HILBERT GRAPH: AN EXPANDABLE INTERCONNECTION FOR CLUSTERS

Fernando Rodríguez Salazar and John R. Barker
Department of Electronics and Electrical Engineering
University of Glasgow, Glasgow G12 8LT, U.K.
{F.Rodriguez, J.Barker}@elec.gla.ac.uk

The use of commodity hardware has played an important role in the construction of high performance computers in recent years. In particular this model of computation has become very popular in research groups and organisations due to its favourable price/performance ratio [1]. Although cluster computing has been very successful for applications consisting of a large number of serial tasks characterised by a low communication cost, the approach has been less successful in the scenario where a small number of tasks with high communication costs need to be executed; such as in multigrid methods and other parallel iterative solvers [2].

Recently new interconnection fabrics such as Infiniband have been introduced, which offer much improved communications [3]. However, to fully exploit its potential, the topology of the interconnection network (IN) is critical, since it significantly determines the cost, performance and usability of the resulting system. The machine should be simple to programme, which implies that it should efficiently support the communication pattern generated by the (possibly legacy) application being executed, which may be the result of automatic parallelisation techniques.

Meshes and tori have been widely used as communication structures for multicomputer systems and can be used to connect groups of workstations together. However these structures exhibit a large diameter and average length, which limits the communication patterns supported by them. To remedy this situation, it has been proposed to extend these structures with “express channels”, allowing messages to bypass a number of nodes.

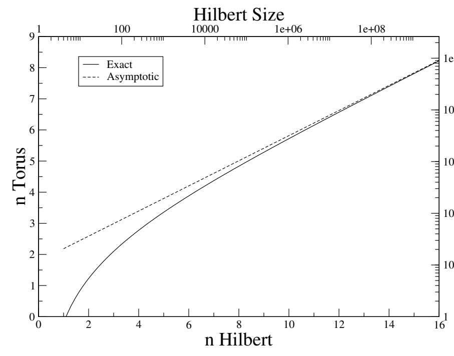
The Hilbert graph introduced in this work is formed by the superposition of a Hilbert curve with an extended mesh. Nodes are placed along the middle of a segment in the Hilbert curve, as shown in figure 1(b,c), and the extended mesh is formed by joining nodes with the same horizontal or vertical position. The Hilbert graph has a fixed degree of four, and exhibits a much better support for random communication patterns and an efficient two dimensional layout, while retaining the same cutwidth complexity of a two dimensional torus. Furthermore, it retains the incremental expandability found in the mesh and torus as shown in figures 2 and 1(b,c), while supporting the increased traffic expected from such expansion much more efficiently (as shown in figure 1(a)).

The present work shows that a torus of size M exhibits the same congestion of a Hilbert graph of size $O(M^{2.477})$. Alternatively the Hilbert graph will tolerate correspondingly larger levels of traffic before saturation. Furthermore, while broadcast in the two dimensional torus takes at least $N^{1/2}$ steps, it is shown that the same operation can be accommodated in $O(N^{1/4})$ steps in a Hilbert graph.

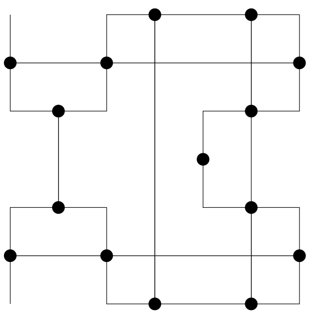
REFERENCES

[1] D.J. Becker et al. Beowulf: A parallel workstation for scientific computation. In *Proc. International Conference on Parallel Processing*, 1995.
[2] R. Buyya, H. Jin, and T. Cortes. Cluster computing. *Future Generation Computer Systems*, (18):v–viii, 2002.
[3] Infiniband Trade Association. Infiniband Technology Overview.

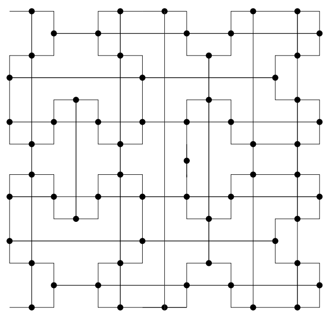
A full journal publication of this work will be published in the Journal of Computational Electronics.



(a)

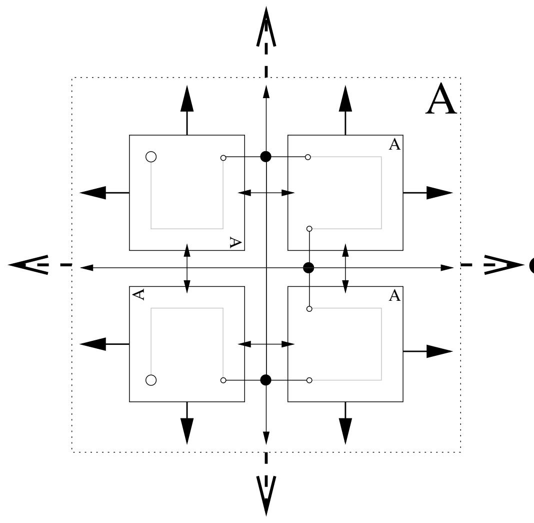


(b) H2

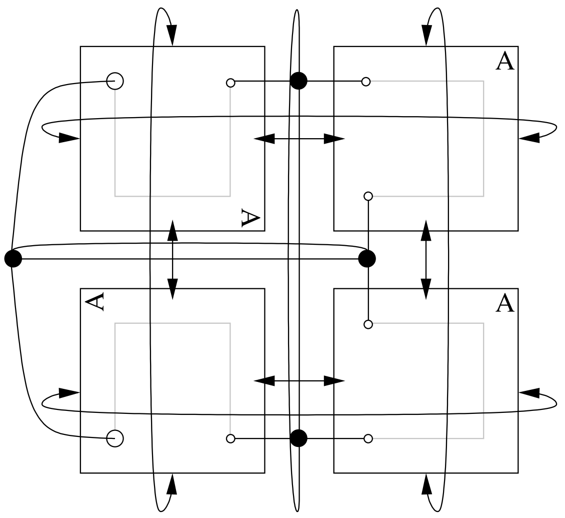


(c) H3

FIGURE 1. In (a) the size of Torus and Hilbert graphs with equal congestion is shown. The exact value as well as the asymptotic behaviour are shown. The plot suggests that the Hilbert graph allows for much larger systems and hence is much more expandable than the Torus interconnection network. In (b-c) the recursive construction of Hilbert Graphs is shown.



(a) Open



(b) Closed

FIGURE 2. Recursive construction of a Hilbert graph. The H_{n+1} Hilbert graph consists of four H_n graphs rotated as shown in (a). The letter 'A' is used as an orientation marker. Extensions are connected to adjacent graphs internally, and are grouped together externally to form the new graph. The top level graph is closed by looping back the extensions as shown in (b).

A full journal publication of this work will be published in the Journal of Computational Electronics.