Designing Large Hierarchical Multiprocessor Systems under Processor, Interconnection, and Packaging Advancements [1]

Debashis Basak and Dhabaleswar K. Panda

Department of Computer and Information Science
Ohio State University
Columbus, OH 43210-1277
Tel: (614)-292-5199, Fax: (614)-292-2911
Email: basak,panda@cis.ohio-state.edu

Abstract

In this paper we present a general framework for architectural design of large hierarchical multiprocessor systems under rapidly changing packaging, processor, and interconnection technologies. Processor boards with larger area (A) and greater pinouts are becoming feasible. Board interconnection technology has advanced from only peripheral connections $O(\sqrt{A})$ to elastomeric surface connections $O(A)$. As processor and interconnection technology is growing, there is a varying demand on the interconnection network of the system. The proposed framework is capable of taking into account all these changing technologies. Each technology is captured through one or more parameter(s) which reflects its level of advancement. Depending on a given set of values of these parameters, the framework guides us to choose the most optimum topology. The framework is illustrated by considering the design problem of the currently popular k-ary n-cube cluster-c scalable architecture. These architectures combine the scalability of k-ary n-cube wormhole-routed networks with the cost-effectiveness of processor cluster designs. Each cluster consists of c processors interconnected through a bus/MIN/direct network. Our results indicate that under surface pinout technology an increase in cluster size(c) is associated with a growth in bisection bandwidth, whereas in periphery pinout it leads to a fall in bisection bandwidth. For systems supporting 16-bit links, cluster sizes of 2 to 3 processors with 3D/4D k-ary n-cube interconnections are optimal. Similarly systems with 32-bit links can support larger cluster sizes(c=7,8) and 3D/4D interconnections.

Keywords: packaging constraints, hierarchical systems, clustered systems, pinout constraint, interconnection network, k-ary n-cube. systems.

[1] This research is supported by the National Science Foundation Grant # MIP-9309627.

Contents

1 Introduction

Designing large parallel systems to provide high computing bandwidth is an issue of important consequence. Rapid developments in the field of processor, memory, network, and, packaging technologies make the task of efficient design of large multiprocessor systems a difficult task. Proposed design guidelines need to take into account changes in these technologies to provide good performance. One basic guideline which holds good universally is that large systems need to have a hierarchical packaging. This is naturally imposed by the packaging constraints like board size, pinouts, limited cabinet size etc. In this light it makes sense to design large systems hierarchically so that there is close relationship between the design and the actual packaging.

A variety of two-level hierarchical configurations have been proposed by researchers in the past to build scalable systems. Though a larger number of levels might be more general, it is commonly believed that most systems in the near future will fit in two hierarchies. Moreover the techniques which work for two-level hierarchies can be easily extended to design more levels. Examples of previous work in this area include two-level systems based on hypercube and other network topologies[5, 15, 18], MINs and n-hop networks [18], two-level systems with k-ary n-cube topologies[2], and combination of bus and mesh/hypercube networks[10]. However, most of these analyses either did not consider packaging and interconnecting constraints, or provided guidelines based only on fixed board sizes with fixed pinouts, not considering changes in board sizes and alternate pinout technologies. The changes in processor and interconnection technology and its relation to packaging technology in designing systems was also not studied. Some of the works suggested guidelines based only on uniform traffic patterns. However, localized traffic patterns are more common in distributed shared memory(DSM) systems with caches [8] or cache-only memories [9, 13]. Localized traffic patterns put lesser load on the higher hierarchies, allowing system designs with low-bandwidth networks to be considered which may have been rejected outright under uniform traffic assumption.

In this paper we present a general framework to provide design guidelines for building large hierarchical multiprocessor systems. We consider a packaging technology with two levels: processors in the parallel computer are organized into a number of boards(clusters) and the boards are then connected through inter-board wires. Though the precise characteristics of existing and emerging packaging technologies are hard to model, Raghunath [17] has shown that a common characteristic of most packaging technologies is a progressive increase in costs and decrease in capacities as we go up the packaging hierarchy. For instance, it is clear that as we proceed up the hierarchy, wires usually get longer with larger inter-wire spacings. Metal wires on VLSI chips may be about 1 micron apart, whereas wires connecting different boards may be as long as 1 meter and spacing between wires on a cable connector may be as much as 1 mm. This results in a higher cost per wire at upper levels of the hierarchy and makes the design of these levels more critical.

We propose a new framework, as shown in Fig. 1, for designing and developing clustered architectures. The basic goal is to design an efficient system with N processors with

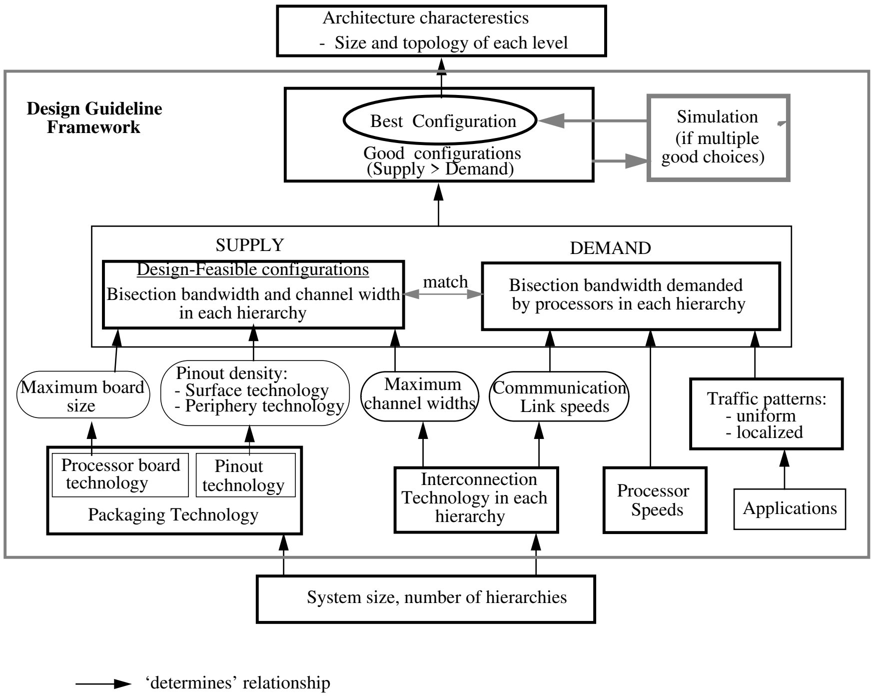


Figure 1: Underlying framework for designing and developing hierarchical architectures under technological constraints.

two-hierarchies. Theoretically a large number of alternate configurations are possible to build such a system. The configurations which conform to the packaging restrictions, like maximum board size, the pinout technology being used - peripheral or surface, and the pinout densities, are defined as *design-feasible* configurations. It is clear that only the design-feasible configurations can translate into real machines. Each of the design-feasible configurations offers some performance depending on the size of the bisection in each hierarchy. Referring to the set of design-feasible configurations and their performance characteristics as what the packaging technology offers, we call this the *supply* side. On the other hand, processors make demands on the bisection bandwidth in each hierarchy. This demand depends on the speed of processors. The speed of the communication links determine the size of the bisection required to handle the demanded traffic. The subset of design-feasible configurations which offer performance greater than what is demanded are defined as *good* configurations. The final step is to select a *best* configuration from amongst the *good* configurations. This is done on the basis of relative performance and cost of these systems. Simulation modeling may be employed to decide among very close candidates.

