Torus-connected Toroids: An Interconnection Network for Massively Parallel Systems

Antoine Bossard
Graduate School of Science, Kanagawa University
Tsuchiya 2946, Hiratsuka, Kanagawa, 259-1293 Japan
ORCID: 0000-0001-9381-9346

ABSTRACT

Modern supercomputers are massively parallel systems which include hundreds of thousands of compute nodes. To maximise performance, connection of these nodes is critical. To this end, torus-based interconnection networks have proven popular. For instance, the world no. 1 supercomputer as of May 2022, the Fujitsu Fugaku machine, is based on the torus topology. On the other hand, the network dimension of such torus-based devices is kept rather low, for instance three for the Fujitsu Fugaku supercomputer. So, in order to connect the numerous compute nodes with such a conventional approach, the arity of the torus is increased, which is at the cost of a greater diameter. Additional layers can also be introduced, thus inducing hierarchical interconnection networks (HIN). In this paper, we describe a novel topology for the interconnection network of massively parallel systems: torus-connected toroids. The merit of this network compared to existing topologies is that while retaining the popular torus structure, it smartly combines it with another layer, toroids, to reduce the diameter of the network. The proposal is formally evaluated and compared to conventional approaches to show its advantages.

KEYWORDS

Interconnect, Dependability, Supercomputer, High-Performance, Routing, Topology

1. INTRODUCTION

The TOP500 ranking lists the fastest supercomputers in the world. From these data, it can be noticed that the torus topology has been very popular as interconnection network of such massively parallel systems. For example, the world no. 1 supercomputer as of May 2022, the Fujitsu Fugaku machine (TOP500, 2021), is based on the torus topology. Several other, high-profile machines, such as the Fujitsu K – formerly world no. 1 –, and the IBM BlueGene/Q – rated the “Greenest Supercomputer in the World” in 2010’s (IBM; Scogland et al., 2013) – are based on the torus topology (Ajima et al., 2009, 2018; Chen et al., 2011).

Besides, hierarchical interconnection networks (HIN) have also proven popular as they enable to reduce the network diameter, and thus cost (Bossard and Kaneko, 2015b,c). This is the case, for instance, of the Tofu interconnection network used by the previously mentioned Fujitsu supercomputers: Tofu combines a 3-dimensional torus on the higher level with a 3-dimensional (2, 3)-ary torus on the lower level (Ajima et al., 2009). We say that the higher level connects meta-nodes. A low degree and a low diameter are two properties which are now critical considering the huge number of compute nodes included by modern supercomputers: several machines in the top ten of the TOP500 world rankings have already passed the million-node barrier.

Recognising these two trends of the supercomputing industry – tori and HINs –, and aiming at further reducing the network degree and diameter compared to, say, Tofu, we describe in this paper the torus-connected toroids network (TCT) for the interconnect topology of massively parallel systems. Designing a hierarchical interconnection network which connects meta-nodes according to an n-dimensional torus requires for routing algorithm simplicity that each meta-node consist of 2n nodes. A first attempt was made with the torus-connected cycles (TCC) topology (Bossard and Kaneko, 2015a), itself inspired by older works such as the well known cube-connected cycles (CCC) topology (Preparata and Vuillemin, 1981). One issue of the TCC topology is that since meta-nodes are structured according to the ring topology, the diameter of the network rapidly increases as the network dimension is raised. With this TCT proposal, because meta-nodes include more nodes than those of Tofu and TCC, the network arity need not be as high as in the case of TCC to connect numerous compute nodes.
2. TOROIDS

Meta-nodes in a TCT are structured according to what we call toroids. The definition and properties of a toroid are given in this section.

2.1 Definition

**Definition 1.** The node set $V$ of an $n$-toroid has $2n$ nodes ($n \geq 1$): considering the Cartesian product $P = [0, 1] \times [0, 1] \times [0, 1, \ldots, ([n - 1]/2)]$, we have $V = P$ when $n$ is even and $V = P \setminus \{([1, 0, (n - 1)/2]), (1, 1, (n - 1)/2)\}$ otherwise. Two nodes $u = (u_x, u_y, u_z)$ and $v = (v_x, v_y, v_z)$ of $V$ are adjacent if and only if one of the following conditions holds:

- $u_x = v_x \land u_z = v_z$
- $u_y = v_y \land u_z = v_z$
- $u_x = v_x \land u_y = v_y \land u_z = v_z \pm 1 \mod [n/2]$
- $n$ odd $\land u_y = v_y \land u_x \neq v_x \land u_z = v_z \pm 1 \mod (n + 1)/2 \land (u_z = (n - 1)/2 \lor v_z = (n - 1)/2)$

where $\land$ and $\lor$ denote the logical conjunction and disjunction, respectively.

