RZ 3774 (# 99784) 04/26/2010
Computer Science 7 pages

Research Report

On the Optimum Switch Radix in Fat Tree Networks

C. Minkenberg*, R.P. Luijten*, G. Rodriguez‡

*IBM Research – Zurich
8803 Rüschlikon
Switzerland

‡Barcelona Supercomputing Center
Nexus II, C/ Jordi Girona, 29
08034 Barcelona
Spain

Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

On the optimum switch radix in fat tree networks

Cyriel Minkenberg, Ronald P. Luijten
IBM Research – Zurich
Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland
{sil,lui}@zurich.ibm.com

German Rodriguez
Barcelona Supercomputing Center
Nexus II, C/ Jordi Girona, 29, 08034 Barcelona, Spain
german.rodriguez@bsc.es

***Abstract*—Based on a realistic, yet simple cost model, we compute the switch radix that minimizes the cost of a fat tree network to support a given number of end nodes. The cost model comprises two parameters indicating the relative cost of a crosspoint vs. a link, and the crosspoint-independent base cost of a switch. These parameters can be adapted to represent a given technology used to implement links and switches. Based on these inputs, the resulting model allows a quick evaluation of the switch radix that minimizes the overall cost of the network. We demonstrate that the optimum radix depends most strongly on the relative cost of a link, and turns out to be largely *independent* of the network size. Using a first-order cost bounds analysis based on current CMOS and link technology, our model suggests that the optimum switch radix is in the range of hundreds of ports, rather than the tens of ports being offered today by most commercial switch products.**

I. Introduction

Interconnection networks for supercomputers and data centers often employ a topology from the family of topologies collectively referred to as *fat trees*. Examples are supercomputers such as the Connection Machine CM-5 [1], IBM's Roadrunner at LANL, MareNostrum at Barcelona Supercomputing Center, and the JUROPA and HPC-FF systems at Jülich Supercomputing Centre. The fat tree is also currently the preferred topology for InfiniBand networks. Moreover, fat trees have recently attracted increasing attention for use in commercial data center networks [5], [6].

An HPC or data center installation generally comprises *end nodes*, which source and sink data (e.g., compute nodes, storage servers, database servers, etc.) and an *interconnection network*, which serves to transport data between end nodes. Because systems are becoming increasingly distributed in nature, the importance of the interconnection network has been on the rise, and is currently one of the key factors determining overall system performance. The flip side of this is that the interconnect also accounts for an increasingly substantial part of the *cost* of the overall system.

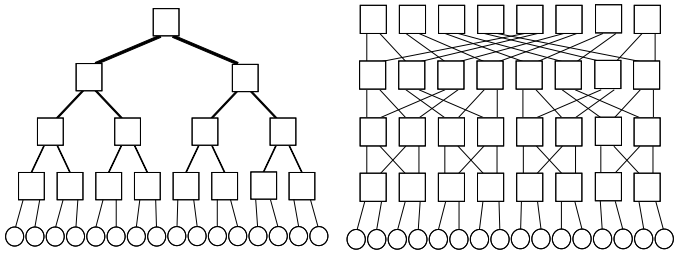
Based on this premise, this paper answers a straightforward question: To interconnect a given number of end nodes with a fat tree network, what is the switch radix r that minimizes the overall network cost? By switch radix we mean the number of ports per switch, assuming that all switches in the network have the same radix.

The authors of [7] also performed a study of switch radix optimization, but with a different objective, namely that of minimizing the end-to-end latency, based on the premise that for a given technology the aggregate switch bandwidth is a given, raising the question whether to divide this aggregate up into fewer faster ports or more slower ports. Their main conclusion was that, given technology trends, the optimum switch radix is increasing from the point of view of minimum end-to-end latency. Here, on the other hand, we focus on optimizing overall network cost rather than latency.

We first review the definition of a fat tree, or more specifically k-ary n-tree, in Sec. II. In Sec. III, we propose a cost model for the network as a whole based on a simple switch cost model that is quadratic in r. We use the cost model to obtain the optimum switch radix as a function of the number of nodes n and cost-model parameters a and b for single-sized and double-sized fat trees in Secs. IV and V, respectively. We conclude in Sec. VII.

II. Fat Trees

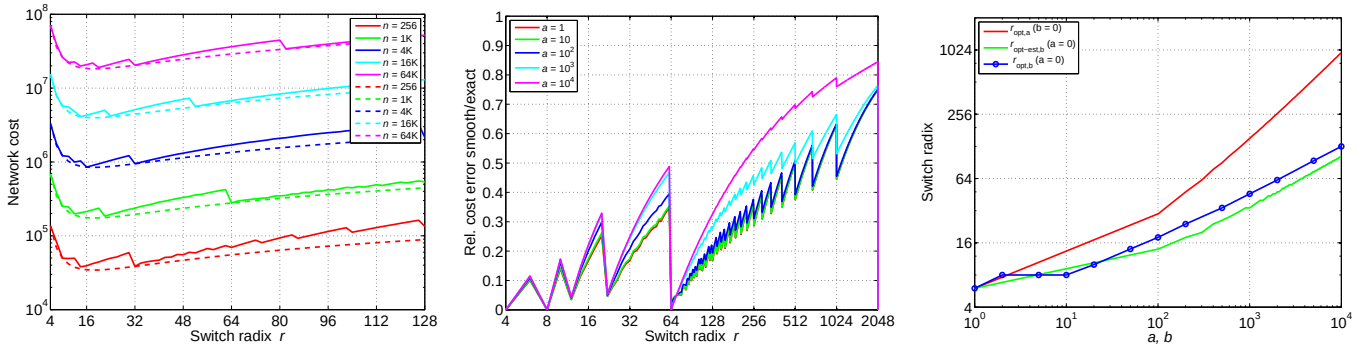
Fat tree networks were introduced by Leiserson [2] as k-ary tree topologies, in which the upward links at each level are a factor k faster than the downward links to ensure that the bisection bandwidth remains constant, see Fig. 1(a). The main problem with implementing such a tree is that the switch port rates become unmanagably high towards its root.



