Department of Computer Science

Advanced Computer Architecture Group

"Ball Computer"
Or
How I Learned to Stop Worrying and Love Computing with 3D Wireless Grids

Amir Mansoor Kamali, Christopher Crispin-Bailey, Jim Austin
{amir, chrisb}@cs.york.ac.uk, jim.austin@york.ac.uk

THE UNIVERSITY *of York*

Introduction

We are investigating if wireless devices can replace wireline networks to serve as an interconnect network for a massively parallel computer.

The radio devices are getting faster, cheaper, smaller and consume less energy compared to last decade. Some of them occupy a fraction of a square centimetre on silicon. They are already used in other applications including wireless NoC and chip to chip wireless communications.

In this project we want to find a more precise picture of the benefits and restrictions of a 3D wireless massively parallel machine. A novel concept machine called the ***Ball Computer*** is presented which is basically a multi-channel 3D wireless grid. It serves as an interconnect network for a parallel machine. The machine is implemented only in a simulation environment. A novel method to share the available radio bandwidth is introduced. To test the performance of the platform a couple of task models are built which are basically artificial traffic load generators. Moreover, a set of simulation and visualisation tools is implemented.

Design Questions

The gap between wireless and wireline technologies is much narrower than last decades. However, wireless still has some big problems. The main challenges are:

Bandwidth

Data communication rates have sharply increased during past decade (Figure 1.a). Most researches are currently on implementing 60GHz radios on silicon. Some has tried higher bands including 120GHz. Newly emerged 1THz on-chip radios are very promising. This can radically increase the available data rates.

Energy Consumption

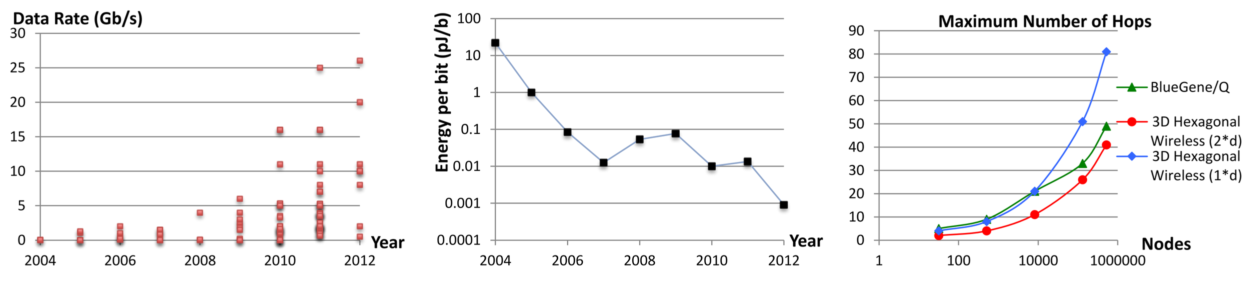
Wirelines need as low energy per bit as 1 pJ/bit. The energy consumption for a wireless transmission over 10mm is at least tens of pJ/bit. Figure 1.b is just an estimation, but it shows the energy per bit is falling for radio communication year by year.

Power Delivery[1]

An effective solution is yet to be found on this challenging field. Some possibilities are:

a) Capacitive coupling: Occupies very small area but it is suitable only for very short ranges in nanometre scale.
b) Inductive coupling: Is capable of transfering power to longer distances but both methods are not good enough from the performance point of view.
c) Radio waves.
d) Optical Connections.

A combination of RF for longer distances and optical links for shorter ones seems to be a good choice.



(a) (b) (c)

Figure 1[2]**:** ***(a)*** *Data rates of on-chip radio devices over the last decade.* ***(b)*** *An estimation of energy for a 1Gbps link over 10mm.* ***(c)*** *Comparison between the number of hops in a Blue Gene/Q and a Ball Computer.*

Area

Many on-chip radio modules are available with just a few square millimetre of chip area. Good news is higher frequencies need smaller on-chip antennas.