As a remark, when $n$ is even, an $n$-toroid is isomorphic to a $(2, 2, n/2)$-torus, and when $n$ is odd to a $(2, 2, (n + 1)/2)$-torus with the two edges $\{(0, 0, (n - 1)/2), (1, 0, (n - 1)/2)\}$ and $\{(0, 1, (n - 1)/2), (1, 1, (n - 1)/2)\}$ contracted (Diestel, 2016). Sample drawings of $n$-toroids are given in Figure 1.

![Figure 1. Sample drawings of n-toroids (1 ≤ n ≤ 6)](image)

For the sake of simplicity, the three dimensions of a toroid are called X, Y and Z and respectively correspond to the coordinates $u_x$, $u_y$ and $u_z$ of node $u = (u_x, u_y, u_z)$. Furthermore, when $n \geq 5$, the edges that connect a node of Z coordinate 0 and one of Z coordinate $[(n - 1)/2]$ are called wrap-around edges.

2.2 Topological Properties

From Definition 1, we directly have that an $n$-toroid has degree four when $n$ is even. When $n$ is odd, the minimum degree is four and the maximum degree is five. Precisely, the $2n - 2$ nodes $(x, y, z)$ with $0 \leq x, y \leq 1$ and $0 \leq z < (n - 1)/2$ have degree four and the two nodes $(0, y, z)$ with $0 \leq y \leq 1$ and $z = (n - 1)/2$ have degree five.

The diameter of a toroid is discussed next.

**Theorem 1.** The diameter $\tau(n)$ of an $n$-toroid is as follows:

$$\tau(n) = \begin{cases} 
\lfloor n/4 \rfloor + 2 & \text{when } n \text{ is even} \\
\lceil (n - 1)/4 \rceil + 2 & \text{when } n \text{ is odd}
\end{cases}$$

**Proof.** We first consider the case $n$ is even. Without considering wrap-around edges, the maximal distance on the Z dimension is $n/2$. Considering wrap-around edges, the maximal distance on the Z dimension is $\lceil n/4 \rceil$. For $n$ odd, the maximal distance on the Z dimension is $\lceil (n - 1)/4 \rceil + 1$. Therefore, the diameter is $\lceil (n - 1)/4 \rceil + 2$. 

The maximal distance on the X dimension is 1, and 1 as well on the Y dimension. Therefore, the maximal distance in an n-toroid when n is even is \([n/4] + 2\).

Next, we consider the case n is odd. Without considering wrap-around edges, the maximal distance on the Z dimension is \((n - 1)/2\). Considering wrap-around edges but not the two nodes \((0, y, z)\) with \(0 \leq y \leq 1\) and \(z = (n - 1)/2\), that is considering a \((2, 2, (n - 1)/2)\)-torus, the maximal distance on the Z dimension is \([(n - 1)/4]\). Finally, at most two edges are required to either move on both the X and Y dimensions, or to both reach a node of Z coordinate \((n - 1)/2\) and move on the Y dimension (there is no move possible on the X dimension from or to a node of Z coordinate \((n - 1)/2\)). Therefore, the maximal distance in an n-toroid when n is odd is \([(n - 1)/4] + 2\).

### 2.3 Shortest-Path Routing

In this section, we describe and evaluate a shortest-path routing algorithm in a toroid. This routing method is detailed with pseudo-code in Algorithms 1 (two utility functions) and 2 (the main function). In the pseudo-code, a left-pointing arrow \(\leftarrow\) is the assignment operator whereas a right-pointing one \(\rightarrow\) denotes path concatenation (between two paths, between one path and one node or between two nodes).

