Designing Clustered Multiprocessor Systems under Packaging and Technological 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

Packaging technologies impose various physical constraints on bisection bandwidth, pinout, and channel width of a system whereas processor and interconnect technologies lead to certain demanded throughput on network bisection. Earlier studies in literature have proposed hierarchical and clustered interconnections by considering the effect of limited packaging constraints. Pinout technologies and capacity of packaging modules have been ignored, often leading to configurations which are not design-feasible. In this paper, we solve this design problem by proposing a new *supply-demand optimization framework*. This generalized framework uses parameterized representation of processor board area, pinout technologies (periphery or surface), channel width, and channel speed. The family of flat k-ary n-cube topologies and their clustered variations (k-ary n-cube cluster-c) are evaluated to derive optimal configurations which can lead to cost-effective design of scalable parallel systems using wormhole-routing. The analysis identifies processor board area, channel width, and pinout density as critical parameters. The study indicates that cluster-based parallel systems can deliver better performance with lower cost. It is predicted that optimal configurations for future systems will be cluster-based (2-10 processors per cluster) with 3D/4D/5D inter-cluster interconnection. This framework is quite general to capture technological trends of future years. The framework is validated by the design solutions of current machines using contemporary technologies.

Keywords: Multiprocessor System Design, Scalable Systems, Clustered Architectures, Hierarchical Organization, k-ary n-cube Interconnection, Packaging Constraints, Pinout Technology.

[1] This research is supported by the National Science Foundation Grant # MIP-9309627. A preliminary version of this paper has appeared in the proceedings of the *International Conference on Parallel Processing*, 1994.

Contents

1 Introduction

Rapid developments in the field of processor, interconnect, and, packaging technologies make the task of efficient design of large multiprocessor systems a difficult one [1, 2, 7, 11, 12, 14, 17]. Design guidelines to derive the most optimal interconnection have to take into account changes in these technologies to yield best results. Several previous studies have considered packaging constraints [3] while selecting the best system configuration. These studies include Dally's [11] analysis of k-ary n-cube interconnection under a VLSI model with constant bisection bandwidth, and Abraham and Padmanabhan's study [1] under a constant pinout from a processing node. Agrawal's [2] analysis of the same class of networks considers three different constraints: constant bisection width, constant channel width, and constant pinout and is based on a more comprehensive model of node and wire delays. However, Ranade [23, 24] and Yew [15] argued that neither Dally's VLSI model with limited bisection bandwidth nor the limited pinout model as proposed in [1, 2] is adequate for very large systems. Both models confine to only one level of packaging hierarchy whereas large systems typically employ several levels of packaging.

It was demonstrated in [24] that considering two-level hierarchical/clustered systems widen the scope of the design-space. The architectural levels can be chosen to closely match the packaging hierarchy leading to better designs. 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[12, 20, 24], MINs and n-hop networks [24], two-level systems with k-ary n-cube topologies[4], and combination of bus and mesh/hypercube networks[14]. However, most of these analyses did not take into account packaging and interconnecting constraints. Guidelines developed in [14, 24] were based only on fixed board sizes with fixed pinouts, and did not consider changes in board sizes and alternate pinout technologies. The impact of the changes in processor and interconnect technologies in relation to packaging technology was also not studied.

Analogously in our analysis we focus on two-level clustered architectures based on a k-ary n-cube cluster-c [4, 6] interconnection. These are clustered extensions of the flat k-ary n-cube systems. Our design model is more comprehensive and flexible in terms of considering packaging and interconnect constraints. Among other things we allow varying board sizes, only reasonable channel widths, and flexible pinout from board which depends on its size. System design under alternate pinout technologies like periphery and surface is also explored. The increase in level of integration with time in terms of smaller chips, larger wiring boards, and wiring and pinout densities is also captured and the impact of each studied on system design.

