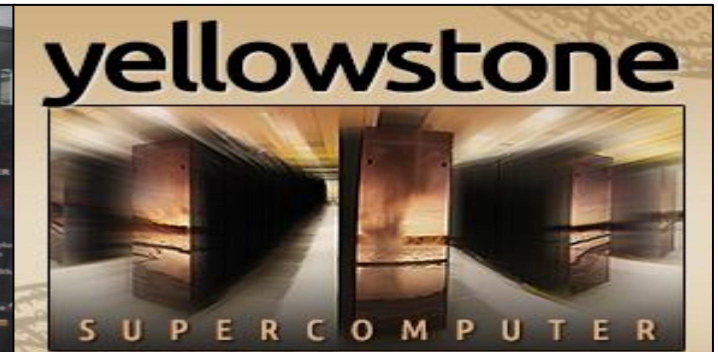
Quasi Fat Trees for HPC Clouds and their Fault-Resilient Closed-Form Routing

Eitan Zahavi*
Isaac Keslassy
Avinoam Kolodny

HOTI 2014

Technion - EE Department; *and Mellanox Technologies

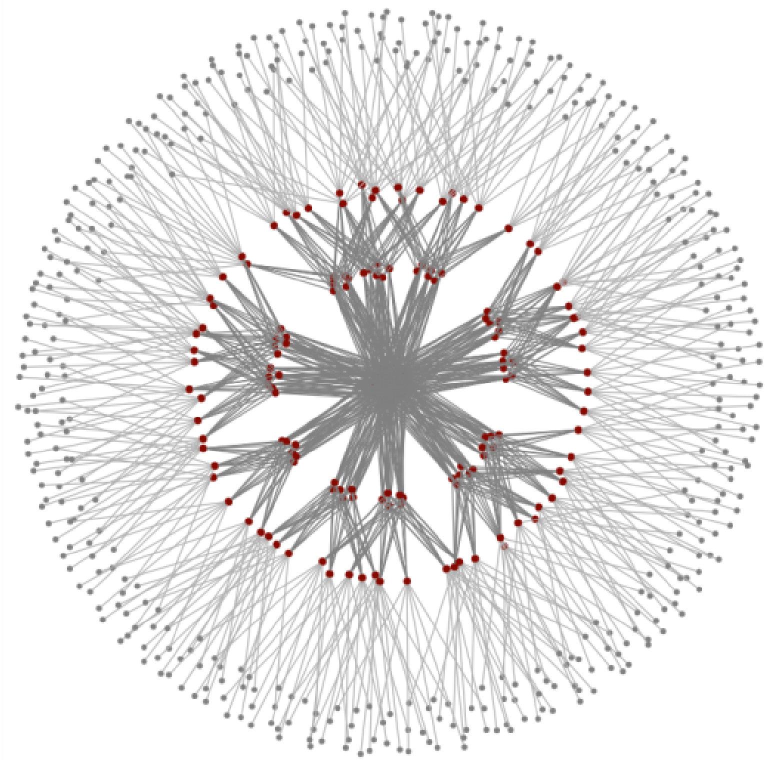
What is common to these Clusters?



- They are all Utility Clusters
- Their network topologies is ***Quasi Fat Tree***

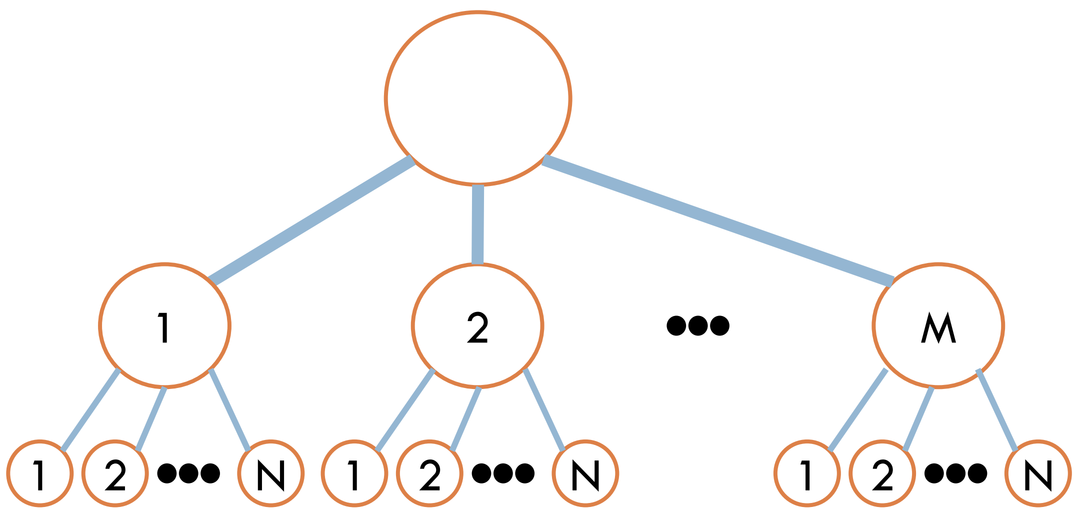
Fat Trees

- Commonly used in HPC
- Lately also in Data Centers
- Why?
  - Flexible tradeoff of BW vs. cost
  - Superior in number of minimum paths
  - Very shallow, very wide