The paper is organized as follows. In section 2 we discuss the two-level k-ary n-cube cluster-c architecture [2] used in this paper to illustrate the working of the design framework. In section 3 we discuss the trends in growth of processor board sizes, alternate pinout technologies, channel width technology, and derive the design-feasible configurations and their characterestics. Processor speeds, communication link speeds and the performance demand are discussed in sec. 4. In Section 5 we show how to derive the good configurations and discuss important considerations in choosing the best configuration. Finally, concluding remarks and future work are presented.

2 Architectural model and alternatives

In this section we discuss the class of k-ary n-cube cluster-c systems [2, 7]. Such architectures are currently the trend in building large scalable systems. Examples are Intel Paragon [12], Stanford DASH [6] and Cray T3D [4]. A k-ary n-cube cluster-c system consists of two levels. The lower level consists of k^n clusters of processors interconnected by a higher level direct k-ary n-cube network(also referred to as inter-cluster network or *internet*). Each cluster consists of c processors interconnected through buses/MINs/direct networks. The significance of such systems is twofold: processor clusters allow easy exploitation of communication locality and the k-ary n-cube internet provides easy scalability to the system. Figure 2 shows the overall configuration, while Fig. 3 shows an example when clusters are specifically bus-based.

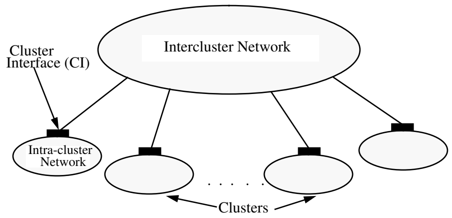


Figure 2: A two-level hierarchical system.

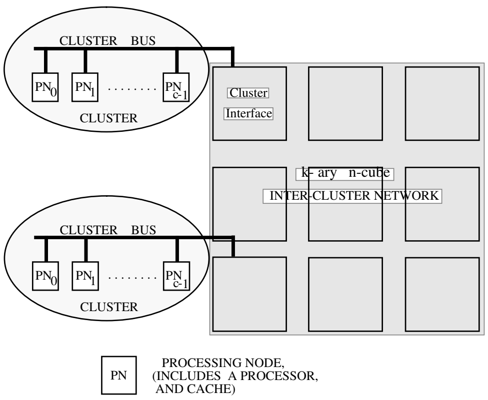


Figure 3: An example k-ary n-cube bus-c architecture made up of bus-based clusters. A cluster bus connects c processing nodes and a cluster interface. The k^n cluster interfaces are interconnected through a k-ary n-cube network.

Though in this paper we work with k-ary n-cube internets, our design framework is more general and can work for other internet topologies like DeBruijn graphs and n-hop networks [3, 18]. Without loss of generality, in this paper, we emphasize on k-ary n-cube because it is a currently popular trend which we believe will continue in the near future also. Thus the insights provided by our analysis in this paper will prove useful in designing and packaging future machines which are based on the k-ary n-cube cluster-c model.

Now let us consider designing a system with N processors. There are many different alternatives to form a 2-level clustered configuration. There are basically two degrees of freedom: A) Size of levels and B) Topology choice in each level.

A) Size of levels: Consider N to be 1024. One can design this system with 64 clusters of 16 processors each, 16 clusters of 64 processors each, and so on. Note that for a given system size, fixing the size of one level automatically determines the size of the other level.

A processor cluster may be accommodated on a single board(referred to as *physical clustering*) or it may span multiple boards (referred to as *logical clustering*). In the rest of the paper we use *clustering* to imply physical clustering.

B) Topology choices: Having fixed the size of each level we still have the freedom to vary the topology in each level. For example, in a system with 64 clusters with 16 processors each, one can have a 4-ary 3-cube internet with bus-based clusters, 8-ary 2-cube internet with MIN-based clusters, and so on. Each alternative leads to a different clustered architecture with different properties in terms of message latency, message throughput, synchronization costs etc. Our objective is to select the configuration that offers best system performance.

3 Deriving *design-feasible* configurations

For a design alternative of a given system size, the specific architectural characteristics like bisection bandwidth, channel widths etc. can only be precisely stated with respect to the packaging technology in which it is molded. We first discuss current trends in board and pinout technologies. Then we present a general scheme in which architectural characteristics are parameterized with respect to the board and pinout technology. Finally we discuss the general trends and constraints on the interconnection channel widths, which impose restrictions on the design space.

3.1 Processor board technology

Processor boards cannot be arbitrarily large in size. Their size is restricted by electrical, mechanical, and board fabrication constraints. In our system we assume each board to contain c processors chips, intra-cluster interconnection network, local memories and the cluster interface(consisting of part of the internet router logic and cache-coherence logic if it is a DSM system [8]). The maximum size of processor board restricts the number of processors(c) that can be put on a board and imposes a natural limit on the pinout.

It has been shown in [8] that the area of processor and memory chips dominate the space requirements. Let us define the area required by a processor chip, its local memory, and a per-processor fraction of the network and other logic, to be a. In our analysis, we assume this area to be organized as a square, with a perimeter of $4\sqrt{a}$. Of course a depends on the amount of per-processor memory provided in the system. In the ensuing discussion when we talk about a *processor area a* we include in it a reasonable amount of per-processor memory. We use a as an unit of board area and measure board sizes in terms of a.

In recent years, board sizes have grown in physical dimensions. Progressive improvements in the processor-memory integration has also resulted in a becoming smaller leading to more processors on the same board. We now have contemporary prototypes like the MIT J-machine [14] which has 64 processors on a board. The technology at a given time determines the largest board size $(m.a)$, which can fit at most m processors. The maximum board size determines the largest cluster size in our system design. Without loss of generality, in this paper we assume m to be 64.

Given a board size $(j.a)$, with $(j \leq m)$, we can accommodate upto j processors on it. This is called the *capacity* of the board. One can *under-populate* a board by placing smaller than the capacity number of processors on it. A direct effect of under-populating is that it leads to wastage in precious board space, as a consequence larger number of boards are required, leading to greater system volume. The advantage of under-populating is that the remaining area of the board may be used to support larger number of inter-board links which may improve system performance. However, assuming that these links are expensive [18], this performance gain might not be cost-worthy. We therefore choose to avoid under-populating and develop design guidelines assuming a board area is packed to its capacity. Thus we assume that for building a system with N processors we are given exactly $(N.a)$ total board space. In other words the sum total of areas of all boards in the system is $(N.a)$.

3.2 Two pinout technologies

The pin-count P out of a board has a direct influence on the data rate that can flow in/out of a given processor cluster. It has been shown in [18] that for any given inter-board network topology(3D mesh, hypercube), P also directly determines the bisection bandwidth of the inter-board network, and hence the peak performance of the system.