In this paper we propose a new framework, as shown in Fig. 1, for designing and developing clustered architectures. We consider a packaging technology with two levels: the lower level made of chips/multi-chip modules and the upper level made up of processor wiring boards. We analyze the problem of designing an efficient system with N processors using two-levels. 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 by us as *design-feasible* configurations. It is clear that only the design-feasible configurations can translate into real machines. Each of the design-feasible configurations offer a level of performance depending on its architectural characteristics. 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, to sustain processor performance in terms of throughput places a demand on the size of the required bisection bandwidth [4, 24] of the system. The design-feasible configurations which offer performance greater than what is demanded are defined as *good* configurations. Among the good choices the one which provides desired performance at optimal cost is defined as the *best* configuration. Our goal is to derive this best topology. Simulation modeling is also employed to determine exact performance to decide between very close *good* candidates.

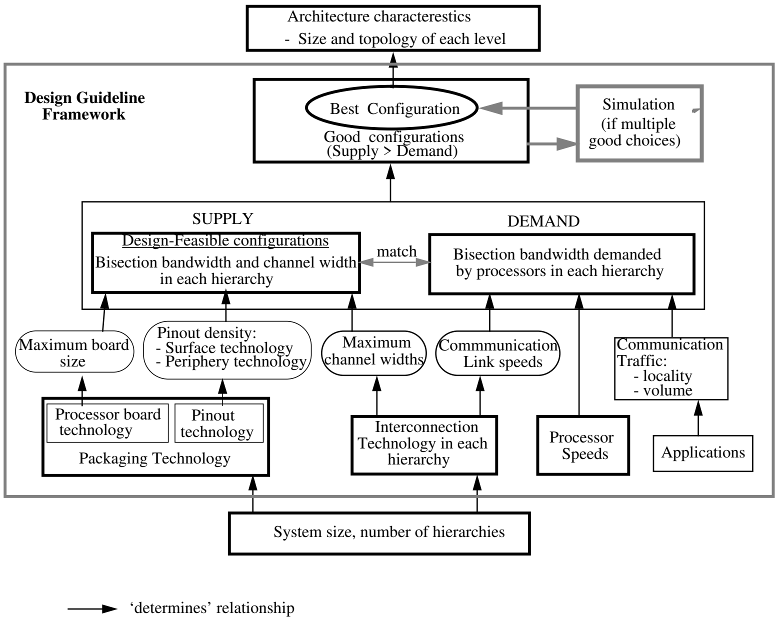

Figure 1: A supply-demand optimization framework for designing and developing scalable architectures under varying processor, interconnection, and packaging technologies.

The paper is organized as follows. In Sec. 2 we present the two-level k-ary n-cube cluster-c architecture used in later discussion. Processor speeds, communication link speeds and the performance demand are discussed in Sec. 3. In Sec. 4 we discuss the trends in growth of processor board sizes, alternate pinout technologies, and channel width technology. In Sec. 5 expressions for offered channel width and bisection bandwidth and the design-feasible configurations under different packaging and processor and interconnect technologies are derived. In Sec. 6 we derive the good configurations and discuss important considerations in choosing the best configuration. Finally, concluding remarks and future work is presented in Sec. 7.

2 Two-Level Clustering with k-ary n-cube cluster-c Organization

Many current parallel systems like the CRAY T3D [10], Intel Paragon [16], and the Stanford DASH [13] are taking a two-level clustering approach. Recently, we have introduced a new k-ary n-cube cluster-c organization [4, 5, 21] to capture this upcoming trend in building scalable parallel systems. In this organization, the lower level consists of k^n clusters of processors. These clusters are 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 leading to a total of $N = (k^n.c)$ processors in the system. This interconnection achieves two main design objectives: a) direct internet provides easy scalability and b) processor clusters allow easy exploitation of communication locality. Figure 2 shows the overall configuration of such a system.

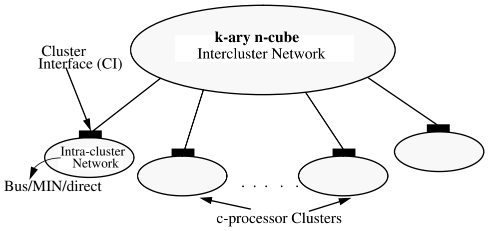

Figure 2: Two-level clustering with k-ary n-cube cluster-c organization.

The interconnection within a cluster (also referred to as intra-cluster network or *intranet*) can be chosen as bus/MIN/star network/direct network as shown in Fig. 3. Each cluster is connected to the rest of the system through a cluster interface. The main task of the cluster interface is to handle the volume of communication to/from the cluster. Other functionalities may be added to the cluster interface to efficiently implement various communication, synchronization, and cache-coherence operations to enhance overall system performance [13, 21]. However, such discussion is beyond the scope of this paper.