**Function** \(\text{route-xy}(a, b)\)

**Input:** two coordinates of either the X or Y dimension

**Output:** a sequence of coordinates on the corresponding dimension

if \(a = b\) then return \(a\) else return \(a \rightarrow b\);

**Function** \(\text{route-z}(n, a, b)\)

**Input:** \(n\) and two coordinates of the Z dimension

**Output:** a sequence of coordinates on the Z dimension

if \(a > b\) then return \(\text{reverse}(\text{route-z}(n, b, a))\);

else

c \leftarrow a; p \leftarrow a;

while \(c \neq b\) do

if \(n\) even then

if \(b - a \geq [n/4]\) then \(c \leftarrow c - 1 \mod n/2\) else \(c \leftarrow c + 1 \mod n/2\);

else

if \(b - a \geq [(n + 1)/4]\) then \(c \leftarrow c - 1 \mod (n + 1)/2\) else \(c \leftarrow c + 1 \mod (n + 1)/2\);

\(p \leftarrow (p \rightarrow c)\);

return \(p\);

Algorithm 1. Functions for routing on one single dimension in an \(n\)-toroid

**Function** \(\text{toroid-spr}(n, u, v)\)

**Input:** \(n\), a source node \(u = (u_x, u_y, u_z)\) and a destination node \(v = (v_x, v_y, v_z)\)

**Output:** a shortest path from \(u\) to \(v\)

if \(n\) even then

\(p_x \leftarrow \text{map}(\lambda: x \mapsto (x, u_y, u_z), \text{route-xy}(u_x, v_y))\);

\(p_y \leftarrow \text{map}(\lambda: y \mapsto (v_x, y, u_z), \text{route-xy}(u_y, v_y))\);

\(p_z \leftarrow \text{map}(\lambda: z \mapsto (v_x, v_y, z), \text{route-z}(u_x, v_z))\);

return \(p_x \rightarrow p_y \rightarrow p_z\);

else // \(n\) odd

if \(u_x = v_x = 0\) then

\(p_y \leftarrow \text{map}(\lambda: y \mapsto (v_x, y, u_z), \text{route-xy}(u_y, v_y))\);

\(p_z \leftarrow \text{map}(\lambda: z \mapsto (v_x, v_y, z), \text{route-z}(u_x, v_z))\);

return \(p_y \rightarrow p_z\);

else if \(u_x = v_x = 1\) then

\(p_y \leftarrow \text{map}(\lambda: y \mapsto (v_x, y, u_z), \text{route-xy}(u_y, v_y))\);
\( p_z \leftarrow \text{map}(\lambda: z \mapsto \begin{cases} (0, v_y, z) & \text{if } z = (n - 1)/2 \\ (v_x, v_y, z) & \text{else} \end{cases}) \), route-z(u_z, v_z) ; \\
\text{return } p_y \rightarrow p_z ; \\
\text{else if } u_x = 1 \text{ then} // u_x \neq v_x \land u_x = 1 \\
\text{crossed } \leftarrow \text{false}; p_z \leftarrow \emptyset ; \\
\text{foreach } z \text{ in } \text{route-z}(u_z, v_z) \text{ do} \\
\text{if } \text{crossed} \lor z = (n - 1)/2 \text{ then } p_z \leftarrow (p_z \rightarrow (0, u_y, z)); \text{crossed } \leftarrow \text{true} ; \\
\text{else } p_z \leftarrow (p_z \rightarrow (u_x, u_y, z)) ; \\
(c_x, c_y, c_z) \leftarrow \text{last}(p_z) ; \\
p_z \leftarrow \text{map}(\lambda: x \mapsto (x, u_y, v_z)), \text{route-xy}(c_x, v_x) ; \\
p_y \leftarrow \text{map}(\lambda: y \mapsto (v_x, y, v_z)), \text{route-xy}(u_y, v_y) ; \\
\text{return } p_z \rightarrow p_x \rightarrow p_y ; \\
\text{else // } u_x \neq v_x \land u_x = 0 \\
\text{return reverse(toroid-spr}(n, v, u)); \\
\)

Algorithm 2. Shortest-path routing in an \( n \)-toroid

We next empirically confirm the maximum length and investigate the average length of a path produced by this shortest-path routing algorithm in a toroid. To this end, we have implemented the algorithm and conducted a computer experiment as follows: for \( 2 \leq n \leq 8 \), the computer program solves 1000 random problem instances, that is, repeatedly generates randomly a source node and a destination node and calculates the path between them. For each such value of \( n \), the maximum path length and the average path length are calculated from the 1000 selected paths. The obtained results are illustrated in Figure 2, with the standard deviation shown for the average path length. In addition, the theoretically established diameter is plotted for reference.