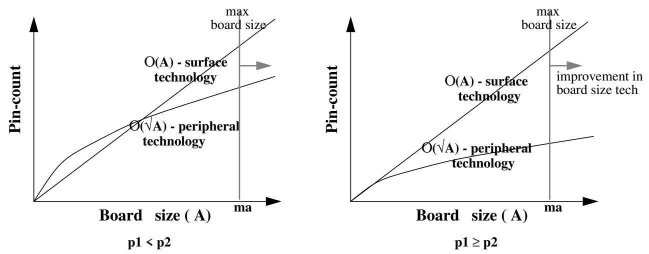
Currently two different types of technologies are being employed by the computer industry:

A) Surface pinouts: In this technology the surface of the board is utilized for external connections using elastomeric connectors [14]. Under this technology the pin-count from a board of area A is $O(A)$. Let us define parameter $p1$ to be the pin-count that can be supported on a given board area a using surface pinout technology. This implies that a square board area of $(j.a)$ would have a peripheral pin-count of $(j.p1)$.

B) Peripheral pinouts: This is the more traditional technology. Only the periphery of the board is used for external connections. Thus if the area of the board is A, the peripheral pinout is limited by $O(\sqrt{A})$. Let us define parameter $p2$ to be the pin-count that can be supported from a perimeter of length $4\sqrt{a}$(this corresponds to the perimeter of a square board area a) using peripheral pinout technology. This implies that a square board area of $(j.a)$ would have a peripheral pinout of $(p2\sqrt{j})$.

It is easy to see that the pincount in the surface technology could be potentially an order of magnitude more than the $O(\sqrt{A})$ periphery technology. This technology has been used in the J-machine prototype and is partly the reason why 64 processors could be packed on a single board. In Fig. 4 we show the growths of pincount with board area under the two different technologies. We also compare the relative growth of pincounts in the two technologies as board sizes are increased, under different relationship between $p1$ and $p2$. The vertical clipping line in each graph is to remind the reader that the board size cannot grow continuously and is limited by some maximum size $(m.a)$.

Board size A (in units of *a*)	Pincount in O(A) surface tech	Pincount in O(√A) periphery tech
1	1p1	1p2
2	2p1	√2p2
4	4p1	2p2
8	8p1	2√2p2
16	16p1	4p2
32	32p1	4√2p2
64	64p1	8p2



Growth of pincount with board area under the two technologies are shown for two different relationships between p1 and p2. The parameters p1 and p2 reflect the pin-count that can be supported from a unit board area *a* under the two respective technologies.

Figure 4: Comparison of two pinout technologies: a) Peripheral and b) Surface

It is to be noted that, as expected, for large board sizes the surface technology always supersedes the peripheral technology. However if $p1 < p2$, then for some small board sizes, the peripheral technology might beat the surface technology.

We also make the following observation regarding the future trends in values of $p1$ and $p2$. Based on the development in processor-memory integration technology, we believe that processor-area a will reduce with time or remain around the same if memory per processor is also increased. However we do not expect pinout density per unit area or per unit length to change in comparable leaps. This implies that values of $p1$ and $p2$ will either fall or remain around current values. Without loss of generality, in this paper we use values of $p1$ in the range of 64-32 and those of $p2$ in the range of 256-128 . The higher limits are derived from the current design trends of CRAY T3D [4] and J-machine [14].

3.3 Parameterized representation of inter-cluster bisection bandwidth and channel width

In this section we develop a general scheme to represent architectural characteristics(bisection bandwidth and channel width) of the inter-cluster network. The representation is parameterized with respect to the two pinout technologies, under fixed total board area. Ideally we should consider similar characterization of the lower-level cluster-networks. However as shown in [18], lower levels have more flexibility in packaging, allowing system designer to choose channel widths and bisection bandwidths with more freedom. It is the design of the higher level inter-cluster network, under more rigid packaging constraints, that is critical in determining overall system performance.

A) Inter-cluster bisection bandwidth B: The bisection bandwidth B of an architecture is the minimum number of wires that need to be cut in order to divide the network into two equal parts. It limits the number of messages that can cross from one half to another half of the network. This puts a restriction on the peak performance of the system. Table 1 gives the bisection bandwidths of the internets for various topologies with $N = 1024$ processors under the two pinout technologies. The same information is also shown graphically at the bottom of the table. The following observations can be made. Under $O(A)$ pinout technology, we have 1)the total number of internet links(pin-count) remains constant at $1024p1$, 2) bisection bandwidth increases with cluster(board) size, and 3) larger cluster sizes with 3D, 4D or hypercube internets provide similar B. Similarly, under $O(\sqrt{A})$ pinout technology, we have 1) total internet links(pin-count) drops with larger cluster sizes, 2) bisection bandwidth falls with increasing cluster(board) size(except in the case of 2D internet where it remains constant at $16p2$), and 3) larger cluster sizes with 3D, 4D or hypercube internets provide similar B.

Broadly speaking the $O(A)$ technology can potentially support larger cluster sizes, with a corresponding growth in B, and hence performance, while maintaining same total board area and same total pinout. The $O(\sqrt{A})$ technology, on the other hand, fails to keep up B and hence performance with increasing cluster sizes.

B) Inter-cluster channel width W: Wires in a channel can be divided into two categories: wires that carry data and those that carry channel control signals. For example in Cray T3D [4], a channel width is 16(data part) + 8 (control part) = 24 wires. Actually the 16-bit data part in channels can be observed as a trend in most current generation machines [4, 7, 12]. Many factors like path-width inside routers, connector technology etc. restrict channel widths from being arbitrarily large. However it is expected that technologies in near future would allow channels to carry 32-bit or 64-bit data [16], corresponding to $W \approx 40$ and $W \approx 72$, respectively.

Table 2 gives the channel widths of the internets for various topologies with $N = 1024$ processors under the two pinout technologies. To illustrate the example, we choose the pinout densities as $p1 = 64$ and $p2 = 256$ for surface and periphery technologies, respectively. We also show the absolute channel widths in parenthesis corresponding to these values. Bottom

Table 1: Bisection Bandwidth(B) of Different Inter-cluster Network Configurations with Increasing Board Area ($A = c.a$) to design a $N = 1024$ processor system with: 1) Surface Pinout O(A) Technology and 2) Periphery Pinout O($\sqrt{A}$) Technology.

