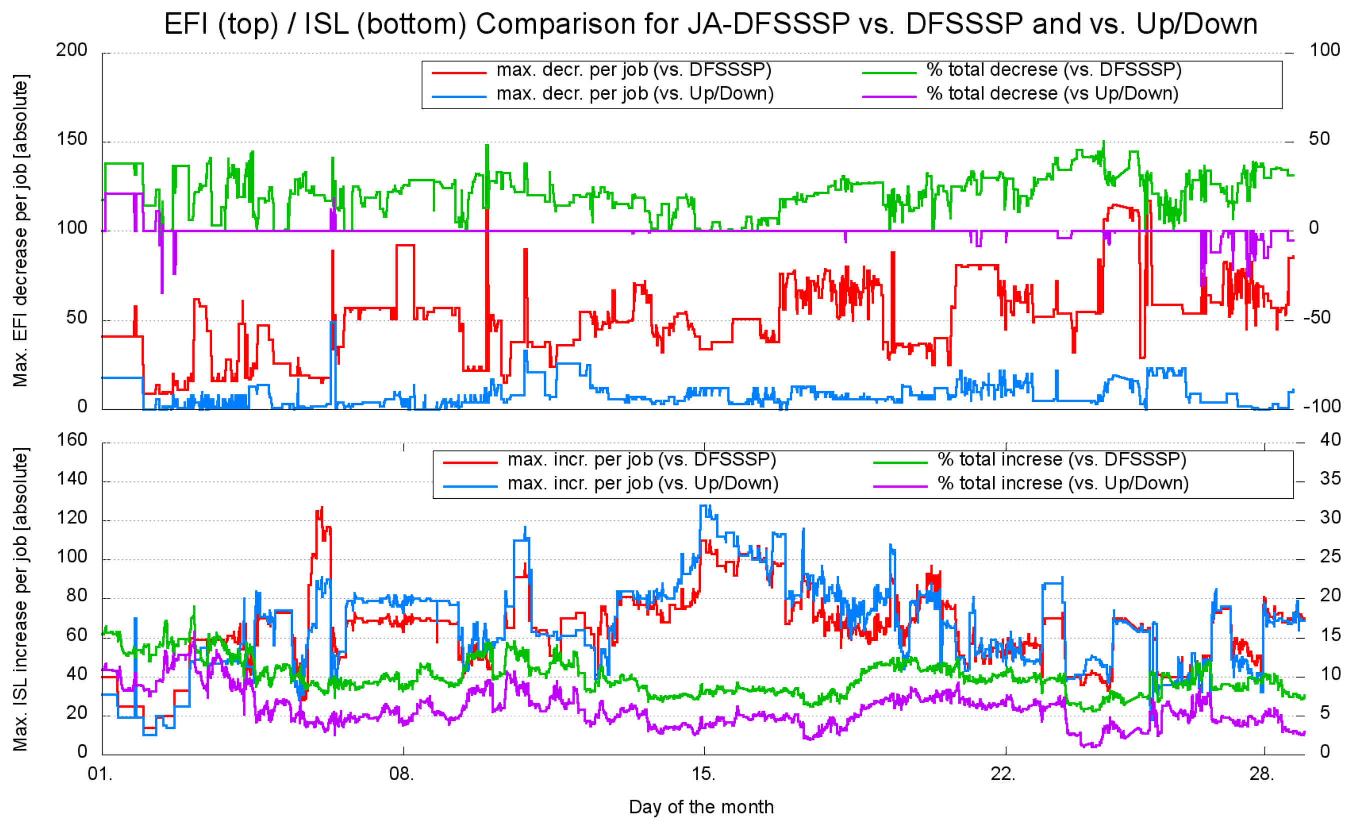
Increasing Fabric Utilization with Job-Aware Routing

Improvement of Network Metrics for a Multi-Job HPC Environment

Our job-aware DFSSSP (JA-DFSSSP) routing is designed to increase the IB fabric utilization of an HPC-system, which is used by many users for varying workloads. The figure to the right compares our JA-DFSSSP to the following two IB routing algorithms:

a) Up*/Down* routing, a common choice in production HPC-systems (incl. Taurus)
b) the "default" DFSSSP routing, as implemented in OFED's OpenSM v3.3.18

