Fail-in-Place Network Design
Interaction between Topology, Routing Algorithm and Failures

TOKYO TECH Pursuing Excellence

Introduction

The growing system size of high performance computers results in a steady decrease of the mean time between failures. Exchanging network components often requires whole system downtime which increases the cost of failures. In this work, we study a fail-in-place strategy where broken network elements remain untouched. We show, that a fail-in-place strategy is feasible for todays networks and the degradation is manageable, and provide guidelines for the design.

Contributions

- We show that fail-in-place network design can be accomplished with an appropriate combination of topology and routing algorithm.
- We conducted detailed case studies with TSUBAME2.5 and Deimos, and showed that the currently used routing method on TSUBAME2.5 can be improved, increasing the throughput by up to 2.1x (and 3.1x for Deimos) for the fault-free network while increasing their fail-in-place characteristics.

TABLE I. COMPARISON OF NETWORK-RELATED HARDWARE AND SOFTWARE FAILURES, MTBF/MTTR, AND ANNUAL FAILURE RATES

Fault Type	Deimos*	LANL Cluster 2	TSUBAME2.5
	Percentages of network-related failures		
Software	13%	8%	1%
Hardware	87%	46%	99%
Unspecified		46%	
	Percentages for hardware only		
NIC/HCA	59%	78%	1%
Link	27%	7%	93%
Switch	14%	15%	6%
	Mean time between failure / mean time to repair		
NIC/HCA	X† / 10 min	10.2 d / 36 min	X / 5–72 h
Link	X / 24–48 h	97.2 d / 57.6 min	X / 5–72 h
Switch	X / 24–48 h	41.8 d / 77.2 min	X / 5–72 h
	Annual failure rate		
NIC/HCA	1%	X	$\gg$ 1%
Link	0.2%	X	0.9%‡
Switch	1.5%	X	1%

*Deimos' failure data is not publicly available
†Not enough data for accurate calculation
‡Excludes first month, i.e., failures sorted out during acceptance testing