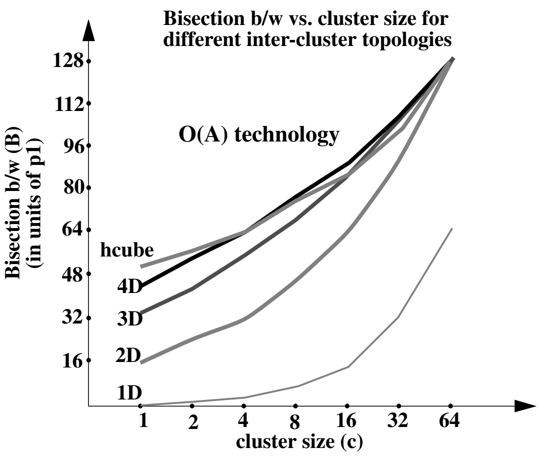
Surface Pinout Technology

Cluster (board) size c	No.of clusters (boards) M = N/c	Pincount., P in O(A) technology	Total system pin-count	1D internet B=P	2D internet $B=\frac{\sqrt{M}P}{2}$	3D internet $B=M^{2/3}P/3$	4D internet $B= M^{3/4}P/4$	Hypercue $B=\frac{MP}{2\log M}$
1	1024	p1	1024p1	p1	16.0p1	33.8p1	45.2p1	51.0p1
2	512	2p1	1024p1	2p1	22.6p1	42.6p1	53.8p1	56.8p1
4	256	4p1	1024p1	4p1	32.0p1	53.7p1	64.0p1	64.0p1
8	128	8p1	1024p1	8p1	45.2p1	67.7p1	76.1p1	73.1p1
16	64	16p1	1024p1	16p1	64.0p1	85.3p1	90.5p1	85.3p1
32	32	32p1	1024p1	32p1	90.4p1	107.5p1	107.6p1	102.4p1
64	16	64p1	1024p1	64p1	128.0p1	130.0p1	128.0p1	128.0p1

Periphery Pinout Technology

Cluster (board) size c	No.of clusters (boards) M = N/c	Pincount., P in O($\sqrt{A}$) technology	Total system pin-count	1D internet B=P	2D internet $B=\frac{\sqrt{M}P}{2}$	3D internet $B=M^{2/3}P/3$	4D internet $B= M^{3/4}P/4$	Hypercue $B=\frac{MP}{2\log M}$
1	1024	1.0p2	1024.0p2	1.0p2	16.0p2	33.8p2	45.2p2	51.0p2
2	512	1.4p2	716.8p2	1.4p2	16.0p2	30.0p2	37.8p2	40.0p2
4	256	2.0p2	512.0p2	2.0p2	16.0p2	26.9p2	32.0p2	32.0p2
8	128	2.8p2	358.4p2	2.8p2	16.0p2	23.8p2	26.7p2	25.7p2
16	64	4.0p2	256.0p2	4.0p2	16.0p2	21.3p2	22.6p2	21.3p2
32	32	5.6p2	181.0p2	5.6p2	16.0p2	18.9p2	18.9p2	18.0p2
64	16	8.0p2	128.0p2	8.0p2	16.0p2	16.0p2	16.0p2	16.0p2

The above tables are represented below through graphs



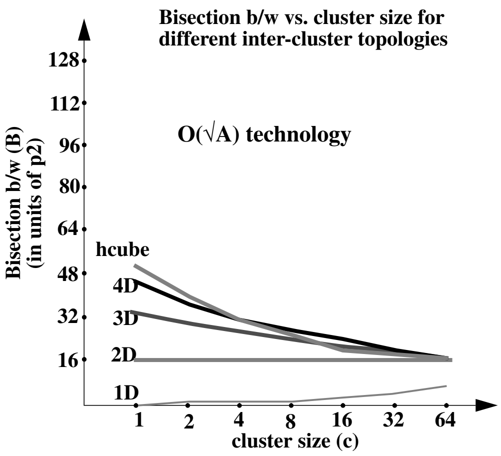


Table 2: Channel Width(W) of Different Inter-cluster Network Configurations with Increasing Board Area($A = c.a$) to design a $N = 1024$ processor system with: 1) Surface Pinout O(A) Technology and 2) Periphery Pinout O($\sqrt{A}$) Technology.

Surface Pinout Technology (p1 =64 is assumed to derive representative values, shown in parenthesis)

Cluster (board) size c	No.of clusters (boards) M = N/c	Pincount., P in O(A) technology	Total system pin-count	1D internet W=P/4	2D internet W=P/8	3D internet W=P/12	4D internet W=P/16	Hypercube $W=\frac{P}{2\log M}$
1	1024	p1 (64)	1024p1	0.25p1 (16)	0.12p1 (8)	0.08p1 (5)	0.06p1 (4)	0.05p1 (3)
2	512	2p1 (128)	1024p1	0.50p1 (32)	0.25p1 (16)	0.16p1(10)	0.12p1 (8)	0.11p1 (7)
4	256	4p1 (256)	1024p1	1.00p1 (64)	0.50p1 (32)	0.33p1(21)	0.25p1 (16)	0.25p1(16)
8	128	8p1 (512)	1024p1	2.00p1(128)	1.00p1 (64)	0.66p1(42)	0.50p1 (32)	0.56p1(36)
16	64	16p1(1024)	1024p1	4.00p1(256)	2.00p1(128)	1.33p1(85)	1.00p1 (64)	1.33p1(85)
32	32	32p1(2048)	1024p1	8.00p1(512)	4.00p1(256)	2.66p1(170)	2.00p1(128)	3.2p1 (204)
64	16	64p1(4096)	1024p1	16 p1(1024)	8.00p1(512)	5.33p1(341)	4.00p1(256)	8.0p1 (512)

Periphery Pinout Technology (p2 =256 is assumed to derive representative values, shown in parenthesis)

Cluster (board) size c	No.of clusters (boards) M = N/c	Pincount., P in O($\sqrt{A}$) technology	Total system pin-count	1D internet W=P/4	2D internet W=P/8	3D internet W=P/12	4Dinternet W=P/16	Hypercue $W=\frac{P}{2\log M}$
1	1024	1.0p2 (256)	1024.0p2	0.25p2(64)	0.12p2(32)	0.08p2(21)	0.06p2(16)	0.05p2(12)
2	512	1.4p2 (362)	716.8p2	0.35p2(90)	0.18p2(45)	0.12p2(30)	0.09p2(22)	0.08p2(20)
4	256	2.0p2 (512)	512.0p2	0.50p2(128)	0.25p2(64)	0.17p2(42)	0.12p2(32)	0.13p2(32)
8	128	2.8p2 (724)	358.4p2	0.70p2(180)	0.35p2(90)	0.23p2(60)	0.17p2(45)	0.20p2(51)
16	64	4.0p2(1024)	256.0p2	1.00p2(256)	0.50p2(128)	0.33p2(85)	0.25p2(64)	0.33p2(85)
32	32	5.6p2(1448)	181.0p2	1.40p2(360)	0.70p2(180)	0.47p2(120)	0.35p2(90)	0.56p2(144)
64	16	8.0p2(2048)	128.0p2	2.00p2(512)	1.00p2(256)	0.66p2(170)	0.50p2(128)	1.00p2(256)

Feasible system configurations(inter-cluster topology with associated cluster size c) under three channel width constraints. The value of c = '-' indicates the configuration is infeasible

W $\approx$ 24

O(A) pinout with p1=64	O($\sqrt{A}$) pinout with p2 =256
1D: c = 1,2 2D: c = 2,3 3D: c = 4,5 4D: c = 5,6 H : c = 5,6	1D: c = - 2D: c = - 3D: c = 1,2 4D: c = 2,3 H : c = 2,3