(a) Binary 4-level fat tree. (b) Binary 4-tree.

Fig. 1. Fat trees.

A more practical and scalable variant of such trees requiring only switches with the same radix and the same port speed at all levels are k-ary n-trees [3], see Fig. 1(b). Formally, these topologies and their slimmed versions (i.e., those not providing full bisectional bandwidth) belong to the family of *extended generalized fat trees* (XGFTs) [4]. This family includes many popular multi-stage interconnection networks, such as m-ary complete trees, k-ary n-trees, Leiserson's fat trees, and slimmed k-ary n-trees. Here, we use the term "fat tree" as a synonym for k-ary n-tree.



(a) C_{ss} vs. its smoothed version $\widehat{C}_{\text{ss}}$ for n from 256 to 64K, a = 10, b = 100.

(b) Cost error ratio $\widehat{C}_{\text{ss}}/C_{\text{ss}}$ as a function of r with a as a parameter, for n = 1,024 and b = 100.

(c) $\widehat{r}_{\text{opt,a}}$ as a function of a ($b = 0$); $\widehat{r}_{\text{opt,b}}$ and $r_{\text{opt-est,b}}$ as a function of b ($a = 0$).

Fig. 2. Smoothed cost function $\widehat{C}_{\text{ss}}$.

An XGFT$(h; m_1, ..., m_h; w_1, ..., w_h)$ of height h has $h+1$ levels, divided into $N = \prod_{i=1}^{h} m_i$ end nodes (leaves of the tree) at level $l = 0$, and switching nodes at levels $1 \le l \le h$ (inner nodes of the tree). Each non-leaf node in level i has m_i children, and each non-root has w_{i+1} parents [4]. XGFTs are constructed recursively, with each sub-tree at level l having parents numbered from 0 to $(w_{l+1} - 1)$.

III. Cost Model

Here, we will specifically consider k-ary l-trees,[1] which can be described using the XGFT specification shown above as XGFT$(l; k, \ldots, k; 1, k, \ldots, k)$. In such a network, the radix r of each switch at level $1 \le i < l$ equals $r = m_i + w_{i+1} = 2k$. We assume that the switches at level l also have radix $r = 2k$, with the upward-facing ports being unconnected to allow for future network extension. Later on, we will use the term "single-sized" fat tree to distinguish this network from the "double-sized" one in which the upward ports of the top stage are used to connect k^{n-1} additional subtrees, thus doubling the total number of end nodes. Note that k-ary l-trees have constant bisection bandwidth.

The number of end nodes $N(r, l)$ supported by a (single-sized) k-ary l-tree equals

$$N(r, l) = (r/2)^l, \;\; r = 2k. \tag{1}$$

Using (1), we derive simple expressions for several basic complexity metrics for a fat tree network that supports n end nodes using switches with radix r.

$$L(r, n) = \left\lceil \frac{\log(n)}{\log\left(\lfloor r/2 \rfloor\right)} \right\rceil \tag{2}$$

$$S(r, n) = L(r, n) \cdot \left\lceil \frac{n}{\lfloor r/2 \rfloor} \right\rceil \tag{3}$$

$$I(r, n) = n \cdot (L(r, n) - 1) \tag{4}$$

$$C_{\text{sw}}(r, b) = r^2 + b \tag{5}$$

[1] In literature the term "k-ary n-tree" is normally used, but from here on we prefer to use l to indicate the number of levels and n the number of nodes.

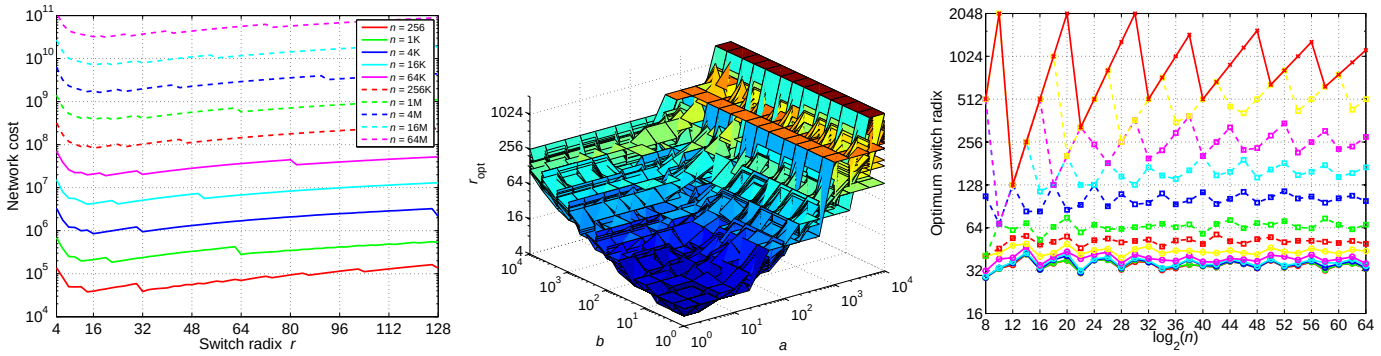
Equations (2)–(5) express the number of levels $L(r, n)$, the number of switches $S(r, n)$, the number of links $I(r, n)$, and the switch cost function $C_{\text{sw}}(r, n)$ of such a fat tree. Note that $L(r, n)$ counts only the number of *switch* levels and that $I(r, n)$ counts the number of bidirectional *inter-switch* links, not including the links between end nodes and switches, because this number is independent of the topology.