Number of Hops

In a parallel architectures like Blue Gene/Q a node can contact any other node in small number of hops due to its high dimensionality. For example Blue Gene/Q uses an elaborate system of wires to implement a 5D network. In a hexagonal wireless 3D network not all neighbours are orthogonal; therefore, the number of hops are larger than Blue Gene/Q for very large networks. Increasing the radio range (Blue graph in figure 1.c) means less number of hops but at the same time lower bandwidth for each link.

Simplicity and Scalability

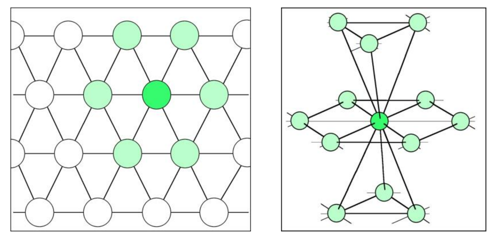
A wireless network saves wiring costs. The complexity of the system is also reduced. A wireless network is highly scalable as nodes can be easily added just by tuning to proper frequency channels.

1: The authors of this poster thank Mr. Richard Hind for providing data for this part of the poster.

2: Graphs (a) and (b) are derived from a survey by the authors published in "A. M. Kamali, C. Crispin-Bailey, J. Austin; "On advantages and Limitations of 3D Wireless Grids as Parallel Platforms"; International Conference on Selected Topics in Mobile and Wireless Networking, MoWNeT'2013; August 19th-21st 2013; Montreal, QC, Canada; pp 48-55."

Ball Computer Architecture

A 3D wireless network of processing elements are proposed as an interconnect network for a massively parallel computer. The platform is called the ***Ball Computer***. The network has hexagonal FCC topology. Each node has up to 12 neighbours in a 3D grid (6 in 2D). Figure 2 shows a node and its neighbours in both 2D and 3D networks.



(a) (b)

Figure 2: *Nodes in* ***(a)*** *a 2D; and,* ***(b)*** *a 3D grids. In real world nodes are in physical contact with all of their immediate neighbours.*

Each node is made of a small electronic board inside a small plastic ball (1 or 2 cm in diameter). Each node has a processor and 8 radio transceivers. A network partitioning algorithm guarantees no packet collision by tuning radio modules to channels defined by the algorithm. The algorithm dictates that for a 3D FCC grid of any size each node needs 8 radio links. Moreover, increasing the size of the network does not affect the total number of channels.

Network Partitioning

Figure 3 shows how a node in a ball computer interacts with its neighbours. The central node has eight radio links each of which is shared with 3 neighbours.

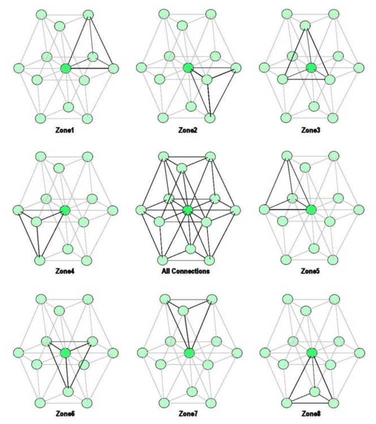


Figure 3: *An illustration of a node with its 12 neighbours in a 3D environment.*

Each pyramid-shape subset of the network is called a zone. This is the outcome of a novel greedy network partitioning algorithm. The Hidden Node Problem is solved and no two packets collide when no two close zones operate on the same channel.

The algorithm satisfies the following criteria:

a) Eliminating packet collision by solving Hidden Node Problem;
b) Maximising the connectivity between nodes by providing redundant links; and
c) Keeping the number of channels as low as possible.

Network partitioning has already been used in mobile phone networks and wireless internet access networks with different criteria. Our algorithm has two sub-algorithms:

Zoning

The zoning algorithm partitions the whole network into overlapping subsets (zones) in a way that there is no hidden node in any zones (as shown in figure 4).

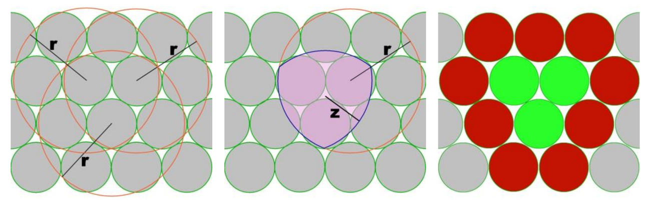


Figure 4: *A zone forms out of overlapping radio ranges. Nodes inside a zone communicate on a channel which is forbidden for Nodes outside the zone (Green nodes in contrast with red nodes).*

Inside a zone all nodes can hear from others and no packet collision can happen.