The Fat Tree Evolution

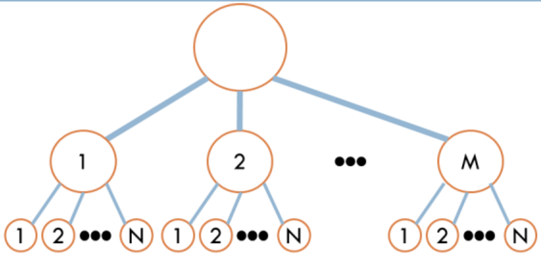
- A Tree
  - Trivial forwarding
  - Link capacity grows towards the root
    - This ensures no contention for ALL permutations

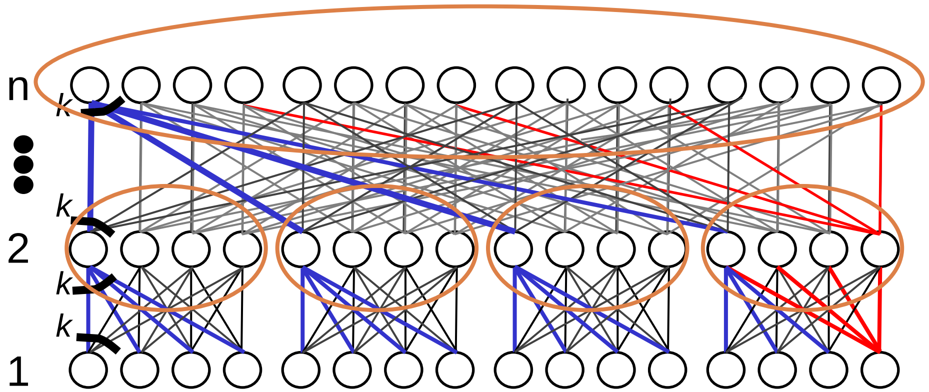


[1] C. E. Leiserson, “Fat-trees: universal networks for hardware-efficient supercomputing,” *IEEE Trans. Computing* 1985.

Multi Root Fat Trees

- E.g. k-ary n-tree
  - Scaleable since all switches/links are identical
  - ***Static Routing = some permutations are contending***
  - The constant "k" – represents half the number of switch ports
    - If same switch device is used for all levels, top switches utilize only half their ports



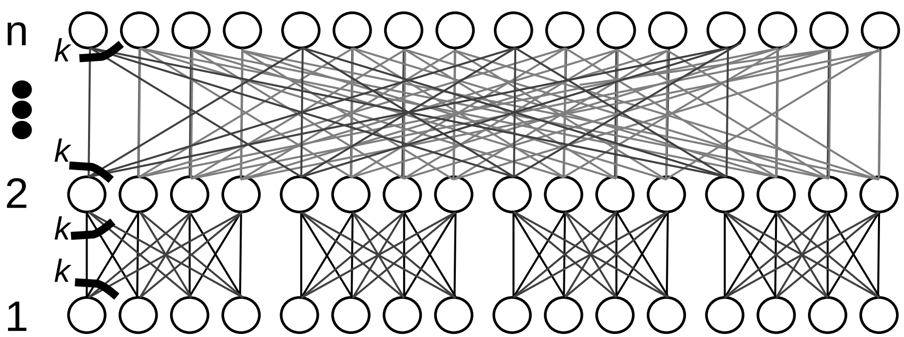


4-ary 3-tree

[2] F. Petrini and M. Vanneschi, "k-ary n-trees: high performance networks for massively parallel architectures," in *Parallel Processing Symposium, 1997.*

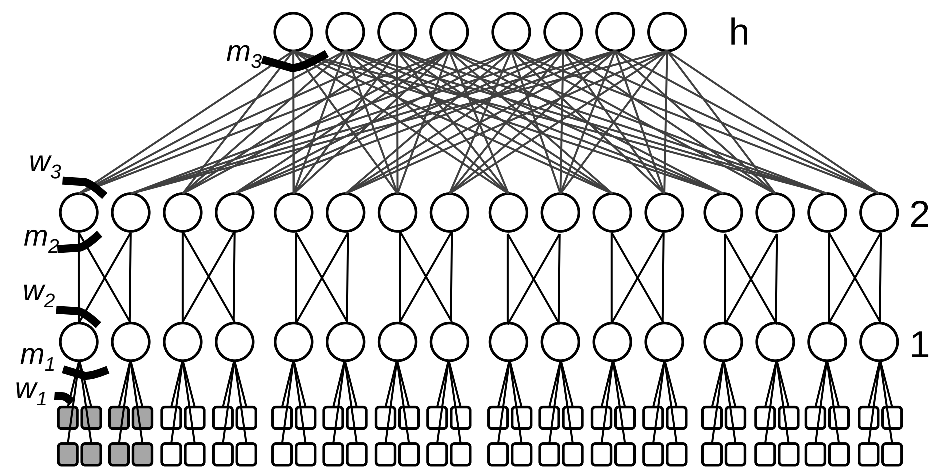
Extended Generalized Fat Tree - *XGFT*

- Allows varying of "k" per level
- Graph structure is formally defined
  - Maintains: A single link between switches; Being collection of trees
- Flexible Bisectional Bandwidth