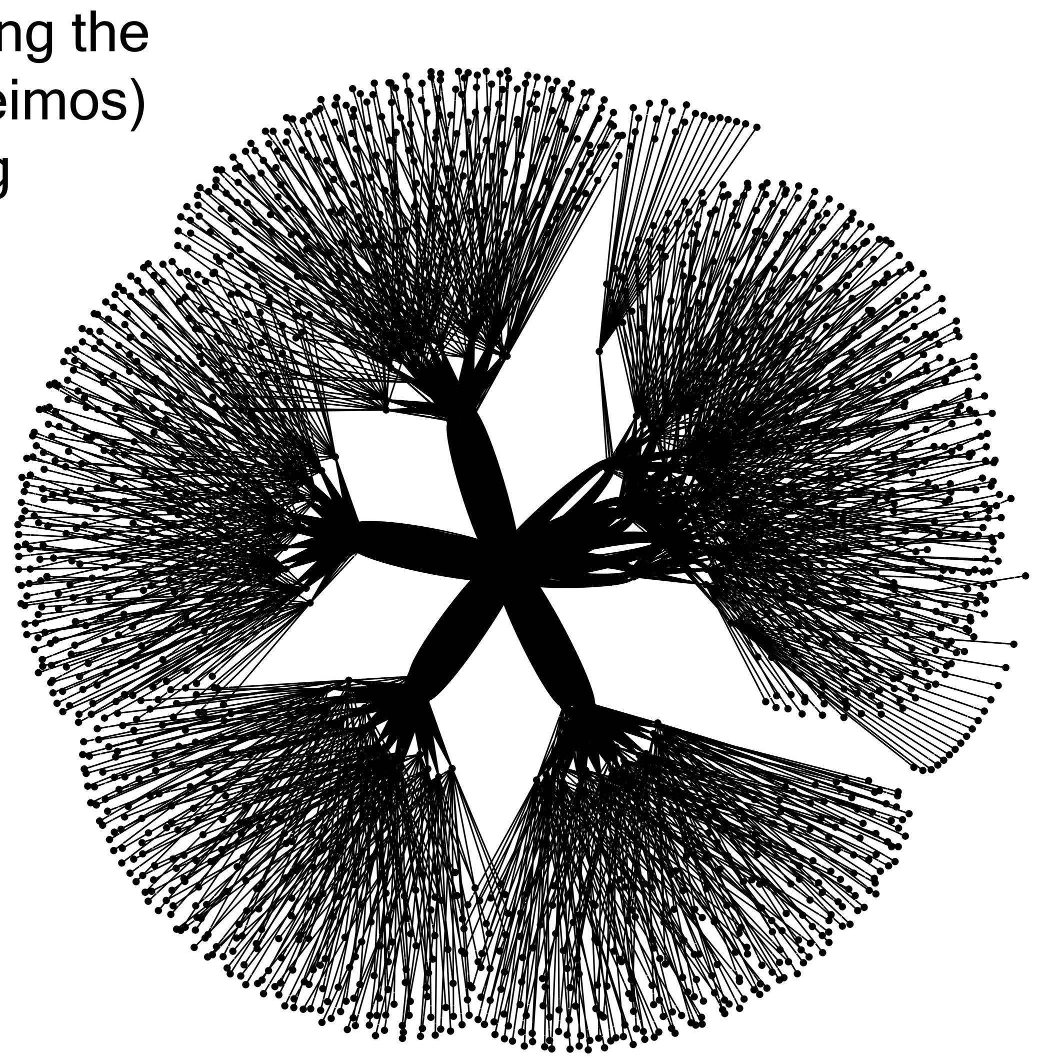

Fig.1: Network of TSUBAME2.5 1,555 nodes connected with 258 switches and 3,621 links

Toolchain

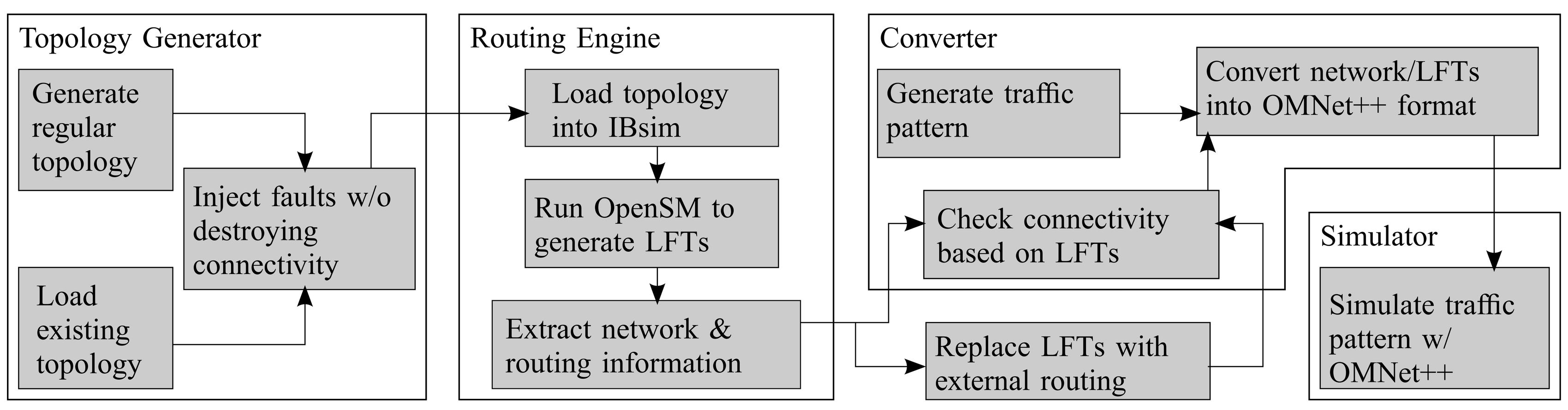

Fig.2: Simulation toolchain based on topology generation engine, IB tools, and simulation engine (OMNeT++)

Our toolchain allows system designers to plan future fail-in-place networks and operation policies while taking failure rates into consideration and allows administrators to evaluate the current state of the network by comparing it to the fault-free state.
OMNeT++ and IB model provide flit-level simulations of high detail and accuracy for two available modes

- Uniform random injection: measures consumption bandwidth at each sink (after simulation reached steady state)
- Exchange pattern of varying shift distances: determines the throughput of an MPI All-to-All

Presentation at SC14

SESSION: Hardware Vulnerability and Recovery
TIME: 2:00PM - 2:30PM on Wednesday, November 19th
ROOM: 393-94-95
AUTHORS: Jens Domke, Torsten Hoefler, Satoshi Matsuoka

Simulation Results

All topology-agnostic algorithms were able to route all small-scale topologies. However, MinHop and SSSP do not prevent deadlocks and thus create deadlocking routes in some configurations.