For the analysis we used the March 2015's workload (job size/locality) of the 509-node HPC-system Taurus, located at TU Dresden. We compare edge forwarding index (EFI), and



the number of inter-switch links (ISL) used on a "per job" basis and for the sum of jobs running concurrently. Analyzing the effects of different routings on realistic HPC workloads reveals:

a) JA-DFSSSP vs. UpDn: maximum EFI↘ by a value of **8** and ISL↗ by **66** on avg. per job
b) JA-DFSSSP vs. DFSSSP: max. EFI↘ by a value of **50** and ISL↗ by **64** on avg. per job
c) avg. boost of ISL by JA-DFSSSP is **10%** **(6%)** considering all concurrently running jobs compared to DFSSSP (and Up/Down)
d) max. ISL↗ / EFI↘ was **19%/51%** **(16%/21%)**

➔ This indicates a noticeable communication performance improvement thru JA-DFSSSP.

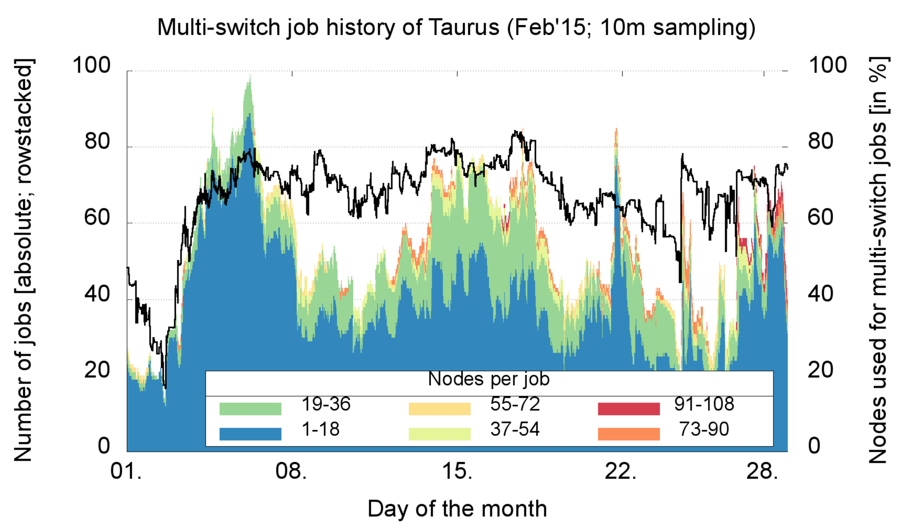
Introduction

The InfiniBand fabric of an HPC-system is a shared resource for simultaneously running applications. The communication performance of these scientific codes depends heavily on the used routing algorithm. Many of the oblivious routing algorithms in the InfiniBand subnet manager [1] optimize for global path balancing which is suboptimal in a multi-job environment. An alternative approach is to attribute as much physical network components (links, switches) as possible to each multi-node application. This can be accomplished by our job-aware DFSSSP [2] routing and Slurm extension [3].

Multi-User/Multi-Job HPC System

Large HPC-systems at national labs and universities are usually used by many users running a diverse set of serial and parallel applications (differ in node count, runtime, and communication pattern, etc). Our intended optimizations are only relevant to multi-node jobs using more than one IB switch in the fabric. We analyze the job history of the Taurus HPC-system (494 compute + 15 I/O and admin nodes in three "full" fat-tree islands; using 52 36-port FDR IB switches), see statistics of March 2015 on the right. Two facts are surprising:

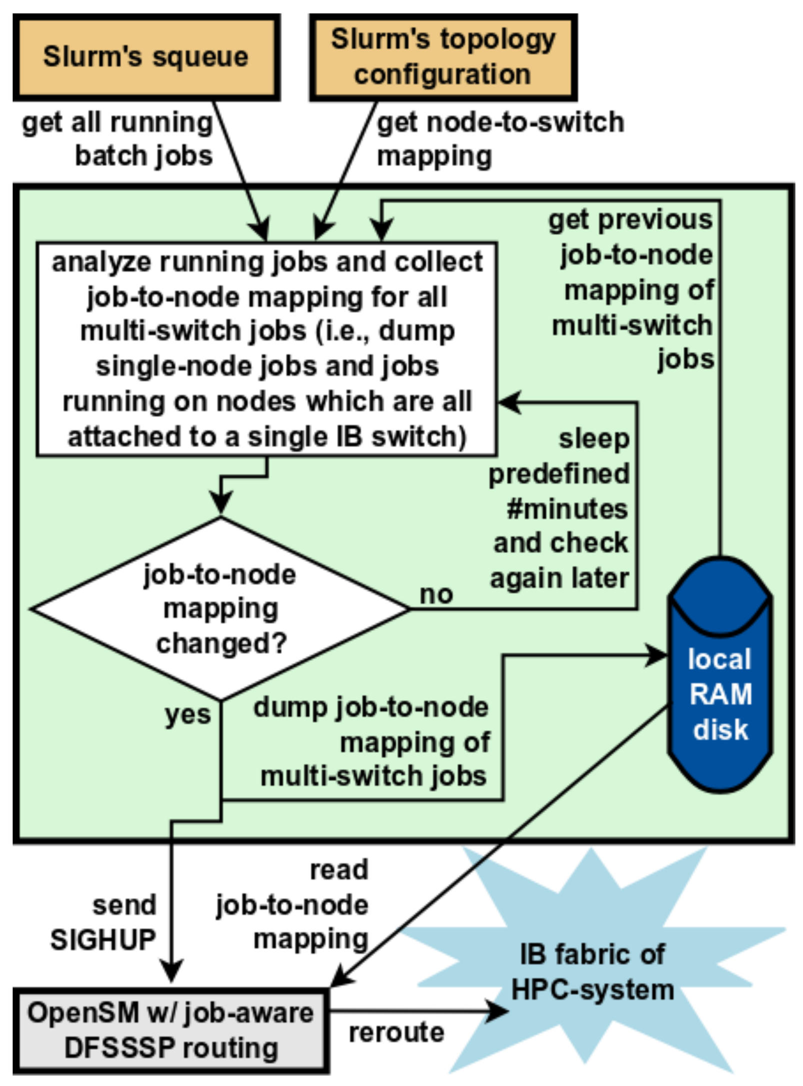
a) only **66.8%** (avg.) of nodes are used by multi-switch jobs
b) a large number of small jobs (size range: 2-18 nodes) is unnecessarily distributed over multiple switches.



Interface between Slurm and OpenSM

We designed a program which analyzes the size and location of running jobs and triggers a rerouting of the fabric (via OpenSM) to optimize the path balancing. This program will identify all running multi-node jobs which have their compute nodes connected to more than one switch. The detailed operational steps are:

```
while (TRUE):
    query squeue for all running jobs;
    get topology (node->switch map);
    filter multi-node, multi-switch jobs;
    if (job-to-node mapping changed):
        write new job-to-node to disk;
        send SIGHUP to OpenSM;
    sleep defined #minutes;
```



The verification if the job-to-node mapping has changed compared to previous iterations ignores the actual job IDs, since a new job (same size/location) does not require recomputation of the forwarding tables.