![Figure 2. Experimental results of the empirical evaluation of the described shortest-path routing algorithm in an \( n \)-toroid. The maximum length and the average length of a selected path has been measured (the formally established diameter is also plotted for reference).](image)

It can be observed from these results that the measured maximum path length confirms the formally established diameter: a selected path is of length at most \([n/4] + 2\) when \( n \) is even and \([(n - 1)/4] + 2\) otherwise. Because the length of some paths is equal to the diameter, there were problem instances whose randomly selected source and destination nodes were diagonally opposed nodes.
3. TORUS-CONNECTED TOROIDS

3.1 Definition

Definition 2. An \( n \)-dimensional \( k \)-ary torus-connected toroids network, denoted as \( \text{TCT}(n, k) \), consists of \( k^n \) toroids. Two nodes \( u = (u_0, u_1, \ldots, u_{n-1}, (u_x, u_y, u_z)) \) and \( v = (v_0, v_1, \ldots, v_{n-1}, (v_x, v_y, v_z)) \) are adjacent if and only if one of the following two conditions holds:

- \( \forall i, 0 \leq i \leq n-1, u_i = v_i \land \tau_n((u_x, u_y, u_z), (v_x, v_y, v_z)) \)
- \( \exists j, \forall i, 0 \leq i \leq n-1, i \neq j, u_i = v_i \land v_j = u_j \pm 1 \text{ mod } k \land u_x = v_x \land u_y = v_y + 1 \text{ mod } 2 \land u_z = v_z \)

where the predicate \( \tau_n(a, b) \) is satisfied when \( a \) and \( b \) are adjacent nodes in an \( n \)-toroid (see Definition 1).

The above first condition induces internal edges, and the second one external edges. Part of a \( \text{TCT}(3, 3) \) is illustrated in Figure 3: for the sake of clarity, only the nodes of one single value of the third dimension are illustrated. From Definition 2, the minimum degree of a TCT is thus five and the maximum degree six.

![Figure 3. Connection scheme of TCT meta-nodes (left). Sample drawing of a 3-dimensional 3-ary TCT (right); only the nodes of one single value of the third dimension are illustrated though](image)

3.2 Simple Routing

A simple routing algorithm in a TCT between a source node and a destination node can be simply derived as follows. Compared to the toroid routing algorithm described in Section 2, this algorithm is rather straightforward and its description here is thus less detailed. We assume that \( k \geq 3 \) to omit additional, verbose details for the special cases \( k = 1 \) and \( k = 2 \), these two cases being impractical considering the numerous nodes which have to be interconnected.

First, a path of meta-nodes is selected with a dimension-order routing algorithm in a torus (Duato et al., 2003) between the meta-node that includes the source node and the meta-node that includes the destination node. Let this path be \( p: m_0 \rightarrow m_1 \rightarrow \cdots \rightarrow m_l \); it is recalled that \( m_0 \) includes the source node and \( m_l \) the destination node.

Second, apply the shortest-path routing algorithm in a toroid as described in Section 2 to find a path inside \( m_0 \) between the source node and the unique node \( u'_0 \) that is adjacent to a node \( u_i \in m_1 \). Inside \( m_i \) (\( 1 \leq i \leq l-1 \)), apply the same toroid routing algorithm to select a path between \( u_i \) and \( u'_i \) where \( u_i \) is adjacent to a node \( u_{i-1}' \in m_{i-1} \) and \( u'_i \) to a node \( u_{i+1} \in m_{i+1} \). Apply the same toroid routing algorithm to select a path inside \( m_1 \) between \( u_i \) and the destination node, where \( u_i \) is adjacent to a node \( u_{i-1}' \in m_{i-1} \).
It should be noted that it is not guaranteed with this routing algorithm that the selected path is shortest. The advantage of this algorithm is its simplicity. The maximum path length induced by this routing algorithm in a TCT is further analysed in the next section.

4. EVALUATION AND DISCUSSION

In this section, we formally and quantitatively compare the torus-connected toroids topology with the Tofu interconnection network used by the Fujitsu K and Fujitsu Fugaku supercomputers.

4.1 Comparison based on an Estimation of the Network Diameter

The topology of Tofu (Tofu 1 (Ajima et al., 2009), Tofu 2 (Ajima et al., 2014) and Tofu D (Ajima et al., 2018)) indistinctly consists of meta-nodes connected according to a three-dimensional torus. Each meta-node is in turn a (3, 2, 2)-torus, thus consisting of 12 nodes. Compared to Tofu, the node degree in a TCT is smaller: it is equal to 5 or 6 depending on the parity of the TCT dimension \( n \), and is equal to 10 in the case of Tofu.