H: hypercube

W $\approx$ 40

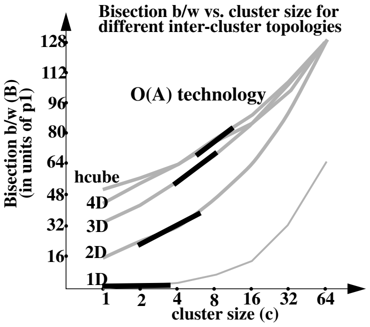
O(A) pinout with p1=64	O($\sqrt{A}$) pinout with p2 =256
1D: c = 2,3 2D: c = 4,5 3D: c = 7,8 4D: c = 9,10 H : c = 8,9	1D: c = - 2D: c = 2 3D: c = 3,4 4D: c = 6,7 H : c = 5,6

W $\approx$ 72

O(A) pinout with p1=64	O($\sqrt{A}$) pinout with p2 =256
1D: c = 4,5 2D: c = 8,9 3D: c =12,13 4D: c =17,18 H : c =13,14	1D: c = 2 2D: c = 5,6 3D: c =11,12 4D: c =20-22 H : c =12,13

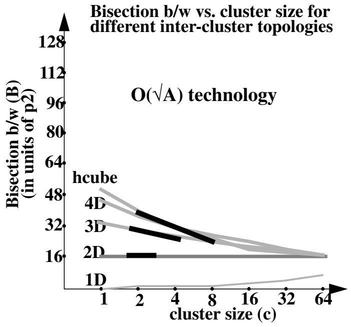
of the table shows the restrictions imposed on system design by three different channel width constraints: 1) $W \approx 24$ corresponds to current generation machines, 2) $W \approx 40$ corresponds to near future generation, and 3) $W \approx 72$ corresponds to machines predicted at the turn of this century [16]. It is to be noted that these constraints restrict allowable cluster sizes to small ranges. In $O(\sqrt{A})$ pinout technology: we observe that $W \approx 24$ restricts cluster sizes to 1-3, $W \approx 40$ restricts cluster sizes to 2-7, and $W \approx 72$ restricts cluster sizes to 2-22. A general trend to be noted is that larger channel widths allow larger cluster sizes in both pinout technologies. However, under O(A)technology this is accompanied by a growth in bisection bandwidth as well, whereas in $O(\sqrt{A})$ technology this is accompanied by a fall in bisection bandwidth.

Although future technologies will offer the choice of using wider channels($W \approx 40, 72$), system designers will still have the option of using narrow channels($W \approx 24$). Figure 5 shows an example with the bisection bandwidth curves obtained earlier(Table 1), clipped so as to show available alternatives under $W \leq 40$(allowing $W \approx 24, 40$). These form the set of design-feasible configurations.



Configurations which support channel width $W \leq 40$

O(A) pinout with p1=64	O(√A) pinout with p2 =256
1D: c = 1 -3	1D: c = -
2D: c = 2 -5	2D: c = 2
3D: c = 4 -8	3D: c = 2 -4
4D: c =5-10	4D: c = 2 -7
H : c = 5-9	H : c = 2 - 6



The darkened sections on the curves show the feasible configurations which support channel width $W \leq 40$

Figure 5: Impact of channel width constraint leading to a set of design-feasible configurations.

4 Processor and communication link speeds

Increase in processing power directly translates to greater demands on the interconnection network bandwidth. To get an estimate of demand on network bandwidth we need to know the desired sustainable *computation to communication ratio* in future systems. For this paper we choose this ratio as 1FLOP/byte as suggested in [19] (however, our analytical framework works for any computation to communication ratio). This implies that a 25MFLOPS processor demands 25MB/s of network bandwidth.

In a hierarchical system not all the traffic reaches all networks. We first explain how to compute the demanded bisection bandwidth in each network in the hierarchy. It is obvious, that a processor performance can only be sustained if the demanded bandwidth does not exceed the offered bandwidth in each network. The bandwidth demanded in each network depends on the underlying traffic pattern. We discuss two commonly used traffic patterns: uniform and localized. Then under each of these traffic patterns we show how to represent the demanded bisection bandwidth in the inter-cluster network parameterized with respect to both processor and network speeds.

4.1 Bisection bandwidth demand in each hierarchy

Let us consider a simple two-level hierarchy with N processors: each low-level cluster consists of c processors and there are $M = N/c$ clusters. Let each processor demand total interconnection bandwidth of D MB/s and let the probability of a generated request not going outside a cluster be p. We represent demanded bisection bandwidth in high-level network as B and demanded bisection bandwidth in each low-level cluster network as B_1. It can be shown [2, 18] that $B = ND(1-p)/2$ and $B_1 = cD(2-p)/2$.

To support larger clusters, the cluster bisection bandwidth B_1 should scale linearly with cluster size c. In this paper, we assume this to be the case, since packaging constraints are less rigid in lower hierarchies allowing thicker channel/buses to be used to scale up network bandwidth [18]. We focus on the more critical upper level inter-cluster network bandwidth B. Our goal is to parameterize this B as a function of processor and network speeds and then compare it with actual offered bisection bandwidth by the packaging technology(derived in sec. 3.3). We note that the demanded B is heavily dependent on the underlying p determined by the traffic patterns. We discuss this briefly in the next section.

4.2 Traffic Patterns

Traffic patterns depend on disparate factors like application characteristics, underlying architecture, mapping efficiency and programming model, etc. A *convincing* way to represent them is to actually generate them through application implementation [8]. However, for comparing a large number of design alternatives this approach is not practical. Researchers therefore rely on two synthetic traffic patterns [11, 18] to give them quick feedback on relative performance of systems.

A) Uniform traffic

Under this assumption, a message or a request generated by a processor is equally probable to go to any other processor in the system. The justification for using this pattern is 1) the application is dynamic with no communication locality, or 2) not enough compiler or human effort is expended on efficient mapping on underlying interconnection, or 3) some high-level programming constructs purposely hide details of the architecture from the user, encouraging simplicity at the cost of performance [17].

B) Localized traffic

Under this assumption, a request by a processor has higher probability of being satisfied by closer neighbors than by distant ones. This is supposed to be more representative for distributed shared memory programming with memory hierarchies and caches [6, 13]. A location often referred by a processor is *attracted* close to the processor [9], thus ensuring locality. However two open problems associated with this pattern is 1) what kind of locality model to choose? and 2) how to fairly compare different alternative designs with different cluster sizes, i.e. a model should reflect that with larger cluster size c, the probability of a request being met inside the cluster increases. We suggest and use a locality model described below. Though no claims can be made on its accuracy we believe that it represents the trend and is reasonably good for comparing among different system choices.

Let p be the probability that a generated message is for intra-cluster network. To evaluate the performance of different system topologies, we need to account for an increase in p as cluster size c increases. We consider an application that can be mapped efficiently on a grid architecture such that a processor requires to communicate only with its nearest neighbors. We can estimate the probability of a message going to each of its neighbors. We assume that the grid can be mapped onto a clustered architecture of the same size such that each cluster gets a sub-grid of size c. It is easy to estimate the fraction of messages staying inside the sub-grid. This gives an estimate on p of the cluster. If we assume a 2D grid to begin with, then in a $\sqrt{c}$ x $\sqrt{c}$ sub-grid p can be derived as $\left(1 - \frac{1.5\sqrt{c}-0.5}{c}\right)$. Thus $(1-p) \approx \frac{1.5}{\sqrt{c}}$, for large c. Table 3 summarizes p for various cluster sizes. Using the grid-model model of clustering, $(1-p)$ fraction of messages going outside a cluster would only be destined to neighboring clusters at most 1-distance away. However to avoid making our model too restrictive, we allow messages going out of the cluster to be destined uniformly to any other cluster in the system.