4-ary 3-tree

$$XGFT(h;\ m_1 \dots m_h;\ w_1 \dots w_h)$$

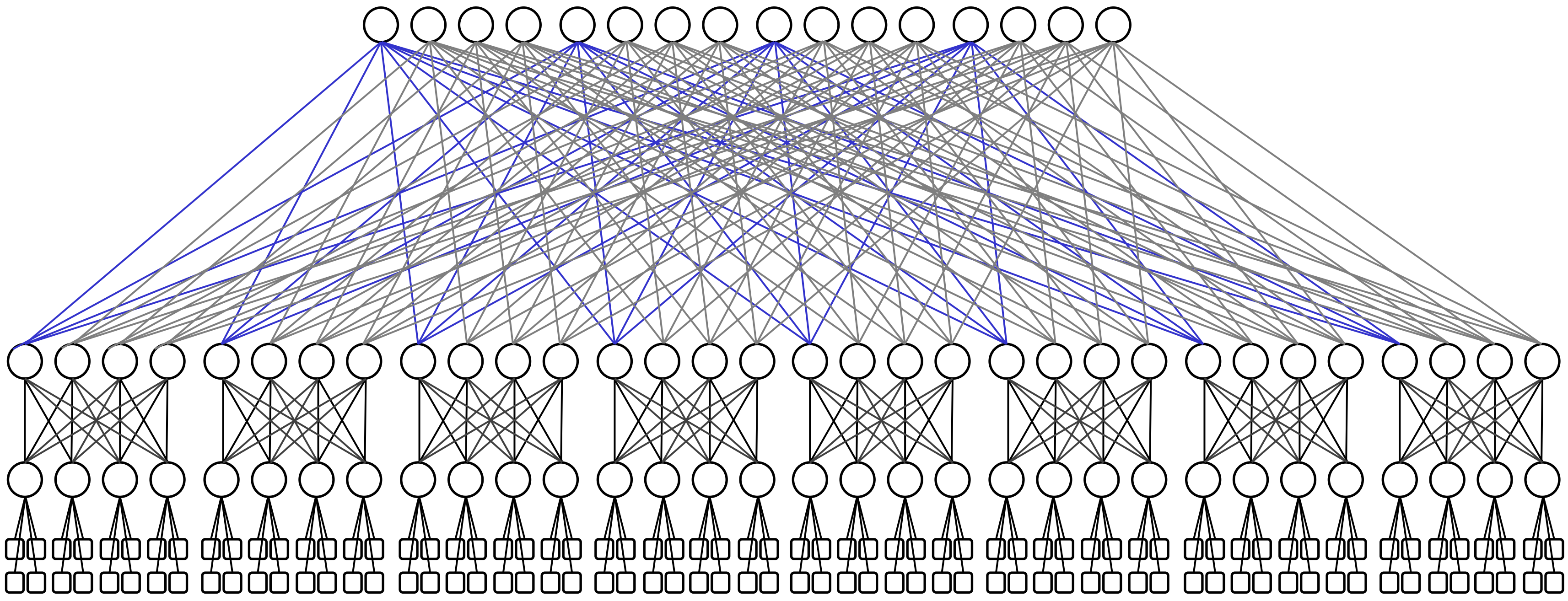


$$XGFT(3;\ 4,2,8;\ 1,2,8)$$

[3] S. R. Ohring, et al., "On generalized fat trees," in *IPPS 1995*

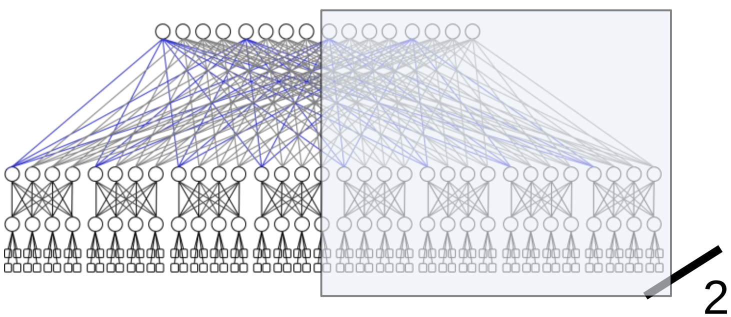
The 3 level XGFT of 8 port switches

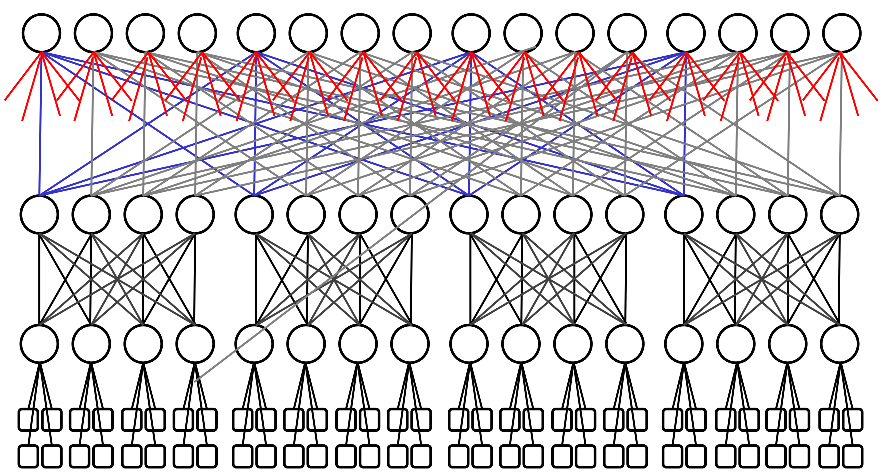
- This is the single and maximal constant bisectional bandwidth XGFT of 3 levels 8 ports switches !

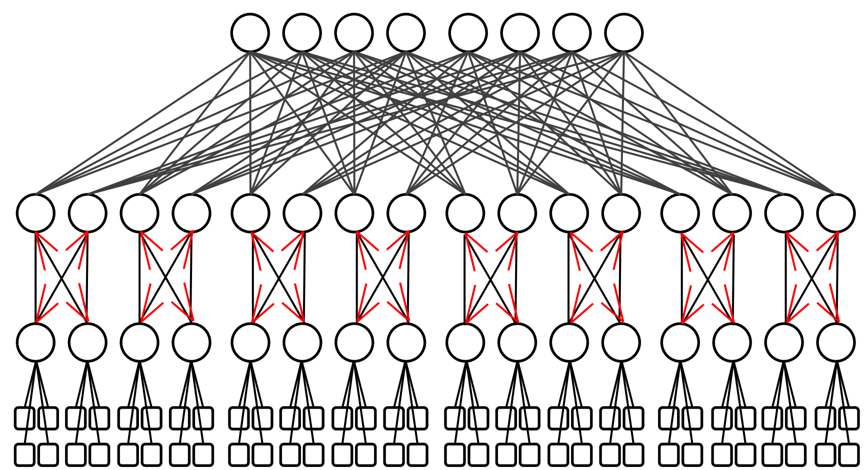


Building non-maximal topology

- Given a switch device of 8 ports, build 64 nodes XGFT tree
  - Remember only a single link is allowed between a pair of switches