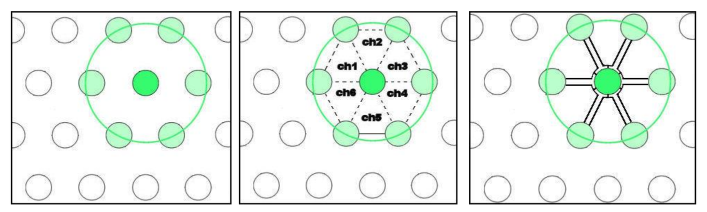


Figure 5: *A central node and its zones. Compare the wireless connections with their wireline equivalent.*

In figure 5 a 2D multi-channel wireless grid is compared with its wireline equivalent.

Channel Assignment

Channel assignment can be compared with map colouring problem. In channel assignment a zone cannot share its colour (channel) with its neighbours and also it cannot share its colour with the neighbours of its neighbours.

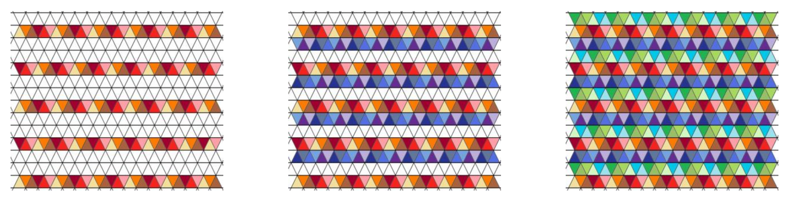


Figure 6: *Choosing channels for zones. Each colour represents a channel.*

The number of colours does not increase when the size of the networks increases. For a 3D network of any size the total number of channels is 89.

Task Modelling

Task models are artificial traffic load generators which mimic a certain task or class of tasks without dealing with task-specific details in order to get rid of unimportant details of real-world tasks. Two main task models were studied in this project:

a) Simple Parallel task model: A node sends a number of tasks to its neighbours and waits for results. The completion of the tasks only depends on its neighbours.
b) FFT task model: A task model inspired (but not restricted to) FFT algorithm which makes a vast and tightly connected network of tasks. In a recursive process, a workload is divided into pieces and shared with other nodes. The finish time of a task in a node depends on many nodes even beyond its local neighbourhood.

Simulation Results

Packet Collision

The network partitioning algorithm produced perfect results and led to complete elimination of packet collision. The number of packet collisions was counted in a series of tests. In a 2D simulated 800-node network the number of distinct channels per node was 1, 2 and 6. The number of collisions was 1731342, 665271 and 0 respectively.

Task Models

The performance of a simple parallel task model and an FFT task model are shown in figure 7. The performance is measured over different data sizes and different intervals between data packets. It can be seen that for larger data sizes the ball computer performs much better.

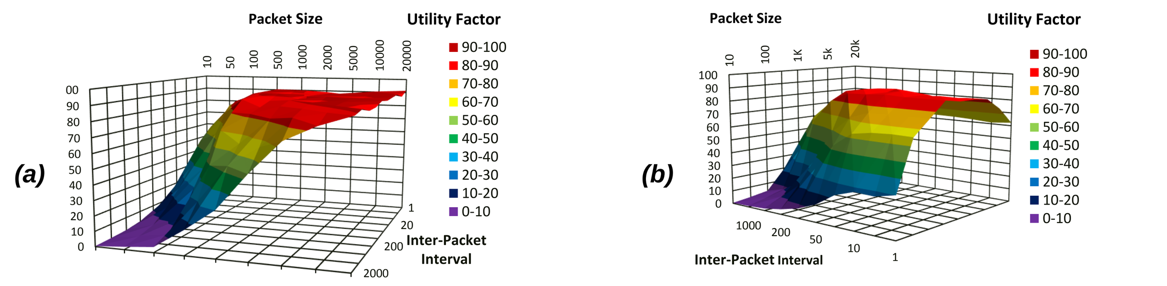


Figure 7: *The performance of an FFT task model in a network of 800 nodes.*

Multi-tasking Nodes

Figure 8 shows how multi-tasking nodes increase the performance of a divide-and conquer task like FFT. Two different data sizes are used to test FFT task model.