The switch cost function $C(r, n)$ is quadratic in r, under the assumption that each switching node is implemented in a single-stage fashion. The complexity of single-stage switching nodes always scales quadratically with r in some way, regardless of their specific implementation:

- The number of crosspoints in an *unbuffered* or *buffered crossbar* equals r^2.
- In an *input-queued* switch with virtual output queues (VOQs), the number of VOQs equals r^2.
- In a purely *output-queued* switch, the aggregate write bandwidth into the output queues equals r^2 times the port rate.
- In an output-queued *shared-memory* switch, the aggregate write bandwidth into the output queues equals r^2 times the packet rate times the address width. Moreover, the wiring complexity of the shared memory scales with r^2, as there is at least one memory location per port, which needs to be connected to each input and output. In practice, the wiring complexity is more likely to be in the order of $b \cdot r^3$, where b is the data-path width [8].
- In a *combined-input-output-queued* switch, a combination of the above quadratic complexities arises.

Because the term r^2 corresponds to the number of crosspoints in a crossbar, we refer to it as *crosspoint complexity*, with the cost of a crosspoint being normalized to unity. In this context the term *crosspoint* is being used generically, and should not be equated with a crosspoint in a crossbar switch. This implies that the unit crosspoint cost depends strongly on switch architecture and implementation and needs to be assessed accordingly.

To allow for some flexibility in the switch cost function and account for a fixed per-switch overhead, we include a base



(a) Cost function C_{ss} as a function of r for n from 256 to 64M, a = 10, b = 100.

(b) Optimum switch radix r_{opt} as a function of a and b with n as a parameter.

(c) Optimum switch radix r_{opt} averaged over all b as a function of n with a as a parameter.

Fig. 3. Cost function C_{ss}.

cost component b that is independent of r. Note that per-port cost components such as tranceivers can be associated with the links and should therefore be incorporated in the per-link cost factor a, which is also the reason that we do not include a linear term in the switch cost function. Parameters a and b are normalized with respect to the unit cost of a crosspoint; in practice, they can be expected to be greater than one.

Note that, although some plots may show functions as being continuous in radix r, there is clearly no practical relevance of non-integer radices. Moreover, as fat trees with constant bisection bandwidth also necessitate *even* radices, we always round up to the nearest even radix.

The overall fat tree cost function is given by (6):

$$C_{\text{ss}}(r,n,a,b) = a \cdot I(r,n) + C_{\text{sw}}(r,b) \cdot S(r,n), \qquad (6)$$

wherein a is a parameter that allows tuning of the relative cost of a link versus the relative cost of a crosspoint and a switch.

The above functions are only valid for $n \geq 2$ and $4 \leq r \leq 2n$. The case $r = 2n$ corresponds to a single switch connecting all n nodes (and having n unconnected upward ports). It does not make sense to choose $r > 2n$, because then there will be unconnected downward ports.

Figure 3(a) shows $C_{\text{ss}}(r,n,a,b)$ for a large range of n, with a = 10 and b = 100. Each curve clearly exhibits a cost minimum at around r = 16, almost independent of n.

We varied both a and b from 10^0 to 10^4 for n ranging from 2^8 to 2^{64} and determined for each combination of these parameters the optimum switch radix r_{opt} that minimizes C_{ss}. Figure 3(b) displays the results in a 3D plot, with a and b along the x- and y-axes and n as a parameter, i.e., one surface for each n. The results indicate that as a and b increase, r_{opt} also increases. In addition, the effect of increasing a is significantly stronger than that of increasing b. Moreover, these results confirm that the value of r_{opt} does not vary much with n as long as $a < 10^3$.

To illustrate this point, we averaged r_{opt} over all values of b and plotted the result as a function of n with a as a parameter, see Fig. 3(c) (a increases from the bottom to the top curve). For small values of a, the curves are almost flat. As a increases, the curves exhibit a "see-saw" pattern, which is due to the ceiling operations in (2) and (3). Nevertheless, even for large a, the amplitude of the see-saw decreases as n increases, converging on a limit value, which we will compute in the next section.

A. *Differentiable cost function*

To explore the behavior of C_{ss} in more depth, we proceed by rendering (5) differentiable by removing the ceiling and floor operations, thus eliminating the discontinuities in its derivative:

$$\widehat{L}(r,n) = \frac{\log(n)}{\log(r/2)}, \qquad (7)$$

$$\widehat{S}(r,n) = \widehat{L}(r,n) \cdot \frac{n}{r/2}, \qquad (8)$$

$$\widehat{I}(r,n) = n \cdot (\widehat{L}(r,n) - 1). \qquad (9)$$

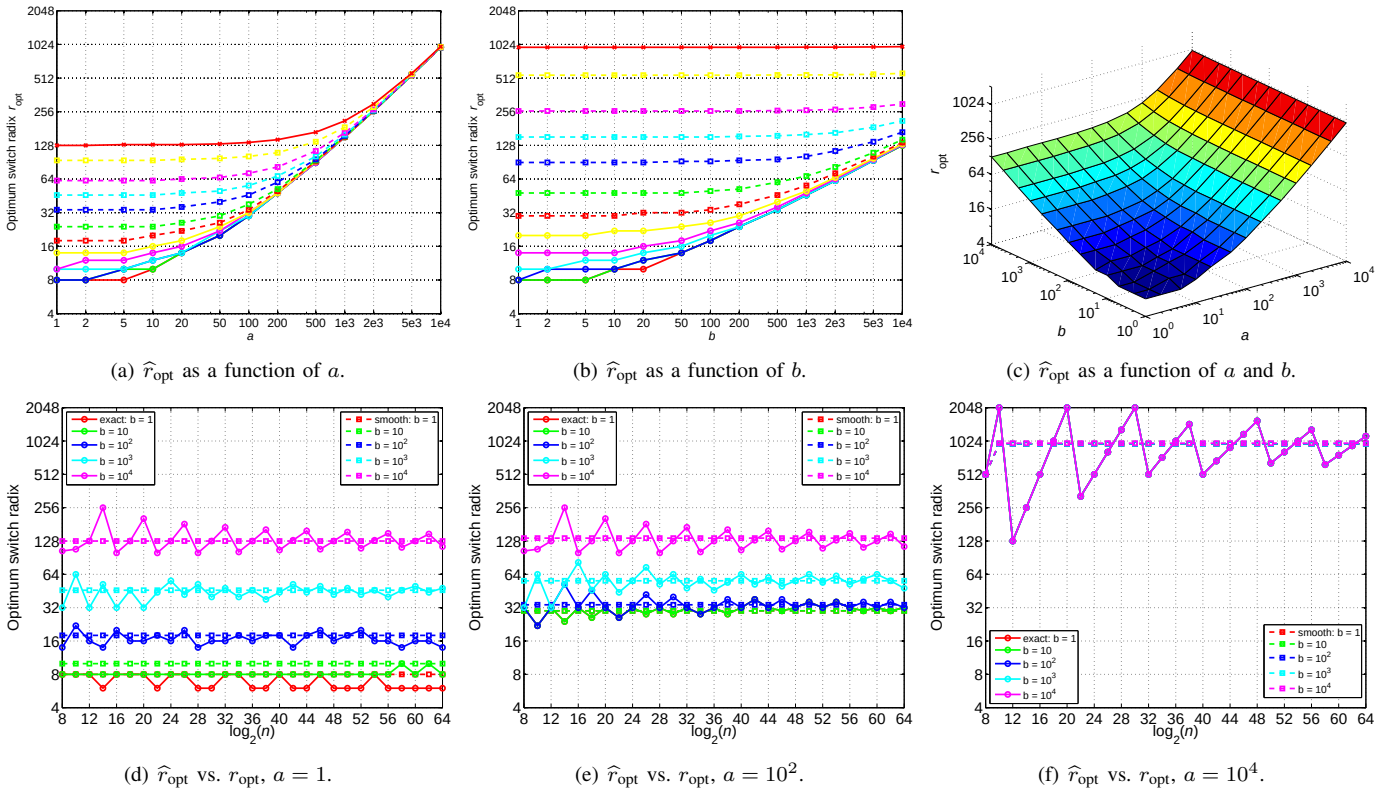
The "smoothed" cost function $\widehat{C}_{\text{ss}}(r,n,a,b)$ is given by (10):

$$\widehat{C}_{\text{ss}}(r,n,a,b) = a \cdot \widehat{I}(r,n) + C_{\text{sw}}(r,b) \cdot \widehat{S}(r,n), \qquad (10)$$

Substituting (5), (7), (8), and (9) into (10) yields (11):

$$\widehat{C}_{\text{ss}}(r,n,a,b) = a \cdot n \cdot \left(\frac{\log(n)}{\log(r/2)} - 1\right) + (r^2 + b) \cdot \frac{2 \cdot n \cdot \log(n)}{r \cdot \log(r/2)}. \qquad (11)$$

Figure 2(a) compares $\widehat{C}_{\text{ss}}$ and C_{FT} for n ranging from 256 to 64K nodes. We observe that $\widehat{C}_{\text{ss}}$ tends to underestimate the actual cost. The main cause is that the smoothing allows fractional tree levels and switches, which obviously does not correspond to reality. Figure 2(b) illustrates this effect by plotting the relative cost error $1 - \frac{\widehat{C}_{\text{ss}}}{C_{\text{ss}}}$ as a function of r for n = 1,024 and b = 100, and different values for a. It can be shown that (for $4 \leq r \leq 2n$) the error reaches its maximum at $r = 2(n-1)$ and that this maximum value approaches 0.75 for large n. However, Fig. 2(b) also shows that for realistic values of n, r, a, and b, the error can be expected to be below 20%.



(a) $\widehat{r}_{\text{opt}}$ as a function of a.

(b) $\widehat{r}_{\text{opt}}$ as a function of b.

(c) $\widehat{r}_{\text{opt}}$ as a function of a and b.

(d) $\widehat{r}_{\text{opt}}$ vs. r_{opt}, $a = 1$.

(e) $\widehat{r}_{\text{opt}}$ vs. r_{opt}, $a = 10^2$.

(f) $\widehat{r}_{\text{opt}}$ vs. r_{opt}, $a = 10^4$.

Fig. 4. Optimum switch radix $\widehat{r}_{\text{opt}}$ as a function of a and b; comparison with r_{opt}.