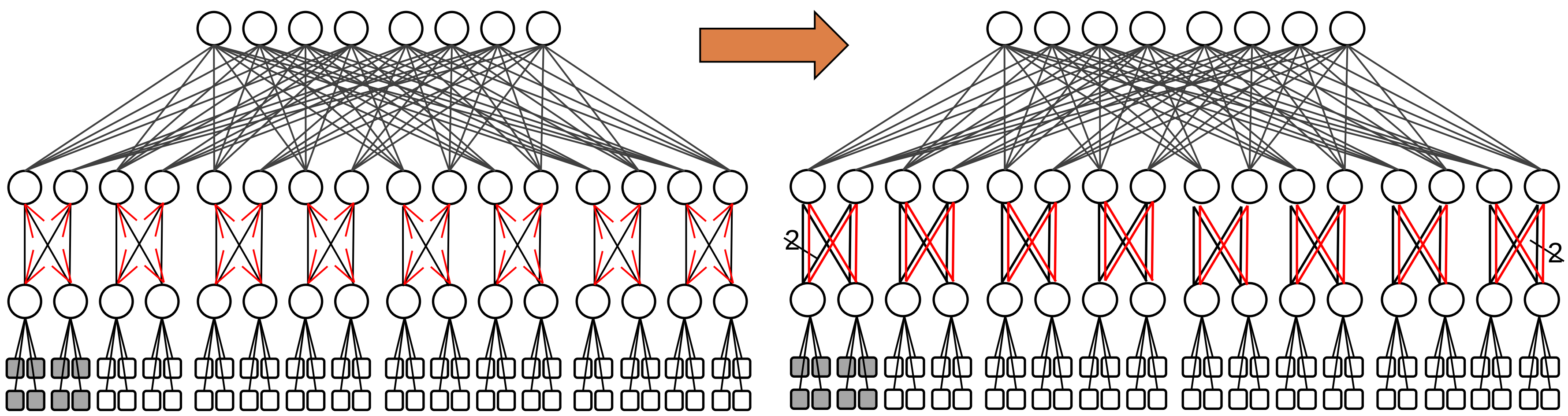






Parallel Ports Generalized Fat Trees

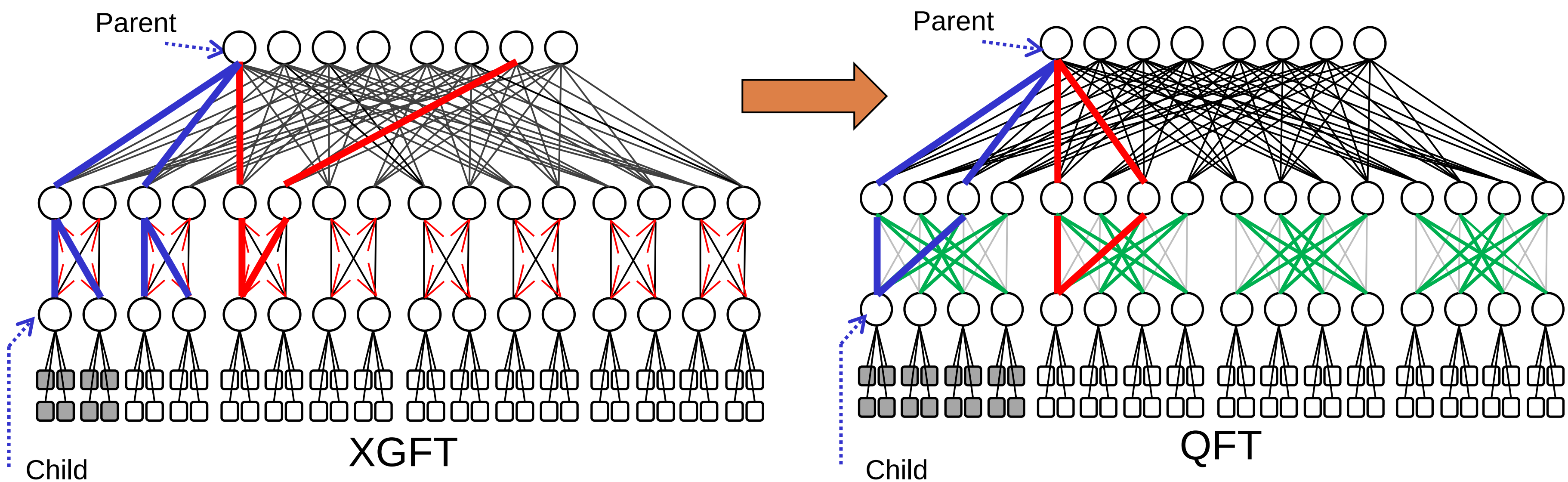
- Allow parallel ports (introduce port objects and numbering)
- XGFT has dangling 2 ports for each levels 1 or 2 switches
- PGFT allows parallel links – so now all ports being used



XGFT PGFT