For deterministically routed IB fat-trees, we show that two failing links may reduce the overall bandwidth by up to 30% when a fault-tolerant topology-aware routing is used, such as Fat-Tree routing.

TABLE II. USABILITY OF TOPOLOGY/ROUTING COMBINATIONS; O : DEADLOCK-FREE; R : ROUTING FAILED; D : DEADLOCK DETECTED

	Fat-tree	Up*/Down*	DOR	Torus-2QoS	MinHop	SSSP	DFSSSP	LASH
	artificial topologies							
2D mesh	r	r	o	o	d	d	o	o
3D mesh	r	r	o	o	d	d	o	o
2D torus	r	r	d	o	d	d	o	o
3D torus	r	r	o	o	d	d	o	o
Kautz	r	r	d	r	d	d	o	o
k-ary n-tree	o	o	o	r	o	o	o	o
XGFT	o	o	o	r	o	o	o	o
Dragonfly	r	r	d	r	d	d	o	o
Random	r	r	o	r	d	d	o	o
	real-world HPC systems							
Deimos	r	o	o	r	o	o	o	o
TSUBAME2.5	o	o	o	r	o	o	o	o
	topology-aware				topology-agnostic			

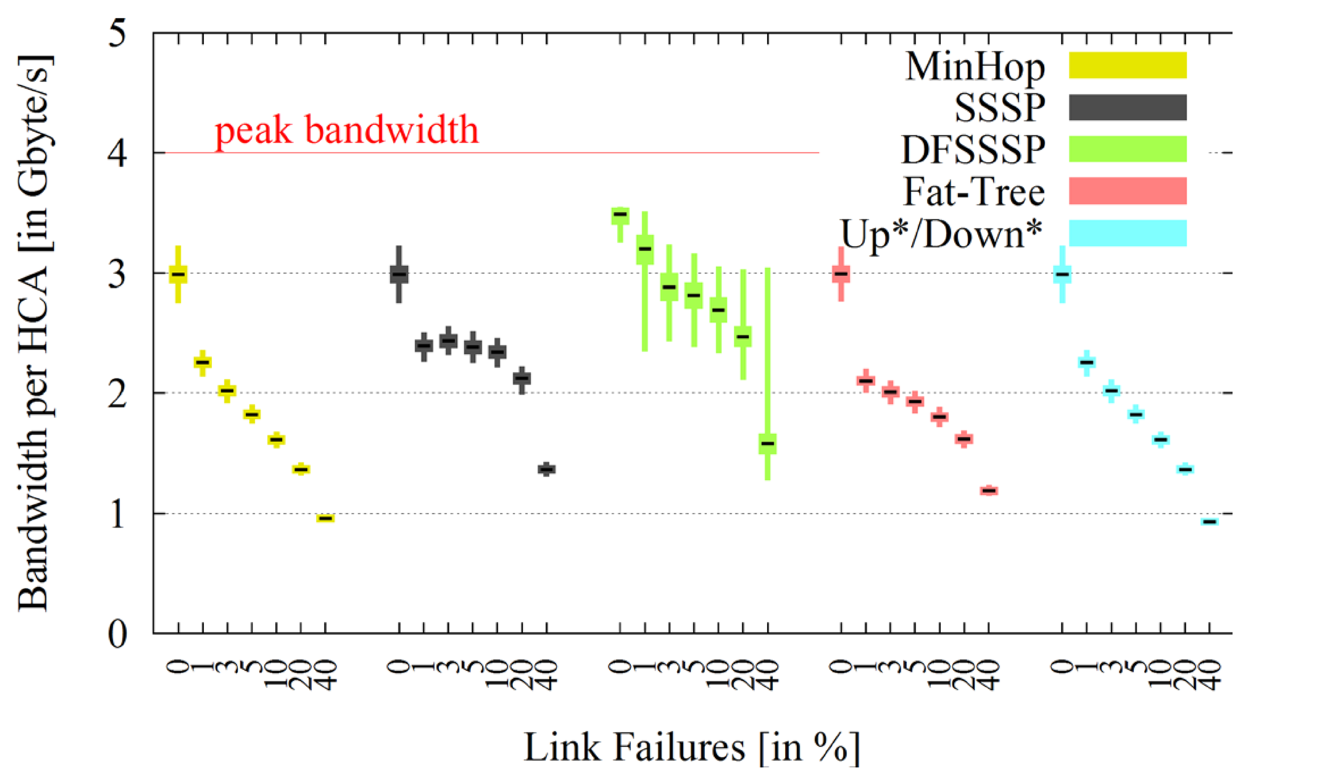

Fig.3: Consumption BW for uniform random injection assuming a 16-ary 2-tree with 256 HCAs