Besides, the network dimension \( n \) is fixed to 3 for the high-level layer of Tofu – the structure of its low-level layer is constant: it does not depend on any variable. So, in the case of Tofu, the arity is solely used to increase the network order, which, as a result, induces a high diameter. We give an estimation of the network diameter of Tofu based solely on their number of nodes (i.e. the network order) and on their dimension \( n \) (\( n = 3 \) in the case of Tofu) for the sake of simplicity. That is, the network arity \( k \) is induced by uniformly distributing nodes on each of the network dimensions.

For Tofu, \( 3 \sqrt[3]{N/12}/2 + 3 = 3 \sqrt[3]{N/12}/2 + 3 \) is an estimation of the network diameter based on \( N \) and \( n \), with \( N \) the number of nodes: \( N/12 \) is the number of meta-nodes, that is, the number of (3, 2, 2)-tori which each make one node of the high-level layer of Tofu, thus multiplied by the number of dimensions, and the term 3 is for the internal edges required at either the source or destination meta-node – only one such internal routing is indeed needed in the case of Tofu.

For TCT, \( n \sqrt{N/(2n)} + 2\tau(n) \) is an estimation of the network diameter based on \( N \) and \( n \), with \( N \) the number of nodes and where \( \tau(n) \) is the diameter of a toroid of an \( n \)-dimensional TCT (i.e. of an \( n \)-toroid) as established in Section 2.2. \( N/(2n) \) represents the number of toroids per dimension, thus multiplied by \( n \) to obtain the total number of toroids. The two \( \tau \) here represent the maximum number of internal edges selected inside the toroid of the source node and inside that of the destination node. Hence, the diameter of a TCT will be smaller than that of Tofu as the network dimension \( n \) increases. The threshold value of the dimension \( n \) depends on the arity \( k \): more details are given below. But even for small values of \( n \), the degree of a TCT remains half (i.e. 5) or half plus one (i.e. 6) that of Tofu (i.e. 10).

4.2 Comparison based on the Maximum Path Length from Simple Routing

Next, we give a more detailed comparison with respect to this diameter issue. Concretely, we compare the maximum length of a path selected with a simple dimension-order (with support for torus wrap-around edges) routing algorithm in the case of Tofu and TCT. Such a simple routing algorithm has been described in Section 3.2 for the case of a TCT. The same idea can be applied to Tofu: the path selected in the high-level layer is made of external edges (i.e. edges between meta-nodes) and edges selected inside a meta-node are internal edges.

On the one hand, the case of a \( k \)-ary Tofu network.

**Lemma 1.** A simple dimension-order (with support for torus wrap-around edges) routing algorithm applied in a \( k \)-ary Tofu network induces a maximum path length \( m_u(k) \) as follows:

\[
m_u(k) = 3 \left\lfloor \frac{k}{2} \right\rfloor + 3
\]
Proof. From the Tofu topology definition, at most \( \lfloor k/2 \rfloor \) external edges are required for each dimension of the high-level three-dimensional torus. There are three dimensions, so at most \( 3 \lfloor k/2 \rfloor \) external edges are selected in total. Once again, the term 3 represents the maximum number of internal edges required inside the meta-node of either the source or destination node.

On the other hand, the case of an \( n \)-dimensional \( k \)-ary TCT network.

Theorem 2. A simple dimension-order (with support for torus wrap-around edges) routing algorithm applied in an \( n \)-dimensional \( k \)-ary TCT network induces a maximum path length \( m_\left(n,k\right) \) as follows:

\[
m_\left(n,k\right) = 2n \left\lfloor \frac{k}{2} \right\rfloor + 2\tau(n) + n - 2
\]

Proof. At most \( \lfloor k/2 \rfloor \) external edges are required on one single dimension. There are \( n \) dimensions, so in total at most \( n \lfloor k/2 \rfloor \) external edges are selected. The number of internal edges is as follows: at most 2 internal edges inside a toroid at a dimension change, at most \( \tau(n) \) for routing inside the toroid of the source node and inside that of the destination node and 1 internal edge for all the other toroids included in the selected path. There are at most \( n-1 \) toroids inside which routing is needed for a dimension change. So, in total, at most \( 2(n-1) + 2\tau(n) + (n \lfloor k/2 \rfloor + 1) - (n-1) - 2 = 2\tau(n) + n \lfloor (k/2) + 1 \rfloor - 2 \) internal edges are selected.