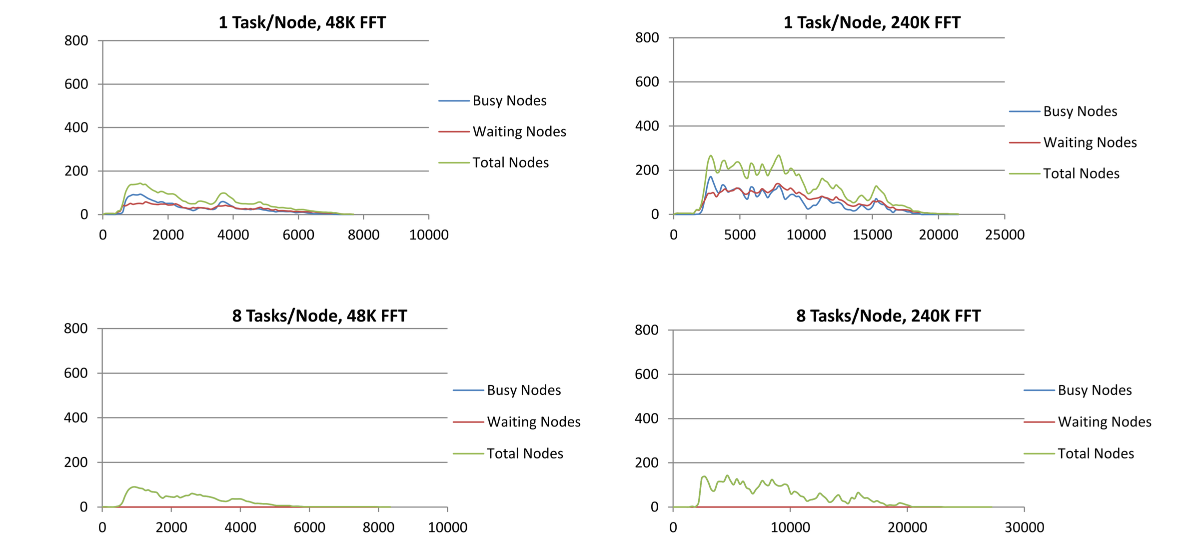


Figure 8: *The effect of multi-tasking on the performance of an FFT task model.*

Figure 9 shows how applying multiple workloads at the same time improves the performance of a network.

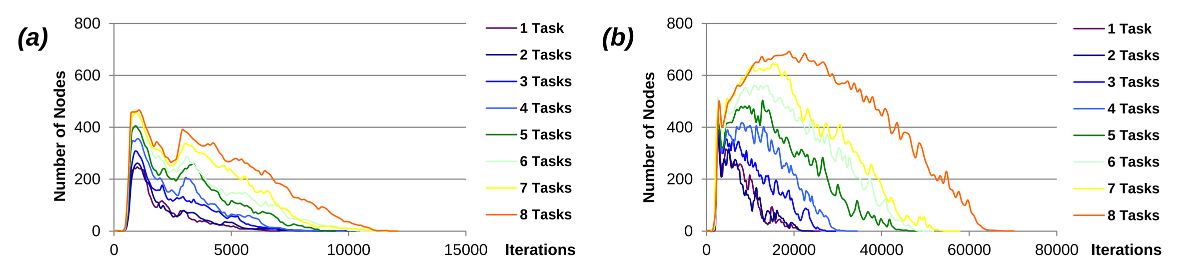


Figure 9: *Number of nodes engaged in a set of multiple FFTs with* ***(a)*** *48KB,* ***(b)*** *240KB of data.*

Number of Branches per Node

The number of neighbours recruited by an FFT task model defines the shape of the dependency tree generated by the task. The optimum number of branches per node is found through a set of experiments (Figure 10). It shows that the optimum number of branches for a node is 6 and 5 for 3D and 2D networks respectively.