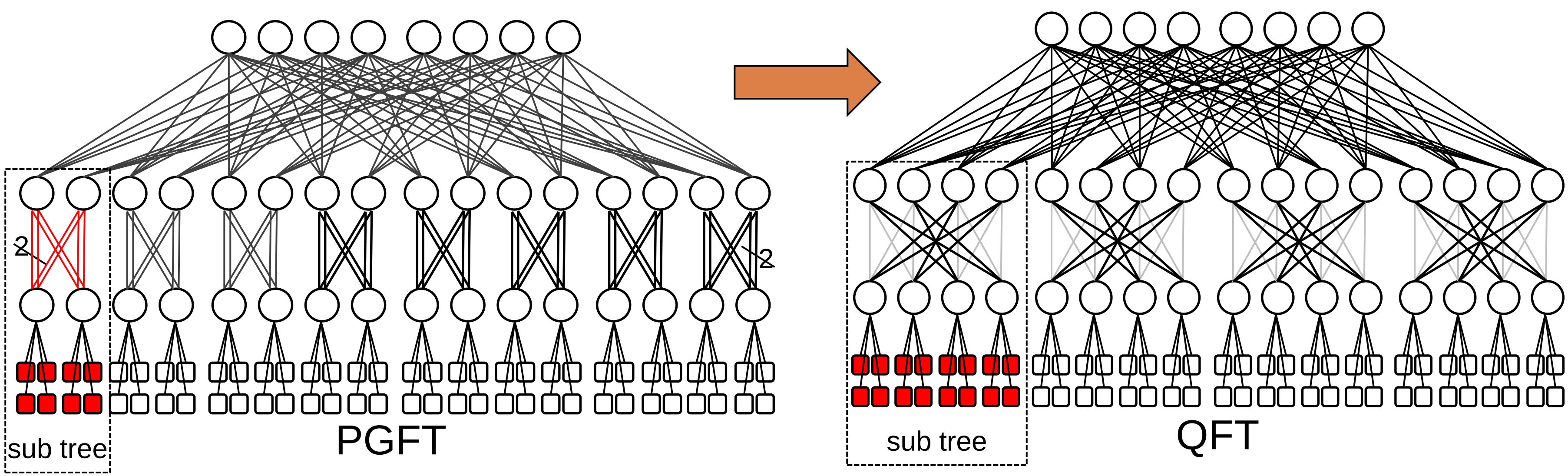
The Quasi Fat Tree

- Provides better use for the extra links:
  Connect to side branches and *reduce average diameter*
- QFT is no longer a collection of trees
- On Fat Tree there is a single path* from parent switch to any of its children. On QFT there are several such paths