IV. Single-sized Fat Tree

To determine the switch radix $\widehat{r}_{\text{opt}}$ that minimizes cost function (10), we differentiate (11) with respect to r, which yields (12):

$$\frac{d\,\widehat{C}_{\text{ss}}}{d\,r} = \frac{2\cdot n\cdot\log(n)}{\log(r/2)} \cdot\left\{2-\frac{(r^2+b)\cdot(\log(r/2)+1)+\frac{a\cdot r}{2})}{r^2\cdot\log(r/2)}\right\} \quad (12)$$

and then solve

$$\frac{d\,\widehat{C}_{\text{ss}}}{d\,r}=0. \quad (13)$$

To find a solution to (13), note that the first product term of (12) is only zero when $n = 1$, which is not a meaningful solution. Therefore, to obtain useful solutions, we need to find the roots of $f_{\widehat{r}_{\text{opt}}}(r,a,b)$, as defined by (14):

$$f_{\widehat{r}_{\text{opt}}}(r,a,b) = r^2\cdot(1-\log(r/2)) +\frac{a\cdot r}{2}+b\cdot(1+\log(r/2)). \quad (14)$$

Note that $r\geq 4$, $a\geq 0$, $b\geq 0$ must hold. We first treat the special cases $b = 0$ and $a = 0$, before proceeding with the general case.

A. Case $a > 0,\ b = 0$

The special case $b = 0$ happens to have an elegant, closed-form solution for the optimum radix $\widehat{r}_{\text{opt,a}}$, given by (15):

$$\widehat{r}_{\text{opt,a}}=\frac{a}{2\cdot\mathcal{W}\left(\frac{a}{4\cdot e}\right)}, \quad (15)$$

where $\mathcal{W}(z)$ is the Lambert W-function, for which holds $z = \mathcal{W}(z)\cdot e^{\mathcal{W}(z)}$, for all complex numbers z.

Most notably, (15) is *independent* of n, i.e., the optimum switch radix depends only on the cost factor a, but not on the size of the network.

B. Case $a = 0,\ b > 0$

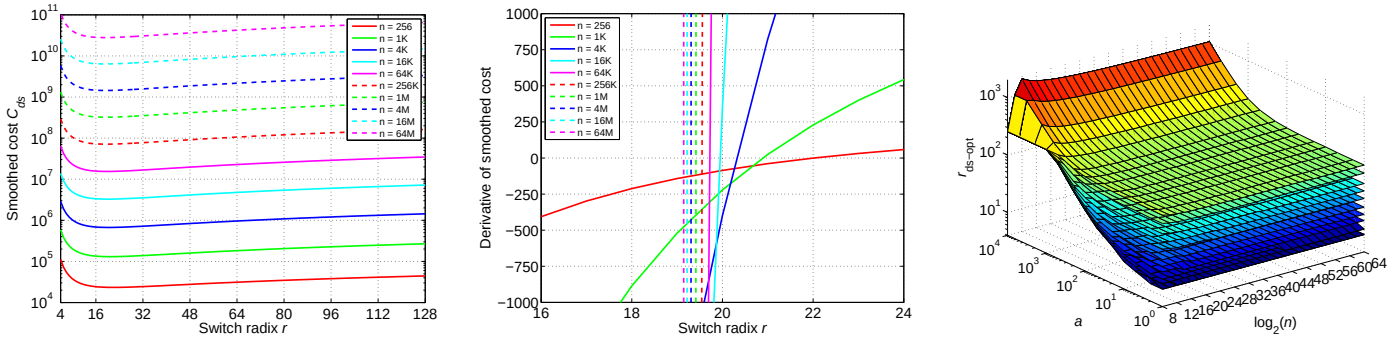
If $a = 0$, the optimum radix $\widehat{r}_{\text{opt,b}}$ can be shown to be equal to the solution to (16):

$$b=\widehat{r}^{\,2}_{\text{opt,b}}\cdot\frac{\log(\widehat{r}_{\text{opt,b}}/2)-1}{\log(\widehat{r}_{\text{opt,b}}/2)+1}. \quad (16)$$

Using a first-order approximation for $\log(x)$ around $x = 1$, we can approximate the term $\frac{\log(r/2)-1}{\log(r/2)+1}$ by $1-\frac{4}{r}$. Substituting this in (16) and solving for r yields (17):

$$\widehat{r}_{\text{opt,b}}\approx r_{\text{opt-est,b}}=2+\sqrt{b+4}. \quad (17)$$

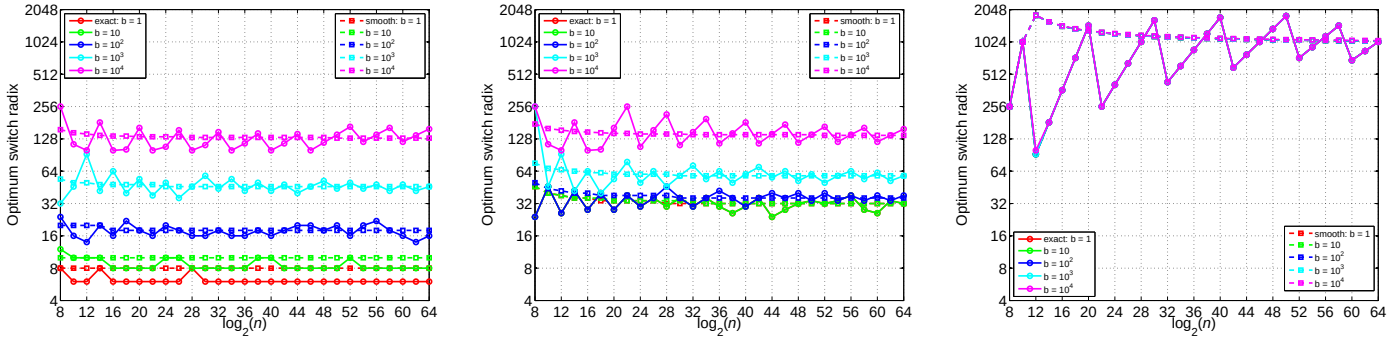
Figure 2(c) plots $\widehat{r}_{\text{opt,a}}$, $\widehat{r}_{\text{opt,b}}$, and $r_{\text{opt-est,b}}$. We exploited the fact that $\widehat{r}_{\text{opt,a}}$ and $\widehat{r}_{\text{opt,b}}$ provide reasonable estimates of $\widehat{r}_{\text{opt}}$ to seed the zero-finding process of the general case (see Sec. IV-C). This technique was also used to find the actual values for $\widehat{r}_{\text{opt,b}}$ shown in Fig. 2(c).



