

Torus interconnect

A torus interconnect is a switch-less network topology for connecting processing nodes in a parallel computer system.

Contents

Introduction

- Geometry illustration
- Torus network topology

Visualization

Performance

- 3D Torus performance simulation
- 6D Torus product performance

Advantages & disadvantages

- Advantages
- Disadvantages

See also

References

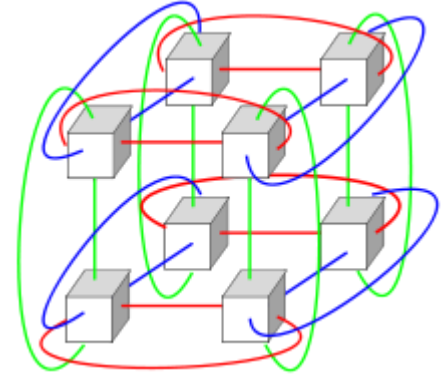


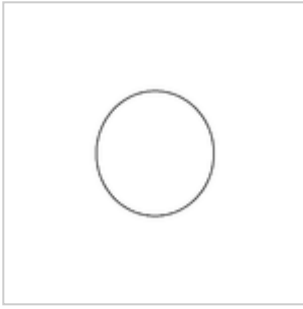
Diagram of a 3-dimensional torus interconnect. It is not limited to 8 nodes but can consist of any number of nodes in a similar rectilinear array.

Introduction

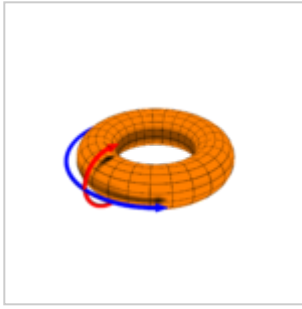
In geometry, a torus is created by revolving a circle about an axis coplanar to the circle. While this is a general definition in geometry, the topological properties of this type of shape describes the network topology in its essence.

Geometry illustration

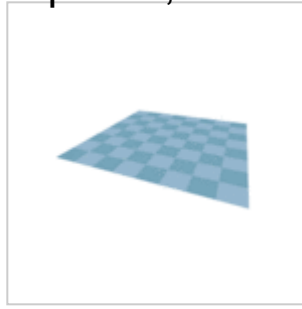
The following images are 1D, and 2D torus. 1D torus is a simple circle, and 2D torus has the shape of doughnut. The animation below illustrates how a 2D torus is generated from a rectangle by connecting its two pairs of opposite edges. Here the concept of torus is used to describe essentially the beginning and ending of a sequence of nodes are connected, like a doughnut. To better illustrate the concept, and understand what the topology means in network interconnect, we give 3 examples of parallel interconnected nodes using torus topology. At one dimension, a torus topology is equivalent to a ring interconnect network, of a shape of a circle. At 2D, it's equivalent to a 2D mesh, but with extra connection at the edge nodes, which is the definition of 2D torus.



1D torus example, a circle.



2D torus example, a donut.



Generating a 2D torus from a 2D rectangle.

Torus network topology

We can generalize the rule from the figures above. Torus interconnect is a switch-less topology that can be seen as a mesh interconnect with nodes arranged in a rectilinear array of $N = 2, 3$, or more dimensions, with processors connected to their nearest neighbors, and corresponding processors on opposite edges of the array connected.[1] In this lattice, each node has $2N$ connections. This topology got the name from the fact that the lattice formed in this way is topologically homogeneous to an N -dimensional torus.

Visualization

The first 3 dimensions of torus topology network are easier to visualize. Below are the description respectively.

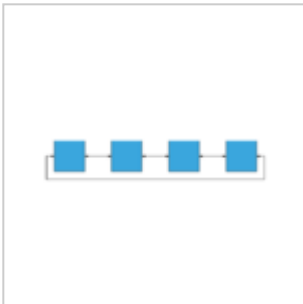


illustration of 1D Torus

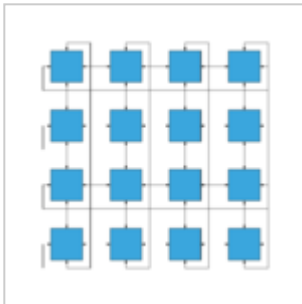


illustration of 2D Torus

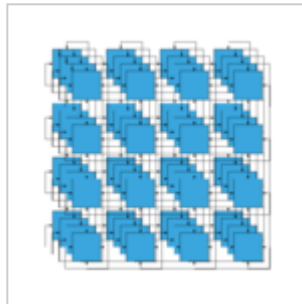


illustration of 3D Torus

- 1D Torus: it is one dimension, n nodes are connected in closed loop with each node connected to its 2 nearest neighbors communication can take place in 2 directions, $+x$ and $-x$. 1D torus is same as ring interconnection.
- 2D Torus: it is two dimension with degree of 4, the nodes are imagined laid out in a two-dimensional rectangular lattice of n rows and n columns, with each node connected to its 4 nearest neighbors, and corresponding nodes on opposite edges connected. The connection of opposite edges can be visualized by rolling the rectangular array into a "tube" to connect two opposite edges and then bending the "tube" into a torus to connect the other two. communication can take place in 4 directions, $+x$, $-x$, $+y$, and $-y$. The total nodes of 2D Torus is n^2
- 3D Torus: it is three dimension, the nodes are imagined in a three-dimensional lattice in the shape of a rectangular prism, with each node connected with its 6 neighbors, with corresponding nodes on opposing faces of the array connected. Each edge consists of n nodes. communication can take place in 6 directions, $+x$, $-x$, $+y$, $-y$, $+z$, $-z$. Each edge of 3D Torus consist of n nodes. The total nodes of 3D Torus is n^3
- N D Torus: it can have N dimension, each node of N dimension torus has $2N$ neighbors, communication can take place in $2N$ directions. Each edge is consist of n nodes. Total nodes of this torus is n^N . The main motivation of having higher dimension of torus is t achieve higher bandwidth, lower latency, and higher scalability.