Table 3: Estimate of Increase in p, the Probability of a Message being for Intra-cluster network, with Increase in Cluster Size c.

Cluster Size c	p	$1-p$
1	0.000	1.000
2	0.125	0.875
4	0.375	0.625
8	0.500	0.500
16	0.656	0.344
32	0.750	0.250
64	0.820	0.180

4.3 Parameterizing inter-cluster bisection bandwidth B

In this section we show how to parameterize the demand on B in the inter-cluster network as a function of both processor and network speeds. The derivation is done under both uniform and localized traffic. We use the following example to motivate the discussion.

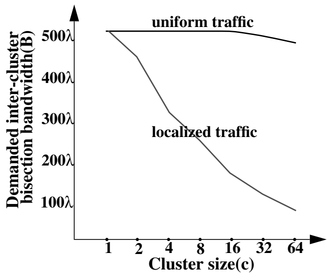
Let $D = 10$MB/s be the network bandwidth demanded by a processor in a $N = 1024$ processor system. This translates to a bisection bandwidth demand of $ND/2 = 5$GB/s under uniform traffic assumption. Let the network clock cycle time be T_c(referred to from now on as *cycle*), the time taken to transfer a bit across an internet wire. This implies that a channel with W wires can transfer W bits simultaneously in time T_c. Therefore $B/T_c = 5$GB/s, under cycle of $10ns$ implies that B be 400 wires, while under a higher cycle of $50ns$ it implies a greater B with 2000 wires.

Observe that an increase in D will capture the improvement in processor speeds while a decrease in T_c will capture the improvement in link speeds in network. To capture both these technologies together we choose to represent processor demand on network in terms of a parameter λ bits/cycle. Thus $D = 10$MB/s under $T_c = 10ns$ leads to $\lambda = 0.8$bits/cycle. Similarly, under $T_c = 50ns$ it is $\lambda = 4.0$bits/cycle. Bisection bandwidth B, under uniform traffic, can be directly represented as $B = N\lambda/2$.

In Table 4 we show the demanded B as a function of λbits/cycle for various system configurations with $N = 1024$, under both uniform and localized traffic models. The values of p used in localized traffic are taken from Table 3.

Table 4: Demanded bisection bandwidth B as a function of λ for both traffic models under various system configurations with $N = 1024$.

Cluster (board) size c	Demand under uniform traffic $B=\frac{N\lambda(1-c/N)}{2}$	Demand under localized traffic $B=\frac{N\lambda(1-p)}{2}$
1	511.5λ	511.5λ
2	511.0λ	448.0λ
4	510.0λ	320.0λ
8	508.0λ	256.0λ
16	504.0λ	176.0λ
32	496.0λ	128.0λ
64	480.0λ	92.0λ



To get representative values of λ we choose various combinations of processor and network speeds and summarize these in Table 5. Most of the values are taken from [16], and are predictions for the near future. It is to be noted that λ values will lie pretty much in the range of 4-10 bits/cycle in the near future.

4.4 Normalizing demanded bandwidth w.r.t. packaging technology

In section 3 we captured board and pinout technologies through the parameters $p1$ and $p2$(corresponding to surface and peripheral technologies respectively). We parameterized the bisection bandwidth offered by a system configuration in terms of $p1$ and $p2$. To compare the processor demanded bandwidth against the offered bandwidth we need to normalize the

Table 5: Representative processor demand on network, in λbits/channel cycle, for various combinations of processor and network speeds

Processor demand on n/w D (MB/s)	Network channel cycle time T_c (*ns*)	Processor demand on n/w λ bits/cycle
50 MB/s	10.0 ns	4.0
100MB/s	6.6 ns	6.0
250MB/s	5.0 ns	10.0
250MB/s	2.0 ns	4.0
1000MB/s	1.0 ns	8.0

demanded bandwidth also in terms of $p1$ and $p2$. This can be easily achieved by dividing the computed bandwidth demands in Table 4 by either $p1$ or $p2$(depending on surface or peripheral technology being considered).

5 Design choices

A *good* configuration is a design-feasible configuration which supports the bandwidth demanded from it. In this section we first decide on *good* configurations by comparing the bisection bandwidth offered by *design-feasible* configurations(derived in sec. 3) to the demanded bisection bandwidth(derived in sec. 4). Among all the good configurations the choice which provides best performance with cheapest cost is defined as the *best configuration* [17]. We take an example to show how to decide the set of *good* configurations. The search for the *best* configuration, however, is not a very clear-cut process. Unknown and inestimable factors like exact link costs, connector costs, and other packaging costs make this decision extremely hard. Through the same example we show some of the factors that may be considered in the choice of the best configuration. Simulation modeling might be useful in cases where relative performances are not clear.

5.1 *Good* configurations

Let us consider the following example of designing a system with 1024 processors, under the following technological parameters: $\lambda = 6.0$, $p1 = 64$, $p2 = 256$, and $W \leq 40(\approx 24, 40)$. Figure 6 shows the *good* configurations for the two pinout technologies and both uniform and localized traffic patterns. The configurations which are above the line of $\lambda = 6.0$ satisfy the bisection bandwidth demands. As we saw earlier in Fig. 5, the darkened sections on the offered bandwidth curves of various internets show the feasible configurations which support the channel width $W \leq 40(\approx 24, 40)$. It is to be noted that in all cases 1D internet is not good at all. With O($\sqrt{A}$) technology, 2D internet with $c = 1, 2$ is good under both traffic patterns. However, with O(A) technology, 2D internet is good with $c = 4, 5$ only under localized traffic. Similarly, 3D internet is good with $c = 2, 3, 4$ under O($\sqrt{A}$), and $c = 4..8$ under $O(A)$ technology. 4D internet is good with $c = 2..7$ under O($\sqrt{A}$) and $c = 5..10$