(a) $\widehat{C}_{\text{ds}}$ for $a = 10$, $b = 100$.

(b) $d\,\widehat{C}_{\text{ds}}/d\,r$ for $a = 10$, $b = 100$.

(c) $\widehat{r}_{\text{ds-opt}}$ as a function of n and a, with b as a parameter.



(d) $\widehat{r}_{\text{ds-opt}}$ vs. $r_{\text{ds-opt}}$, $a = 1$.

(e) $\widehat{r}_{\text{ds-opt}}$ vs. $r_{\text{ds-opt}}$, $a = 10^2$.

(f) $\widehat{r}_{\text{ds-opt}}$ vs. $r_{\text{ds-opt}}$, $a = 10^4$.

Fig. 5. Double-sized fat tree: cost function, derivate cost function, and optimum switch radix.

C. Case $a > 0$, $b > 0$

To solve the general case, we used Matlab™ to numerically find the root of (14) for any given combination of n, a, and b. As Matlab's `fzero()` function requires a seed value for the root, we seeded it with (15) if $a > b$ or with (17) otherwise.

Figures 4(a,b) show the results for n ranging from 2^8 to 2^{64} and a and b from 10^0 to 10^4. Both subplots display the same data set, but Fig. 4(a) plots r_{opt} as a function of a with b as a parameter, whereas Fig. 4(b) is the opposite. The values of a and b increase from left to right (x-axis) or bottom to top (curves within same plot).

Figures 4(d,e,f) plot $\widehat{r}_{\text{opt}}$ and r_{opt} as a function of n with b as a parameter for $a = 1$, 10^2, and 10^4, respectively. These figures clearly demonstrate that for given a and b the values of $\widehat{r}_{\text{opt}}$ and r_{opt} converge as n increases. In addition, the decreasing dependence on b as a increases is also obvious from comparing Figs. 4(d,e,f) against each other.

V. Double-sized Fat Tree

The fat tree configuration considered up to here left half the ports of the top level switches unconnected. We will now consider a fat tree in which these top-level ports are connected to another subtree of height $h-1$, so that the total number of end nodes is doubled; hence, we refer to this topology as the *double-sized* fat tree. In XGFT terms, this corresponds to an $\text{XGFT}(l; k, \ldots, k, 2k; 1, k, \ldots, k)$, with $k = r/2$. This topology gives rise to slightly different expressions for the number of nodes (18), levels (19), switches (20), and links (21):

$$N_{\text{ds}}(r,l) = r \cdot (r/2)^{l-1}, \; r = 2k, \tag{18}$$

$$L_{\text{ds}}(r,n) = 1 + \left\lceil \frac{\log(n/r)}{\log\left(\lfloor r/2 \rfloor\right)} \right\rceil, \tag{19}$$

$$S_{\text{ds}}(r,n) = (L_{\text{ds}}(r,n)-1) \cdot \left\lceil \frac{n}{\lfloor r/2 \rfloor} \right\rceil + \left\lceil \frac{n}{r} \right\rceil, \tag{20}$$

$$I_{\text{ds}}(r,n) = n \cdot (L_{\text{ds}}(r,n)-1), \tag{21}$$

with the overall cost function $C_{\text{ds}}(r,n,a,b)$ given by (22):

$$C_{\text{ds}}(r,n,a,b) = a \cdot I_{\text{ds}}(r,n) + C_{\text{sw}}(r,b) \cdot S_{\text{ds}}(r,n). \tag{22}$$

In a similar approach to Sec. IV, we derive a smoothed cost function $\widehat{C}_{\text{ds}}$ and follow a similar procedure to find its minima for given n, a, and b. In this case, $4 \le r \le n$ should hold.

Figure 5(a) shows $\widehat{C}_{\text{ds}}$ as a function of r for a large range of n and $a = 10$, $b = 100$. Upon close inspection, it appears that the minima occur at different values of n, suggesting that, unlike the single-sized fat tree, the optimum radix depends on n. Figure 5(b) confirms this notion: it plots the derivative of $\widehat{C}_{\text{ds}}$ with respect to r for the same parameters, zooming in to the range where the roots are. Here, we can clearly see how the roots move left as n increases, from about 22 for $n = 256$ down to just over 19 for $n = 64$M. However, we shall demonstrate that, as n increases, the roots converge to those of the single-sized case.