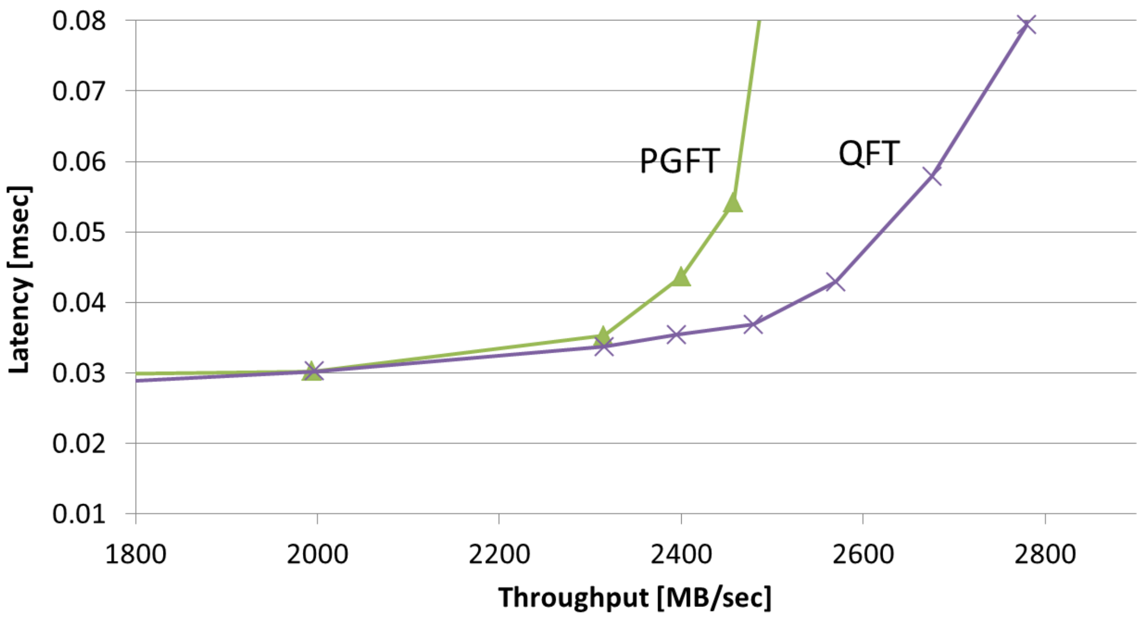
QFT reduces the Effective Diameter

- Smaller jobs see higher impact on their effective diameter
- If jobs fits into a sub-tree it gets lower latency and higher BW
- **QFT provides the maximal number of hosts in a sub-tree**
  - It is independent of cluster size
  - PGFTs sub-tree size is reduced with the cluster size



12 Jobs on PGFT/QFT

- An experiment to show the impact of the larger QFT sub-trees
  - A 4536 nodes 3 level PGFT and QFT Fat Trees
  - 12 jobs of job size Normal(300,20)



- Realistic placement of continuous fragments of TruncNormal(36,10)
- Traffic is random Shift Permutations
- MPI is sensitive to Max Latency / Min BW

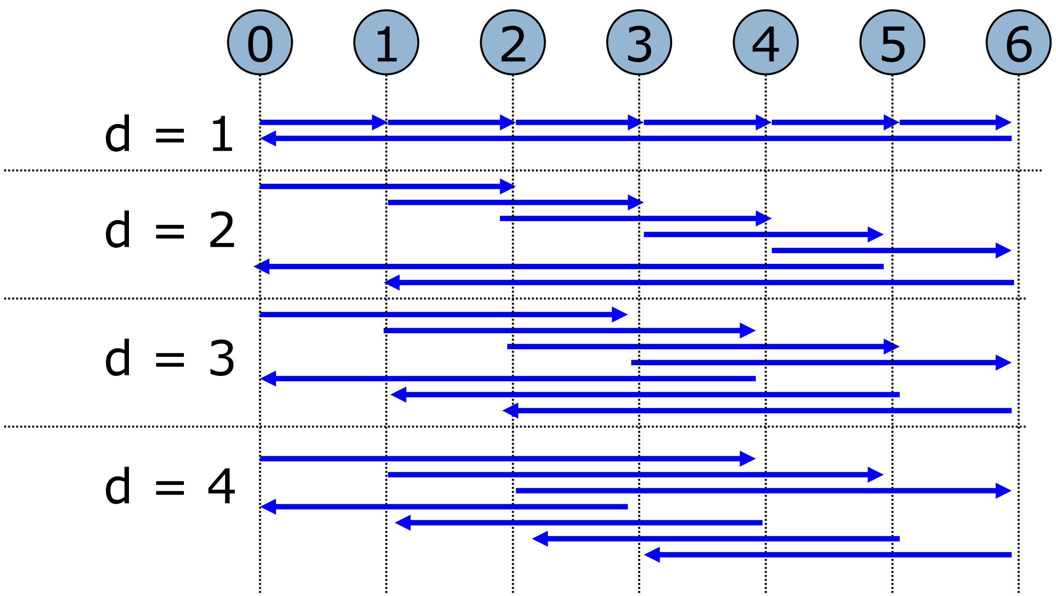
From Many Small Jobs to Single Large Job