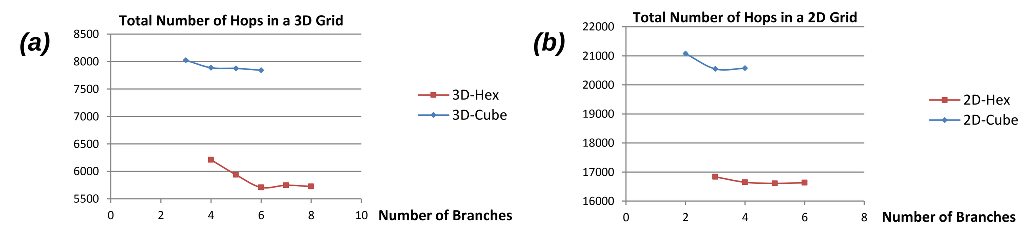


Figure 10: *The Optimum number of branches per node for FFT in* ***(a)*** *3D and* ***(b)*** *2D.*

Network Attributes

The role of network attributes on the behaviour of the network is studied in series of experiments. The data rate of links, computational ability of nodes, data size and the network size are subject to change. Figure 11 shows how the Utility Factor changes in a network of 1000 nodes with FFT task model. Results show higher performances are achieved with larger data sizes and larger "instructions per second / transfer rate" ratio.

Figure 11: *The utility factor of a network of 1000 nodes running FFT task models.*

The same tests are repeated with 2000 nodes. Figure 12 shows a drop in performance for all the attribute values.

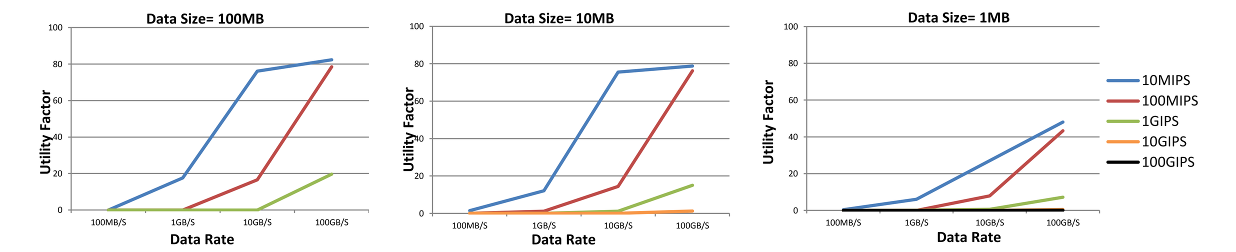


Figure 12: *The utility factor of a network of 2000 nodes running FFT task models.*

This suggests that the size to which this version of FFT task model expands and yields the best performance is probably something between 1000 and 2000 nodes (Also confirmed by figure 13). This is not equal to the best size for a Ball Computer.

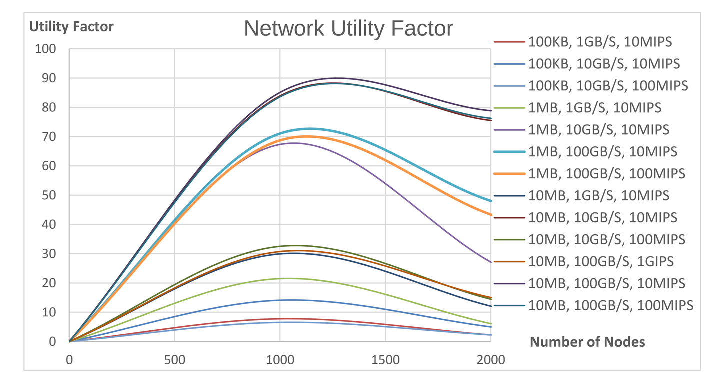


Figure 13: *Estimated optimum task size for an FFT task model is between 1000 and 2000 nodes.*

The same series of experiments are repeated for simple parallel task model and shows that the performance of the network does not change with its size. Figure 14 shows the results for a simple parallel task model applied to a network of size 1000 nodes. The results are almost identical to those of a network of size 2000 nodes. In this task model the dependencies are very local and are not affected by network size.