Because it may still be difficult to identify the values of the TCT parameters \( n \) and \( k \) for which such maximum path length would be shorter than in the case of Tofu, sample values are next given. For a fair comparison, it is needed to consider a Tofu network and a TCT network of same or near same orders. On the one hand, Tofu consists of \( N(k) = 12k^3 \) nodes. On the other hand, TCT includes \( N_\left(n,k\right) = 2nk^n \) nodes.

So, for instance, with \( n = 8, k = 3 \), a TCT connects 104,976 nodes with the length of a longest path obtained with a simple dimension-order (with support for torus wrap-around edges) routing algorithm being 30. This is to be compared with the at most 82,308 nodes connected in the case of a \( k \)-ary Tofu \( (k = 19) \) with the same length of a longest path (i.e. 30). As another example, with a maximum path length of 36, Tofu can connect at most 146,004 nodes \( (k = 23) \), which is to be compared with the 1,180,980 nodes of a TCT with the same maximum path length \( (n = 10, k = 3) \). To connect at least the same number of nodes, Tofu needs a maximum path length of 72 (i.e. \( k = 46 \) or \( k = 47 \), inducing 1,168,032 and 1,245,876 nodes, respectively). It is recalled that modern supercomputers include a number of nodes which is in this order: for example, the Fujitsu Fugaku (it is based on Tofu D) embodies 7,630,848 compute nodes and the Sunway TaihuLight 10,649,600.

Therefore, considering the huge number of compute nodes included nowadays in supercomputers – in the order of several million – the torus-connected toroids topology supersedes the conventional approach both in terms of network degree and network diameter, in other words its cost is lower (it is recalled that the network cost is defined as the network degree multiplied by the network diameter). Additional sample values are given in Table 1 to illustrate the trend discussed previously.

Table 1. A comparison of the network order and the maximum path length induced by a simple dimension-order (with support for torus wrap-around edges) routing algorithm in the case of the TCT and Tofu topologies (sample dimensions and arities)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( k )</th>
<th>Network order</th>
<th>Max. length</th>
<th>( n )</th>
<th>( k )</th>
<th>Network order</th>
<th>Max. length</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>750</td>
<td>17</td>
<td>3</td>
<td>324</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2,058</td>
<td>23</td>
<td>4</td>
<td>768</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5,000</td>
<td>24</td>
<td>10</td>
<td>12,000</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>31,250</td>
<td>29</td>
<td>20</td>
<td>96,000</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>49,152</td>
<td>34</td>
<td>30</td>
<td>324,000</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>187,500</td>
<td>34</td>
<td>40</td>
<td>768,000</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>229,376</td>
<td>39</td>
<td>50</td>
<td>1,500,000</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1,093,750</td>
<td>39</td>
<td>60</td>
<td>2,592,000</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1,048,576</td>
<td>46</td>
<td>70</td>
<td>4,116,000</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>6,250,000</td>
<td>46</td>
<td>80</td>
<td>6,144,000</td>
<td>123</td>
<td></td>
</tr>
</tbody>
</table>
5. CONCLUSIONS

As shown by the TOP500 world supercomputing rankings, the torus topology has proven very popular as interconnection network of massively parallel systems. Besides, the number of compute nodes included in such machines has now passed the million-node barrier, with several supercomputers in the top ten even having two or more million. To retain the computing performance high, it is critical to try to reduce the network degree and diameter. To this end, and recognizing both the torus topology and hierarchical interconnection network trends of the supercomputing industry, we have proposed in this paper the torus-connected toroids topology. After first formally establishing the degree and diameter of a meta-node (i.e. toroid) of a TCT network, conducting a validation experiment and giving the TCT formal definition, we have shown that compared to the Tofu interconnect of the Fujitsu K and Fugaku supercomputers – the latter being ranked no. 1 in the world as of May 2022 –, the TCT topology improves on both the degree and diameter issues.

Regarding future works, one objective is to conduct additional experiments to further investigate the performance of the proposal in average. In addition, the diameter of a torus-connected toroids network remains to be formally established. This is however a difficult problem, and an upper bound on the maximum path length as established in this paper is usually sufficient to prove the performance of the topology.

ACKNOWLEDGEMENT

This research was partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under grant no. 19K11887.

REFERENCES