So far, we showed why QFT provides better performance then other Fat Trees for many concurrent small jobs

Can QFT compete with other Fat Trees for single largest MPI job performance?

MPI Collectives mostly use Shifts

- Collectives are used by most applications for best performance
- Shift permutations are the traffic pattern of ALL the Collectives
  - In most MPI implementations for medium and large messages
  - In every distance "*d*" the permutation is: Dst = (Src + *d*) % N



[4] E. Zahavi, "Fat-Trees Routing and Node Ordering Providing Contention Free Traffic for MPI Global Collectives," IPDPS CASS, 2011

Routing for QFT

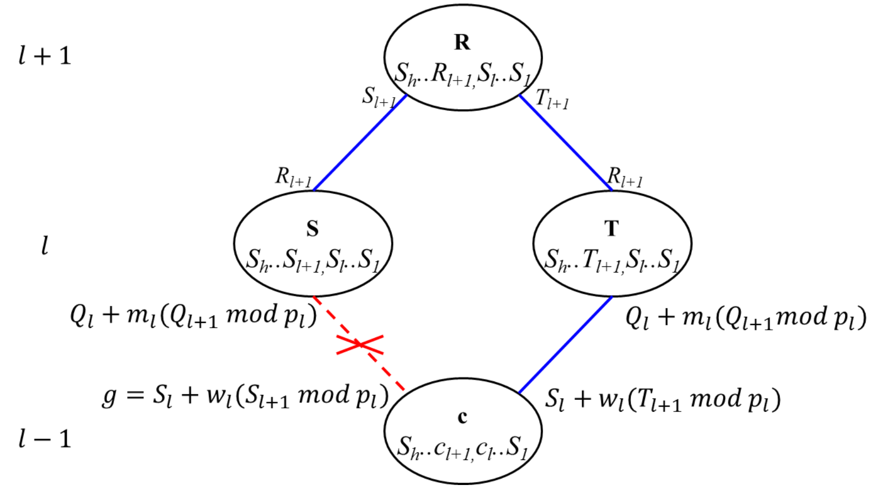
- I.e. how to forward packets in the network (not L3)
- It is non-contending for
  - All Shift Pattern i.e. ALL MPI Collectives
  - FULL network jobs
- Closed Form Routing is a formula: $Port = f(destination)$
- Significant runtime reduction compared to "traversal" based algorithms like DFSSSP, "updn" or the "ftree" engine

Routing is a Formula

- Define: $R'_l = \prod_{i=1}^{l} \frac{w_i p_i}{p_{i-2}}$
- From switch S to host Q (with index d)
- Descendant Criteria: $(\forall i \in \{l+2..h\}\ S_i = Q_i) \wedge \left(\left\lfloor \frac{S_{l+1}}{p_l} \right\rfloor = \left\lfloor \frac{Q_{l+1}}{p_l} \right\rfloor \right)$
- The sub-group used 2 levels below: $u = \frac{\left\lfloor \frac{d}{R'_{l-2}} \right\rfloor mod\ (w_{l-1} p_{l-1})}{w_{l-1}}$
- Up-Port: $g = \left(\left\lfloor \frac{d}{R'_l} \right\rfloor p_{l-1} + u \right) mod\ (w_{l+1} p_{l+1})$
- Down-Port: $f = \begin{cases} d_l & 1 = p_l \\ d_l + m_l (d_{l-1} mod\ p_l) & 1 < p_l \wedge l = h \\ d_l + m_l (d_{l+1} mod\ p_l) & 1 < p_l \wedge l < h \end{cases}$