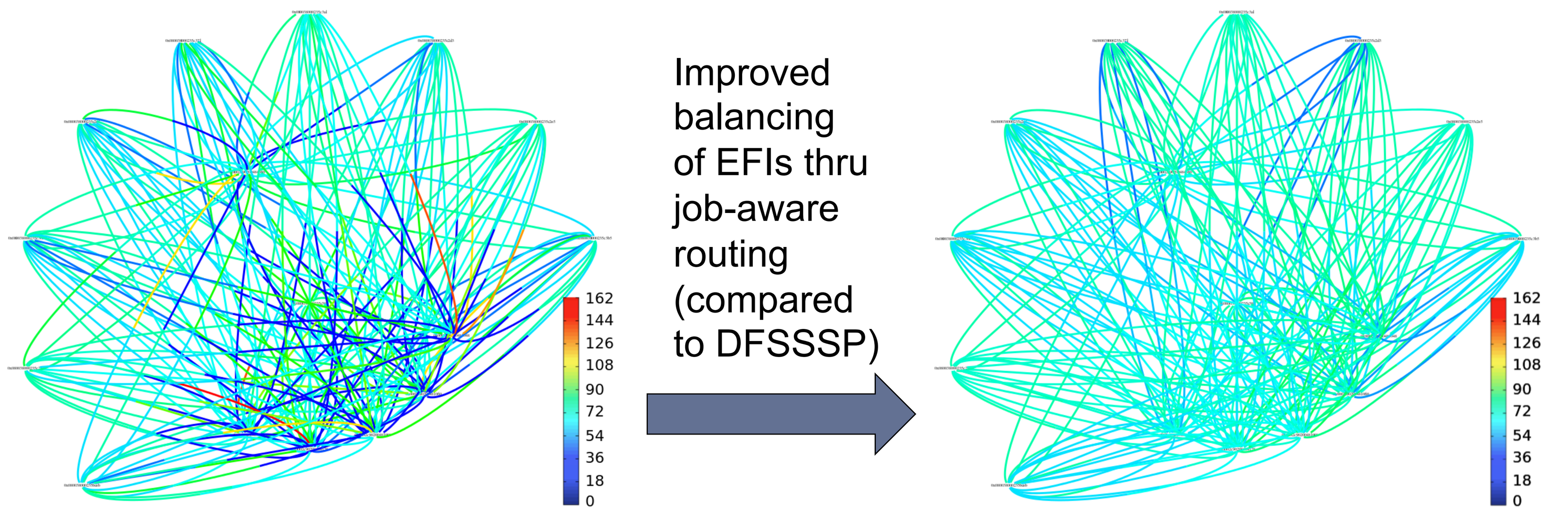
Job-Aware DFSSSP Routing

The deadlock-free single-source shortest-path routing, called DFSSSP [2], is able to optimize the source/destination path-balancing globally. As a result:

☺ The potential throughput of a single application running on the whole system is increased (especially for irregular fabric topologies);
☹ Our analysis showed, that DFSSSP (and other routings implemented in OpenSM) fails to optimize for a multi-job environment, i.e., small/medium sized jobs performance suffers from high edge forwarding indices (EFI) and only use a subset of available links/switches in the fabric.

IB's static and oblivious routing approach can yield in suboptimal performance due to a mismatch between communication pattern and routing, see [4]. We extended the DFSSSP routing to make it job-aware which increase the number of links used and decreases the EFI on a per-job basis. Therefore, we modified OpenSM/DFSSSP to perform additional tasks in each re-routing step:

a) read job-to-node mapping and process the LIDs in the order: LIDs of job 1, LIDs of job 2, … (descending order of job size) and then "unused" LIDs
b) perform a selective edge weight update, i.e., update only if the u➔v path will potentially be used by a multi-node job currently running on the fabric.



The figure above, showing a heat map of EFI for inter-switch links, depicts the improvement when one island of Taurus is used by 3 synthetic, parallel jobs of equivalent size (max. EFI decreased by **59%**; #{used links} increased by **7%**).

Conclusion

Our job-aware DFSSSP routing combined with the batch job analysis/filtering program is capable of optimizing the routing in an IB fabric, resulting in:

a) lower EFI on a "per job" basis (avg. max. ↘ of **51%** compared to DFSSSP)
b) increased fabric utilization, i.e. more switches/links potentially being used.

Acknowledgement

We would like to thank the colleagues from Mellanox and SchedMD, and our colleagues of the ZIH, TU Dresden, for the stimulating discussions about routing optimizations and their help analyzing the job history of the 509-node Taurus HPC-system, located at TU Dresden.

References

[1] Mellanox OFED for Linux User Manual, Rev 2.0-3.0.0 ed., Mellanox Technologies, Oct. 2013.
[2] J. Domke, T. Hoefler, and W. E. Nagel, "Deadlock-Free Oblivious Routing for Arbitrary Topologies", in Proceedings of the 25th IEEE IPDPS, May 2011, pp. 613–624.
[3] M. Jette, A. Yoo, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management", in Lecture Notes in Computer Science: Proceedings of JSSPP, 2002, pp. 44–60.
[4] T. Hoefler, T. Schneider, and A. Lumsdaine, "Multistage Switches are not Crossbars: Effects of Static Routing in High-Performance Networks," in Proceedings of the IEEE Cluster 2008, Oct. 2008.

Jens Domke (jens.domke@tu-dresden.de)
Tel. +49 351 - 463 – 38783
Faculty of Computer Science (APB 1046)
Nöthnitzer Straße 46, 01187 Dresden, Germany

ZIH Center for Information Services & High Performance Computing

DRESDEN concept Exzellenz aus Wissenschaft und Kultur