The memory in the system is distributed physically across the clusters to increase overall memory bandwidth. Organization of memory within a cluster is left as an open choice depending on the size of cluster and its configuration. The exact nature of this distribution is not critical to the analysis presented in this paper. Therefore, without loss of generality we assume the cluster

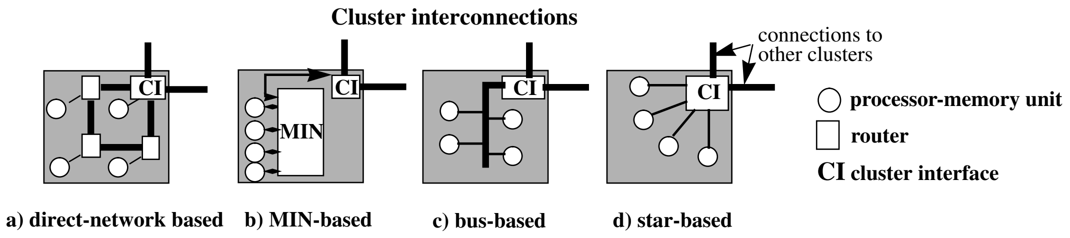

Figure 3: Four possible cluster interconnections under k-ary n-cube cluster-c organization.

memory to be distributed uniformly amongst the processors in the cluster.

Architectural Alternatives

To build an N-processor system, there are vast number of possible alternatives. The degrees of freedom are: number of processors (size) in each cluster and topologies of the two levels (inter-cluster and intra-cluster). Let us consider designing a system with $N = 1024$ processors. This system can be designed 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. Having fixed the size of each level, there is still freedom to vary the topology in each level. For example, in a system with 64 clusters with 16 processors each, the topologies can be: 4-ary 3-cube internet with bus-based clusters, 8-ary 2-cube internet with MIN-based clusters, and so on. Our objective is to select the configuration which, at a minimum cost, offers best system performance.

It is easy to observe that the flat k-ary n-cube can be derived as a special case of the k-ary n-cube cluster-c family by choosing cluster size $c = 1$. Since inter-cluster interconnections are more expensive than intra-cluster interconnections [24], in this paper we primarily emphasize on internet and cluster size. In the following sections, we develop a supply-demand optimization framework to derive optimal k-ary n-cube cluster-c organizations ($c \geq 1$). Table 1 provides a summary of the main symbols and notations used in this paper.

3 Demand on Network Bisection Bandwidth

In this section we first analyze the impact of processor speed on the rate of injection of traffic into the interconnection network. We then show the impact of communication link technology on the bisection bandwidth of a system. The demanded bisection size of the system to support a given processor speed and communication link speed technologies is then presented.

Table 1: Summary of Important Symbols and Notations Used.

a	Unit area to hold a processor chip, associated memory, and router logic
b	Area of a board in units of a (number of processors in a board)
$b' = b/c$	Number of clusters in a board
c	Size of each cluster
n	Dimensionality of inter-cluster network
p_p	Pinout density under periphery pinout technology
p_s	Pinout density under surface pinout technology
t	Time to transfer a bit across a wire (indicates communication link technology)
u_b	Percentage utilization of board area b
u_p	Percentage utilization of available board pinout P_b
B_b	Bisection size of a board (number of wires)
D	Average number of bits injected by a processor per second into the network (indicates processor technology)
N	Total number of processors in a system
$N' = N/c$	Total number of clusters in system
N_{boards}	Total number of processor boards used in a system ($\geq N/b$)
P_b	Total pincount from a board (depends on pinout technology)
W	Inter-cluster channel width (number of wires)
W_b	Maximum value of W constrained by board bisection B_b
W_p	Maximum value of W constrained by board pinout P_b
β_l^d	*Demanded* bisection size in the intra-cluster (*lower*) level network
β_u^d	*Demanded* bisection size in the inter-cluster (*upper*) level network
β_l^s	Offered (*supplied*) bisection size in the intra-cluster (*lower*) level network
β_u^s	Offered (*supplied*) bisection size in the inter-cluster (*upper*) level network
λ	Dt (reflects both processor and communication link technologies)