Fault Resiliency

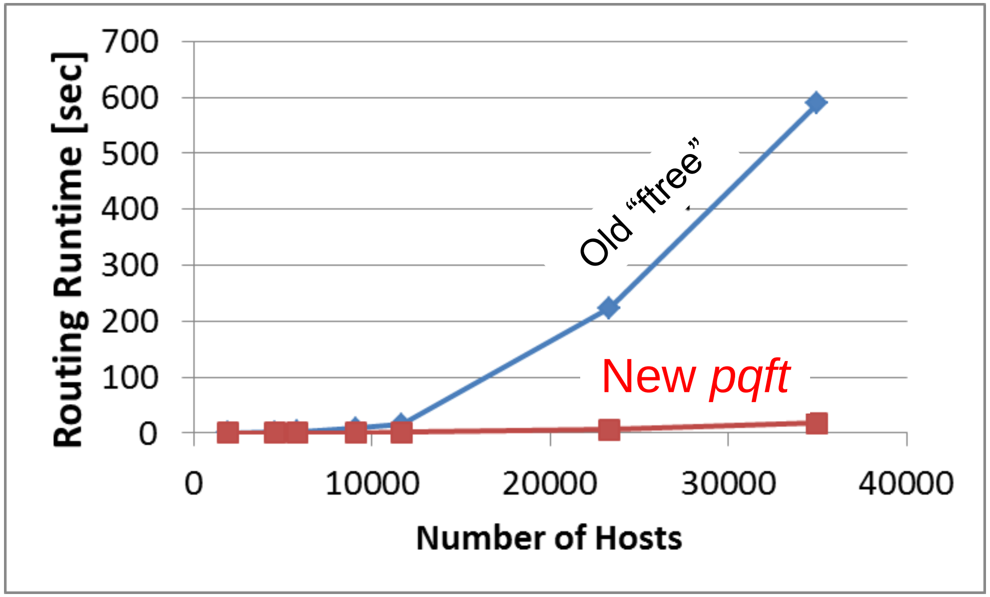
- For XGFT there is a single path down from a switch to host
- So losing a link between level l and $l+1$ means all switches at level l must avoid using matching link
- XGFT Fault Resiliency:
  - Collect all missing links
  - Calculate closed form routing
  - If source or destination switches miss that link randomly choose available (on both switches) link
- QFT have many down paths
  - They are formally recognized
  - Only if all are lost – use the Fat Tree algorithm above



[5] Zhu Ding,et al, “Level-wise Scheduling Algorithm for Fat Tree Interconnection Networks,” SC 2006

Evaluation

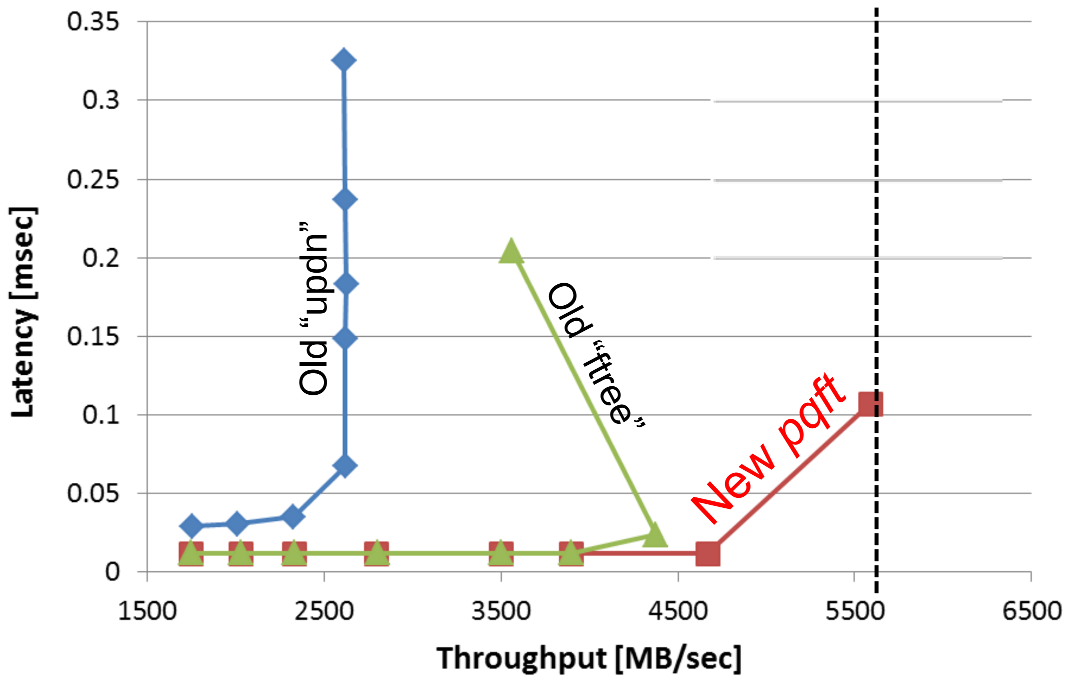
- Correctness
  - Proven as non contending [6]
  - New algorithm coded as part of OpenSM "pqft" engine
  - Verified to provide non-contention routing for all shift permutations
- Runtime improvements (of single thread implementation)



[6] http://technion.ac.il/~ezahavi/papers/qft_tr03_2014.pdf

Single Job BW Saturation

- A single MPI job random Shift permutations
- Results show much higher saturation throughput for new "*pqft*" algorithm – throughput reaches the link BW – no contention



Random Shift permutations

Conclusions

- We defined and formulate QFT
- Non-Contention routing for global shifts
- This routing is closed form and resilient

QFT outperforms other known Fat Tree flavors in both single job and multiple jobs scenarios

Thank you

Questions?