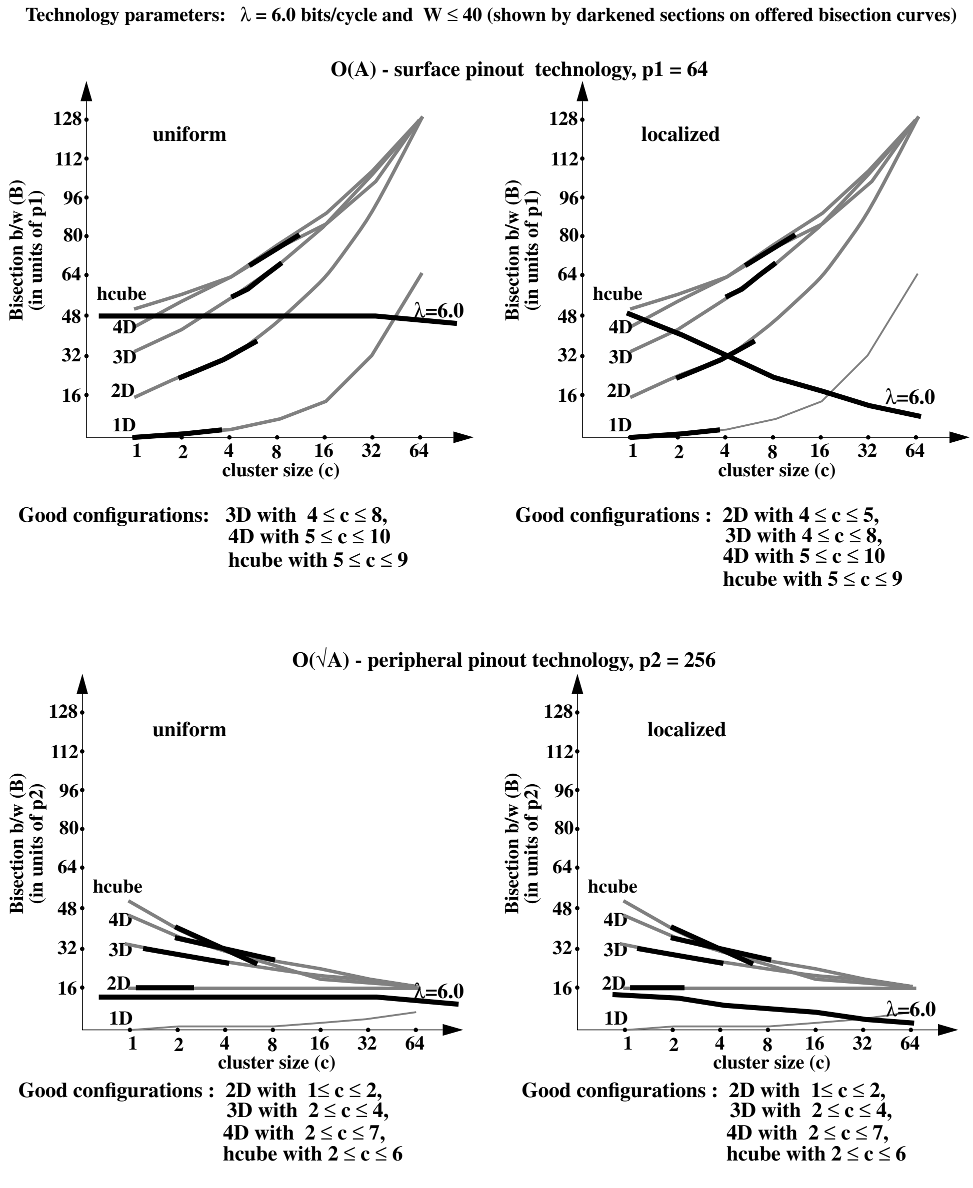


Figure 6: Search for *good* configurations from the set of *feasible* configurations under two traffic patterns: a) uniform and b) localized; and two pinout technologies: 1) Surface $O(A)$ technology and 2) Peripheral O($\sqrt{A}$) technology.

under $O(A)$ technology. Hypercube internet follows 4D closely with good choices being $c = 2..6$ under $O(\sqrt{A})$ and $c = 5..9$ under $O(A)$ technology. It is also to be observed how the localized traffic modeling adds more choices to the design space. With surface technology, the 2D internet with $c = 4, 5$ is a good configuration under localized traffic model but not so under the uniform traffic model.

It is to be noted that in this search for the *best* system we are still left with around ten to fifteen *good* choices, amongst which we have to choose one. We discuss below some guidelines and considerations to reduce the size of this set.

5.2 Choice of cluster size

We discuss the advantages and disadvantages of large cluster sizes. Larger cluster sizes have the following advantages:

1. The number of clusters(boards) in the system decreases, leading to lesser number of inter-board channels. This results in fewer connectors being required. If connector costs are high this will result in savings in cost.

2. Channel widths in internet can be higher leading to shorter messages. This will lead to low contention and low latency.

3. In case of DSM implementation with directory-based cache-coherence, the cache-coherence/ network interface logic and directory memory has to be replicated less number of times. This can yield cheaper systems.

However arbitrarily large clusters may not be possible due to:

1. The scalability restrictions of the cluster network(if it is bus-based) or the cluster interface may limit cluster sizes to small values.

2. In $O(\sqrt{A})$ pinout technology, the offered bisection bandwidth by any internet falls with growth in cluster size. Thus if demand on bandwidth were very high one has to use smaller cluster sizes.

In the following discussion, without loss of generality, we assume that cluster networks and interfaces can be scaled to support larger clusters. However, this is not an unreasonable assumption considering that our analysis indicates that in the near future cluster sizes greater than 10 will not be feasible from other constraints. This indicates that a system designer has flexibility in choosing the cluster sizes upto ten.

5.3 Choosing the *best* configuration

A) Surface pinout O(A) technology: In this technology, irrespective of internet configuration, the total volume of wires in the internet(total number of pins) is fixed. Larger clusters offer greater bisection bandwidth under this technology. Thus a reasonable approach is to go for the larger cluster sizes in each internet among the *good* choices. In our example, under $O(A)$ technology and uniform traffic, we pick among the good configurations, the 3D internet with $c = 8$, 4D internet with $c = 9$, and hypercube with $c = 8$. Some of these cluster values are chosen so as to make the system size as close to 1024 as possible, while maintaining the desired topology.

In Table 6 we compare various characteristics of these three choices: actual bisection bandwidths, average distances to be traveled, number of internet channels(connectors), and internet channel width. We see that most of these properties are quite close, making it tough to make the choice of the *best* configuration. The hypercube may be rejected on the basis of variable degree which makes scalability difficult. But this still leaves us with two choices. Simulation modeling may be used to decide amongst these. However, in this case since cluster sizes are very close and internet sizes are also similar, the analysis for the best internet is equivalent to the choice of the best dimensionality in a flat network given nodes with fixed pinouts. The equivalence holds good because clusters can be treated as supernodes connected through a flat network(internet). Such an analysis has been done by Agrawal in [1], where it was shown that for arbitrary length messages and reasonable switch delays a 3D topology will slightly outperform the 4D topology. Hence we conclude that 5-ary 3-cube cluster-8 configuration will be the best. This configuration also uses less number of internet channels(connectors) to build a less costlier system.

Table 6: Comparing Various Performance and Cost Characteristics of the Three Final Good Configurations to Design $N \approx 1024$ Processor System under Surface Pinout-$O(A)$ Technology.