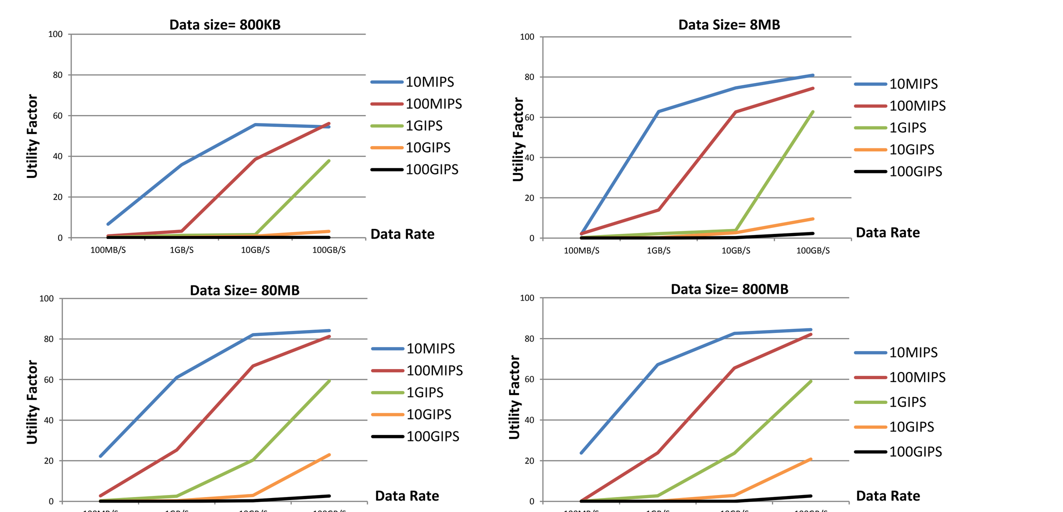


Figure 14: *Utility factor in a network of 1000 nodes running Simple Parallel task models.*

Visualisation Results

The performance of the simulated network can be studied by measuring a number of traffic and computation factors. As an example, figure 15 shows that there are a few nodes which are suffering from congestion of their I/O queues. Architects and software designers can use these data to fine-tune the network parameters or develop more efficient programs for this topology.

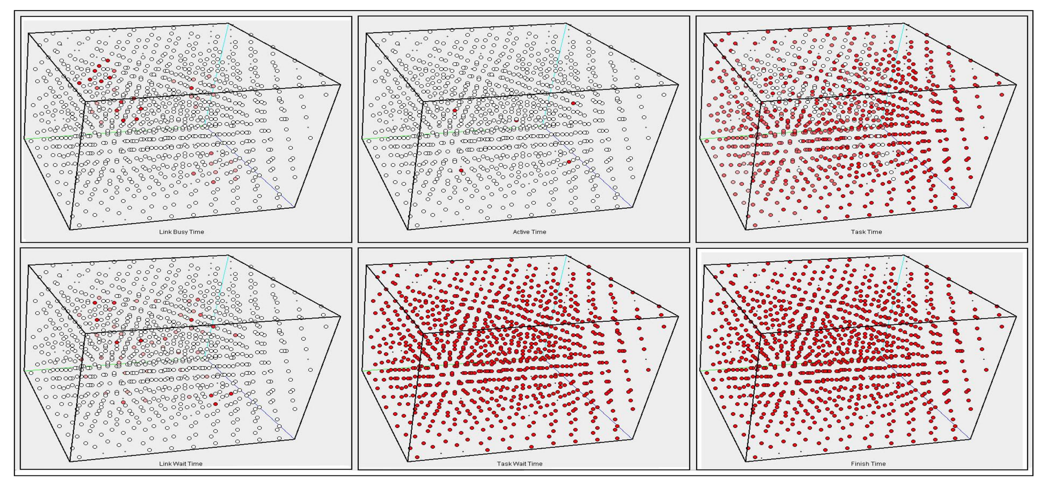


Figure 15: *Studying the behaviour of the network using visualisation tool which shows Bottlenecks.*