TABLE I
ESTIMATED OPTIMUM RADIX BASED ON CURRENT CMOS AND LINK TECHNOLOGY.

	low end	high end
pkts/crosspoint	8	80
bits/crosspoint	4,096	40,960
FETs/SRAM cell	6	6
FETs/crosspoint	24,576	245,760
cost/crosspoint ($)	$4.9{\cdot}10^{-4}$	$4.9{\cdot}10^{-3}$
link	4 pins	10 Gb/s optical
link cost ($)	0.4	25
overhead in gates	$1{\cdot}10^5$	$2{\cdot}10^5$
FETs per gate	4	4
overhead cost ($)	$4{\cdot}10^{-3}$	$8{\cdot}10^{-3}$
rel. link cost a	$8.1{\cdot}10^2$	$5.1{\cdot}10^3$
rel. switch ovhd b	16.3	1.63
optimum radix r_{opt}	$\sim$130	$\sim$550

Equation (23) shows the derivative of $\widehat{C}_{\text{ds}}(r,n,a,b)$ with respect to r; unfortunately, it does not have the clean product form of (12). As a consequence, the zeroes of (22) are indeed *not* independent of n in this case. Note that the optimum switch radix actually *decreases* as n increases.

$$\begin{aligned}\frac{d\,\widehat{C}_{\text{ds}}}{d\,r} &= 2n\cdot\left\{\frac{1}{2}-\frac{\log\left(\frac{n}{2}\right)}{\log^2\left(\frac{r}{2}\right)}+\frac{\log\left(\frac{n}{r}\right)}{\log\left(\frac{r}{2}\right)}\right\}\\ &\quad -\frac{2b\cdot n}{r^2}\cdot\left\{\frac{1}{2}+\frac{1+\log\left(\frac{n}{r}\right)}{\log\left(\frac{r}{2}\right)}+\frac{\log\left(\frac{n}{r}\right)}{\log^2\left(\frac{r}{2}\right)}\right\}\\ &\quad -\frac{a\cdot n}{r\log\left(\frac{r}{2}\right)}\cdot\left\{1+\frac{\log\left(\frac{n}{r}\right)}{\log\left(\frac{r}{2}\right)}\right\}. \qquad (23)\end{aligned}$$

Taking the limit $n\to\infty$ of (23) yields (24):

$$\begin{aligned}\lim_{n\to\infty}\frac{d\,\widehat{C}_{\text{ds}}}{d\,r} &= \lim_{n\to\infty}\frac{-2n\log(n)}{\log^2\left(\frac{r}{2}\right)}\\ \cdot\left\{1-\log\left(\frac{r}{2}\right)+\frac{a}{2r}+\frac{b}{r^2}\cdot\left(1+\log\left(\frac{r}{2}\right)\right)\right\}. &\qquad (24)\end{aligned}$$

Hence, for asymptotically large n we can obtain the optimum switch radix r_{opt} by finding the roots of $f_{\widehat{r}_{\text{ds-opt}}}(r,a,b)$:

$$f_{\widehat{r}_{\text{ds-opt}}}(r,a,b)=1-\log\left(\frac{r}{2}\right)+\frac{a}{2r}+\frac{b}{r^2}\cdot\left(1+\log\left(\frac{r}{2}\right)\right),$$

which by making use of (14) can be rewritten as

$$f_{\widehat{r}_{\text{opt}}}(r,a,b)=r^2\cdot f_{\widehat{r}_{\text{ds-opt}}}(r,a,b). \qquad (25)$$

Therefore, any root $r>0$ of $f_{\widehat{r}_{\text{opt}}}(r,a,b)$ is also a root of $f_{\widehat{r}_{\text{ds-opt}}}(r,a,b)$ and vice versa. It follows that for large n and given a and b the double- and single-sized fat tree networks have the same optimum switch radix.

Figure 5(c) plots $\widehat{r}_{\text{ds-opt}}$ as a function of a and n, with b as a parameter, i.e., there is one surface for each value of b. These results confirm that $\widehat{r}_{\text{ds-opt}}$ declines slightly as n increases and grows with increasing a and b.

VI. ESTIMATED OPTIMUM RADIX FOR CURRENT TECHNOLOGY

In this section we apply our theory to get an insight into what would be a good choice for the radix based on current technology. A comprehensive analysis that accurately reflects all the choices of architecture, chip technology, packaging, board- and link technology is beyond the scope of this paper. Instead, we perform a first order bounds analysis by analyzing two extreme cases. For the lower bound case we assume a low-cost, entry-level switch and link system. We further assume that it is based on a buffered-crossbar architecture which is implemented in a chip. For a buffered crossbar the cost is dominated by the crosspoint memory. To achieve entry-level performance we assume an 8 packet crosspoint memory with a packet length of 64 bytes. To calculate the cost, we use the transistor cost of $1e-8 for current CMOS technology [9]. We assume SRAM technology using 6 transistors per memory bit. To account for the additional SRAM control and crosspoint control and arbitration we simply double the crosspoint cost. For the chip overhead we assume 100k gates that implement configuration, control, and other overheads. For our entry-level link technology, we only account for the per-pin cost of the chip package. We assume a high-pin-count package required for switching chips having a cost of around 10 cents per pin.

For the second bound, we investigate a high-end system consisting of a high-performance switch using optical link technology. We assume a crosspoint with 10 times the packet storage capacity of the entry-level system. Here, we assume $25 cost for a 10 Gb/s connection as provided with current technology.