Higher-dimensional arrays are difficult to visualize but we can see from above rule that each higher dimension adds another pair of nearest neighbor connections to each node.

Performance

A number of supercomputers on the [TOP500](#) list use three-dimensional torus networks, e.g. IBM's [Blue Gene/L](#) and [Blue Gene/P](#), and the [Cray XT3](#).^[1] IBM's [Blue Gene/Q](#) uses a five-dimensional torus network. Fujitsu's [K computer](#) and the [PRIMEHPC FX10](#) use a proprietary three-dimensional torus 3D mesh interconnect called Tofu.^[2]

3D Torus performance simulation

Sandeep Palur and Dr. Ioan Raicu from Illinois Institute of Technology conducted experiments to simulate 3D torus performance. Their experiments ran on a computer with 250GB RAM, 48 cores and x86_64 architecture. The simulator they used was ROSS ([Rensselaer's Optimistic Simulation System](#)). They mainly focused on three aspects 1. Varying network size 2. Varying number of servers 3. Varying message size. They concluded that throughput decreases with the increase of servers and network size. Otherwise, throughput increases with the increase of message size.^[3]

6D Torus product performance

[Fujitsu Limited](#) developed a 6D torus computer model. In their model, 6D torus can achieve 100 GB/s off-chip bandwidth, 12 times higher scalability than 3D torus, and high fault tolerance.^[4]

Advantages & disadvantages

Advantages

- Higher speed, lower latency

Because of the connection of opposite edges, data have more options to travel from one node to another which greatly increased speed.

- Better fairness

In a 4×4 mesh interconnect, the longest distance between nodes is from upper left corner to lower right corner. Each datum takes 6 hops to travel the longest path. But in a 4×4 Torus interconnect, upper left corner can travel to lower right corner with only 2 hops

- Lower energy consumption

Since data tend to travel fewer hops, the energy consumption tends to be lower.

Disadvantages

- Complexity of wiring

Extra wires can make the routing process in the physical design phase more difficult. If we want to lay out more wires on chip, it is likely that we need to increase the number of metal layers or decrease density on chip, which is more expensive. Otherwise, the wires that connect opposite edges can be much longer than other wires. This inequality of link lengths can cause problems because of [RC delay](#).

- [Cost](#)

While long wrap-around links may be the easiest way to visualize the connection topology, in practice, restrictions on cable lengths often make long wrap-around links impractical. Instead, directly connected nodes—including nodes that the above visualization places on opposite edges of a grid, connected by a long wrap-around link—are physically placed nearly adjacent to each other in a folded torus network.^{[5][6]} Every link in the folded torus network is very short—almost as short as the nearest-neighbor links in a simple grid interconnect—and therefore low-latency.^[7]

See also

- [Computer cluster](#)
- [Heartbeat private network](#)
- [Switched fabric](#)

References

1. N. R. Agida et al. 2005 *Blue Gene/L Torus Interconnection Network*, IBM Journal of Research and Development, Vol 45, No 2/3 March–May 2005 page 265 "Archived copy" (https://web.archive.org/web/20110815102821/http://www.cc.gatech.edu/classes/AY2008/cs8803hpc_spring/papers/bgLtorusnetwork.pdf) (PDF). Archived from the original (http://www.cc.gatech.edu/classes/AY2008/cs8803hpc_spring/papers/bgLtorusnetwork.pdf) (PDF) on 2011-08-15. Retrieved 2012-02-09.
2. *Fujitsu Unveils Post-K Supercomputer* HPC Wire Nov 7 2011 (http://www.hpcwire.com/hpcwire/2011-11-07/fujitsu_unveils_post-k_supercomputer.html)
3. Sandeep, Palur; Raicu, Dr. Ioan. "Understanding Torus Network Performance through Simulations" (http://datasys.cs.iit.edu/reports/2014_GCASR14_paper-torus.pdf) (PDF). Retrieved 28 November 2016.
4. Inoue, Tomohiro. "The 6D Mesh/Torus Interconnect of K Computer" (<http://www.fujitsu.com/downloads/TC/sc10/interconnect-of-k-computer.pdf>) (PDF). Fujitsu. Retrieved 28 November 2016.
5. "Small-World Torus Topology" (http://jhnet.co.uk/projects/figures/small_world_torus).
6. Pavel Tvrdik. "Topics in parallel computing: Embeddings and simulations of INs: Optimal embedding of tori into meshes" (<http://pages.cs.wisc.edu/~tvrdik/6/html/Section6.html#AAAAAEmbeddings%20into%20meshes%20and%20tori>).
7. "The 3D Torus architecture and the Eurotech approach" (http://www.eurotech.com/DLA/Products_Eurotech/Aurora/HP_C_3D_Torus_Short_Paper.pdf).

Retrieved from "https://en.wikipedia.org/w/index.php?title=Torus_interconnect&oldid=866504060"

This page was last edited on 30 October 2018, at 18:57 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.