3.1 Effect of Processor Speed

With advances in processor technology, the rate at which processors can execute instructions keeps going up. As a general observation in applications, an amount of computation is associated with a quota of necessary communication, expressed as the *communication to computation ratio(r)* [25]. Intuitively it is clear that with increase in processing speeds computations take less time. This requires the associated communications to also occur faster leading to greater demands on the network bandwidth. With a given processor technology, let us define D bits/sec as the rate of traffic which each processor injects into the network. An increase in processing power leads to a higher value of D. A typical estimate of the range of communication to computation ratio(r), as suggested in [25], is $r = 1$ to 5 Flop/byte. For $r = 1$, this implies that a 25MFlops processor would inject traffic into network at the rate of 25MBytes/sec or 200Mbits/sec.

3.2 Effect of Interconnect Speed

The bisection size β of a network is defined as the minimum number of wires that need to be cut in order to divide the network into two equal parts [11]. It limits the number of bits that can cross from one half to another half of the network. For example, in a k-ary n-cube interconnection with W-bit wide channels, the bisection size can be derived as $\beta = 4k^{n-1}W$ [11]. Let t be defined as the *cycle time*, the time to transfer a bit across a wire in seconds. In other words, a set of W parallel wires can transfer W bits simultaneously in time t. The bisection size β of a network corresponds to a bisection bandwidth of (β/t) bits/sec. A reduction in t allows more data to be sent across any wire in a given time and increases the bisection bandwidth for the same bisection size.

3.3 Characterizing Demanded Bisection Size

Given that each processor injects messages into its cluster network at D bits/sec, let us derive the expressions for minimal demanded bisection bandwidth in the internet and intranet to sustain the associated communication requirement. Represent the traffic rate to be supported across the bisection in the internet (upper-level network) as B_u^d and that in each intranet (lower-level network) as B_l^d. Given that the probability of a generated request not going outside a lower-level network as p, the effective injection rate of a cluster into the inter-cluster network can be derived as $D_{clus} = cD(1-p)$ bits/sec. This leads to a combined total injected traffic of $(D_{clus}N/c)$ into the internet. Under uniform traffic assumption, a generated message by a processor is equally probable to be destined for any other processor in the system. Thus, p can be derived as c/N which approaches 0 for large systems with small cluster sizes. It also indicates that, on average, half the messages are destined to clusters on the other half of the system and thus require to cross the network bisection.

Thus, the minimal bisection bandwidth demanded in the inter-cluster network can be derived as:

$$B_u^d = \frac{D_{clus}N/c}{2} = \frac{D(1-p)N}{2} = \frac{DN}{2} \tag{1}$$

All expressions in this section have been simplified by assuming $p \to 0$. A localized traffic model would yield $(p > c/N)$ leading to lesser demanded bisection B_u^d. However, while designing a general-purpose machine without prior knowledge of the nature of the applications to be executed on it, an uniform traffic model is considered more representative [24].

The demand on bisection bandwidth inside a cluster can be divided into three components, shown as flows f_1, f_2, and f_3 in Fig. 4. The flow f_1 corresponds to the local traffic across bisection and is given by $cDp/2$. The flow $f_2 = cD(1-p)/2$ is the fraction that goes to the upper level across the bisection, and the flow $f_3 = cD(1-p)/2$ is a similar contribution coming from the upper level that crosses the bisection. This leads to a minimal bisection bandwidth demand in the intra-cluster network as:

$$B_l^d = cD(2-p)/2 = cD \tag{2}$$

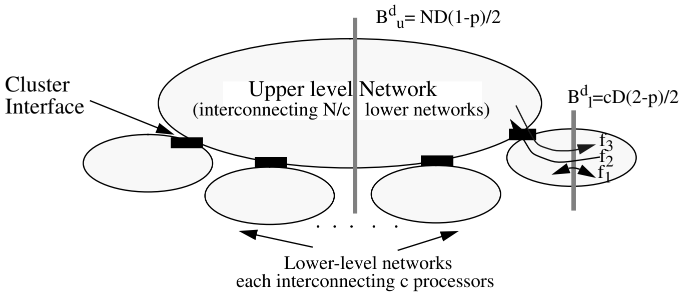

Figure 4: The demanded bisection bandwidths in the inter-cluster (upper-level) network and inside each cluster (lower-level) network under an uniform traffic pattern.