Internet topology	cluster size c	Actual bisection bandwidth	Average distance traveled in internet	Number of internet channels (connectors)	Internet channel width W
3D: 5x5x5	8	4267	3.75	750	40
4D: 4x3x3x3	10	4320	3.25	864	40
Hcube: 2^7	8	4681	3.50	896	40

Demanded bisection bandwidth = 3072

B) Under Peripheral Pinout $O(\sqrt{A})$ technology: In this technology the total volume of wires in the internet(total number of pins) falls with increasing cluster size. This decreasing system volume could translate to a fall in packaging costs. Keeping in mind the benefit of large clusters we go for the larger cluster sizes in each internet among the *good* choices. This implies we choose 2D with $c = 2$, 3D with $c = 4$ and 4D with $c = 7$. In Table 7 we compare various characteristics of these three choices: actual bisection bandwidths, average distances to be traveled, number of internet channels(connectors), total number of wires in internet and internet channel widths. All three systems offer more than 150% of the demanded

Table 7: Comparing Various Performance and Cost Characteristics of the Three Final Good Configurations to Design $N \approx 1024$ Processor System under Peripheral Pinout- O($\sqrt{A}$) Technology.

Internet topology	cluster size c	Actual bisection bandwidth	Average distance traveled in internet	Number of internet channels (connectors)	Volume (number) of internet wires	Internet channel width W
2D: 25x25	2	4525	12.5	2048	180K	40
3D:6x6x7	4	6144	4.75	1512	128K	40
4D:4x4x3x3	7	6093	3.50	1152	95K	40

Demanded bisection bandwidth = 3072

bandwidth. This implies that all three systems would be operating at around $50 - 65\%$ load and therefore offer low latencies. However comparing the average distances to be traveled in the internet we find that the 3D(with $c = 4$), 4D(with $c = 7$) configurations outperform the 2D(with $c = 2$) by a factor of more than 3. This implies that basic latency in the 2D configuration would be substantially higher. In terms of total volume of wires in the internet we see that the 4D(with $c = 7$) configuration has the least number of internet wires. This 4D configuration also has least number of internet channels requiring minimum number of connectors. These observations lead us to conclude that the 4D(4x3x3x3) with clusters of size 7 would be the *best* system configuration.

6 Conclusions and Future Work

In this paper we have presented a comprehensive framework to design large multiprocessor systems. The proposed framework takes into account advancements in packaging, processor, and interconnection technologies while deciding the most optimum system configuration. The elegance of this framework lies in its parameterized representation of different technologies and then deriving an optimal system configuration for any set of technologies and constraints. The significance lies in its generality which has never been taken into consideration by earlier researchers. All earlier works have centered around fixed topology, technology, or packaging constraint.

We have applied the framework to obtain design guidelines for the currently popular k-ary n-cube cluster-c architectures. This research suggests the following guidelines to system designers for building large-scale k-ary n-cube cluster-c parallel systems:

1. Larger channel width technology will allow larger cluster sizes under both surface and peripheral pinout technologies. Some of the more specific obtained results are as follows:

   (a) The current channel width technology of 16 data bits and periphery pinout technology can use cluster sizes of 2 to 3 processors with 3D/4D internets to deliver maximum system performance.

   (b) The near future channel width technology of 32 data bits together with periphery pinout technology can go for cluster sizes of 7 with 4D internet. Similarly, under surface pinout technology, cluster sizes of 8 with 3D internet becomes optimum.

   (c) With channel width technology of 64 data bits, cluster sizes as large as 18-20 are possible. This implies that in the future a processor board technology which allows to integrate around 20 processors on a single board would be desirable.

2. Under surface pinout technology, an increase in cluster size is associated with growth in bisection bandwidth, whereas it is associated with a fall in bisection bandwidth in periphery pinout technology.

3. As processor and interconnect technologies get advanced in future, the processor demand rate on network bandwidth will be expected to stay between 4-10 bits/channel cycle.

4. For designing distributed-shared-memory system, one must consider an appropriate localized traffic model to derive optimal system configuration from a large set of design-feasible configurations.

This research has emphasized on the design of two-level hierarchical systems, with more emphasis on the design of the top-level network. However, our framework is general enough to consider more levels. In the next phase of this research, we are investigating on the design of lower levels. It will also be interesting to see how to extend our analysis to a combination of surface and periphery pinout technology as this might be a more common practice in the future.

References

[1] A. Agrawal. Limits on Interconnection Network Performance. *IEEE Trans. on Parallel and Distributed Systems*, 2(4), Oct 1991.

[2] D. Basak and D. K. Panda. Scalable Architectures with k-ary n-cube cluster-c organization. In *Proc. of the Symposium of Parallel and Distributed Processing*, 1993.

[3] C. Chen, D. P. Agrawal, and J. R. Burke. dBcube: A new class of Hierarchical Multiprocessor Interconnection Network with area efficient layout. *to appear in IEEE Trans. on Parallel and Distributed Systems*, Dec 1993.

[4] Cray Reasearch Inc. *Cray T3D System Architecture Overview*, 1993.

[5] S. Dandamudi and D. Eager. Hierarchical Interconnection Networks for Multicomputer Systems. *IEEE Transactions on Computers*, C-39(6), June 1990.

[6] D. Lenoski et al. The Stanford DASH Multiprocessor. *IEEE Computer*, pages 63–79, 1990.

[7] D. Lenoski et al. The DASH Prototype: Implementation and Performance. In *Proc. of the Int'l Symposium on Computer Architecture*, pages 92–103, 1992.

[8] D. Lenoski et al. The DASH Prototype Logic Overhead and Performance. *IEEE Trans. on Parallel and Distributed Systems*, 4(1):41–61, Jan 1993.

[9] Erik Hagersten and Seif Haridi. A quantitative comparison of efficiency for large shared memory architectures. In *Third Workshop on Scalable Shared Memory Multiprocessors.* International Symposium on Computer Architecure, 1993.

[10] W. Hsu and P. C. Yew. The Performance of Hierarchical Systems with wiring constraints. In *Proc. of the Int. Conf. on Parallel Processing*, Aug 1991.

[11] W. Hsu and P. C. Yew. The Impact of Wiring Constraints on Hierarchical Network Performance. In *Proc. of the Int. Parallel Processing Symposium*, Mar 1992.

[12] Intel Corporation. *Paragon XP/S Product Overview*, 1991.

[13] Kendall Square Research. *KSR Technical Summary*, 1992.

[14] Michael D. Noakes, Deborah A. Wallach, and William J. Dally. The j-machine multicomputer: An architectural evaluation. In *Proc. of the Int. Symposium on Computer Architecture*, pages 224–235, 1993.

[15] K. Padmanabhan. Effective Architectures for Data Access in a Shared Memory Hierarchy. *Jour. of Parallel and Distributed Computing*, 11, 1991.

[16] David A. Patterson. Observations in Massive Parallelism - Trends and Predictions for 1995 to 2000. Technical Report 93-87, DIMACS, Sept 1993.

[17] M. T. Raghunath. *Interconnection Network Design Based on Packaging Considerations.* PhD thesis, U. C. Berkeley, Nov 1993.

[18] M. T. Raghunath and Abhiram Ranade. Designing interconnection networks for multilevel packaging. In *Proc. of the Supercomputing*, pages 772–781, 1993.

[19] E. Rothberg, J. P. Singh, and Anoop Gupta. Working Sets, Cache Sizes, and Node Granularity Issues for Large-Scale Multiprocessors. In *Proc. of the Int. Symposium on Computer Architecture*, pages 14–25, 1993.