Changing from Up*/Down* (default) to DFSSSP routing on TSUBAME2.5 improves the throughput by 2.1x for the fault-free network and increases TSUBAME's fail-in-place characteristics.

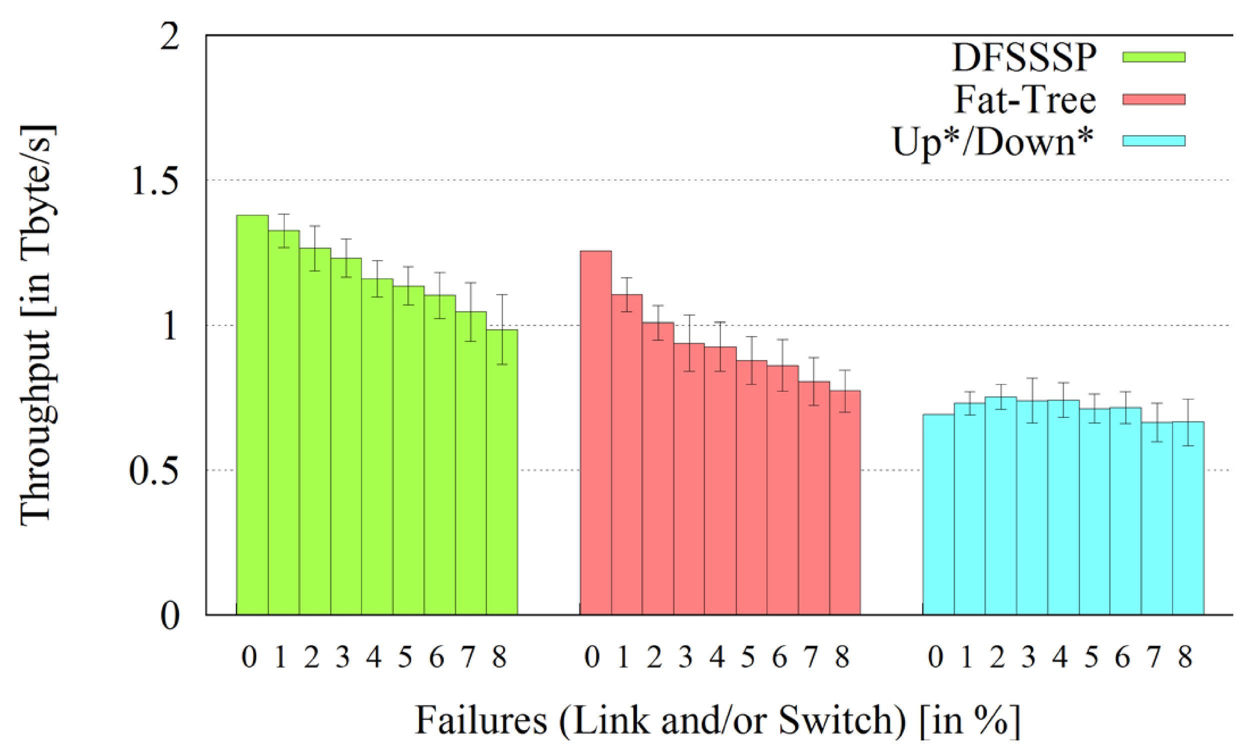

Fig.4: Throughput degradation for MPI All-to-All on TSUBAME2.5 assuming a life span of 8 years while running in fail-in-place mode

ACKNOWLEDGMENT

The authors would like to thank Eitan Zahavi for making the initial IB module for OMNeT++ publicly accessible, and the researchers at Simula Research Laboratory for their effort to port the original IB module to the newest OMNeT++ version. Last but not least, the authors would like to thank the HPC system administrators at LANL, Technische Universität Dresden and Tokyo Institute of Technology for collecting highly detailed failure data of their systems and sharing the information.

http://www.gsic.titech.ac.jp/sc14