Table I shows the resulting costs in dollars of crosspoint, link, and switch overhead for both switch implementations. Based on these numbers, we can compute the link cost and switch overhead in relation to the crosspoint cost, corresponding to the model variables a and b. This enables us to determine the optimum switch radix using the model presented in Sec. IV. Thus, for the entry-level we obtain an optimum radix of about 130, whereas for the high-end case it is around 550. The latter result is overly optimistic, because such a large switch is no longer feasible in a single chip. However, we do conclude that the optimum radix should be in the order of several hundred ports, somewhere inbetween the two extreme cases.

Our bounds analysis, admittedly presenting an oversimplified version of reality, clearly shows that even in the low-end case, where the link costs are only a few cents, the optimum radix is quite high. For the high-end case, with substantially higher link costs, closer to the reality of today's systems, the radix is even higher than the entry level case.

At present, switch vendors offer radices in the order of a few tens of ports only. We believe this is a result of optimizing the cost of a single chip, rather than optimizing cost at the system level. HPC system designers, however, base their technology choices on system cost. The cost of transistors will continue to decrease exponentially, whereas the cost of links generally

decreases at a much slower pace. Given our analysis, the radices of current switch products are too small. We believe that a 200–400 port single-stage 10G Ethernet L2 switch for would offer optimum cost in medium to large data centers and supercomputers.

VII. Conclusions

Using a straightforward cost model for fat tree (k-ary l-tree) topologies with full bisectional bandwidth and same-radix ($r = 2k$) switches at every level, we demonstrated that the optimum switch radix r_{opt} is *independent* of the number of end nodes n, with $n = k^l$. In the "double-sized" case ($n = 2k^l$), the optimum radix does depend on n: interestingly, it slightly *decreases* as n increases, converging on the optimum for the single-sized case ($n = k^l$) as $n \to \infty$.

Regarding the relative cost parameters a and b, which represent the fixed per-link and per-switch costs, we observed that the optimum radix increases as either a or b increases. Moreover, the optimum radix is more sensitive to a; as a increases, the dependence on b decreases.

Specific values for a and b depend on the implementation of the switching nodes and links, including choices for technology and architecture. We derived values for a and b using a cost analysis based on current CMOS and link technology under some reasonable assumptions on switch implementation, which showed that the optimum switch radix is currently in the range of hundreds of ports. Moreover, as historically link costs have decreased at a much slower rate than logic gate costs, a can be expected to increase further, implying that the optimum switch radix can be expected to grow as well.

These results should be of interest to switch manufacturers, who, based on an assessment of values for a and b based on the target technology for a specific switch implementation, can determine which switch radix is most attractive. This applies in particular to networking technologies such as InfiniBand, 10G Ethernet (also known as Convergence Enhanced Ethernet (CEE)), Myrinet™, and QsNet™, which are networking technologies often employed in HPC and DC installations. Our work indicates that optimizing costs at the level of the individual (single-stage!) switch, which leads to the current commercial offerings radices in the range of tens of ports, is clearly sub-optimal from a cost perspective at the system level.

Of course, the design of an interconnection network is not only driven by cost; for instance, requirements regarding performance (latency, bandwidth) and power may impose certain boundaries that limit the choice of switch radix. As an example, for a fixed n, a larger switch radix will reduce the number of stages and thus reduce latency. However, fat trees generally have a relatively low diameter (compared to k-ary n-meshes and -cubes) and per-hop zero-load latency can be minimized by using cut-through switches, so the latency penalty of adding stages is small.

Nevertheless, as the cost of the interconnection network accounts for a substantial share of the total cost of a supercomputer or data center and this share can be expected to grow, optimizing interconnect cost is a worthwhile endeavor.

We intend to continue this work by determining reasonable estimates for a and b based on actual commercial switch implementations, as well as apply the same methodology to other popular topologies, such as k-ary n-meshes and k-ary n-cubes.

References

[1] C. E. Leiserson, "The network architecture of the Connection Machine CM-5," in *Proc. 4th Annual ACM Symposium on Parallel Algorithms and Architectures*, San Diego, CA, Jun. 1992, pp. 272–285.

[2] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Transactions on Computers*, vol. 34 , no. 10, Oct. 1985, pp. 892–901 .

[3] F. Petrini and M. Vanneschi, "k-ary n-trees: High Performance Networks for Massively Parallel Architectures," in *Proc. 11th Int'l Parallel Processing Symposium (IPPS '97)*, Geneva, Switzerland, Apr. 1997, pp. 87–93.

[4] S. R. Öhring, M. Ibel, S. K. Das, and M. J. Kumar, "On generalized fat trees," in *Proc. 9th International Symposium on Parallel Processing (IPPS '95)*, Washington, DC, Apr. 25–28, 1995, pp. 37–44.

[5] N. Farrington, E. Rubow, and A. Vahdat, "Data center switch architecture in the age of merchant silicon," in *Proc. 17th IEEE Symposium on High-Performance Interconnects (HOTI 17)*, New York City, New York, Aug. 2009.

[6] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: A scalable fault-tolerant layer 2 data center network fabric," in *Proc. ACM SIGCOMM '09*, Aug. 2009, Barcelona, Spain.

[7] J. Kim, W. J. Dally, B. Towles, and A. K. Gupta, "Microarchitecture of a high-radix router," in *Proc. ISCA 2005*, Madison, WI, Jun. 2005, pp. 420-431,.

[8] C. Minkenberg, "On packet switch design,", Ph.D. thesis, Eindhoven University of Technology, Sep. 2001.

[9] R. Kurzweil, "The web within us: when minds and machines become one," July 2009 (page 25, www.slideshare.net).