The above expressions B_l^d and B_u^d give the minimal transfer rate to be supported at the network bisections of the upper and lower level networks, respectively. The actual size of these bisections, in terms of number of wires, to support the rate also depends on the speed of the communication links. Let us define the minimal bisection size in the upper-level network as β_u^d. Let t_u be the cycle time in the inter-cluster network. This indicates $\beta_u^d = B_u^d t_u$. Observe that an increase in the value of D captures the advancement in processor speed while a decrease in the value of t_u captures that of link speeds. To capture both these technological advancements together, we introduce a parameter $\lambda_u = Dt_u$. This leads to:

$$\beta_u^d = \frac{Dt_u(1-p)N}{2} = \frac{\lambda_u(1-p)N}{2} = \frac{\lambda_u N}{2}. \tag{3}$$

Analogously, we define the bisection size to handle the traffic rate B_l^d across a lower-level cluster network as $\beta_l^d = B_l^d t_l$, where t_l is the cycle time in the cluster network. Representing

$\lambda_l = Dt_l$, we can write

$$\beta_l^d = c\lambda_l(2-p)/2 = c\lambda_l \tag{4}$$

For the remaining discussion in this section, we treat λ_u and λ_l as same and refer to it as simply λ. It is to be noted that if processor and link speeds increase in the same proportion then λ does not change. This leads to no change in the required β_u^d and β_l^d. To derive representative values of λ in current and future systems, we consider predictions made by Patterson [22]. As shown in Table 2 the future values of λ will broadly lie in the range of 4-10.

Table 2: Sample values of λ for various representative combinations of processor and network speeds.

Processor demand on n/w D (Mb/s)	Network channel cycle time t (*ns*)	λ = Dt
400	10.0 ns	4.0
800	6.6 ns	5.3
2000	5.0 ns	10.0
2000	2.0 ns	4.0
8000	1.0 ns	8.0

To support larger sized clusters, the intra-cluster bisection bandwidth should scale linearly with size c. Since packaging constraints are less rigid in lower hierarchies, it is possible to provide thicker channel/buses for higher bandwidth [24]. Hence in this paper, we focus on minimum bisection size in the upper level, β_u^d. The demanded bisection is parameterized with respect to λ thus capturing advances in both processor and interconnection speeds. In the following section we compare this demanded bisection size with that being offered by the packaging technology.

4 Modeling and Parameterization of Packaging Technologies

The packaging of high performance systems has an extreme impact on their performance. It is quite often the case that system level design is dictated by available packaging technology. We first present a model to capture hierarchical packaging schemes. We then focus on a two-level packaging with the first level as clusters inside a board and the second as across the boards. Current trends in board and pinout technologies are investigated and a general scheme is presented to parameterize the architectural characteristics with respect to these technologies.

4.1 Hierarchical Packaging Model

Some earlier models, proposed to capture packaging constraints, like the VLSI model with limited bisection bandwidth as proposed by Dally [11] and the limited pincount model as proposed by Agrawal [2] deal with only one level of packaging. Large parallel machines, on the other hand, typically employ several levels of packaging. There are inherent technological constraints that limit VLSI die sizes, board areas, number of boards per card-cage, and number of card-cages per cabinet, etc. Hence, multi-level hierarchies are inevitable for packaging large parallel systems consisting of thousands of processors.

Similar to the characterization of multi-level packaging hierarchies in [23], we model the packaging hierarchy as consisting of l levels $0, 1, ..., l-1$. A packaging module in level-i is referred to as a m_i. Each m_i contains some number of m_{i-1} modules and these in turn contain m_{i-2} modules and so on. We treat processing elements, memory units and routing nodes as the lowest level m_0 modules. A module m_i is associated with a property vector $prop_i$ = $(dims_i,\ B_i,\ P_i,\ t_i)$ which describes its various technological capacities. These are as follows:

- $dims_i$: the range of *sizes* and *dimensions* that a packaging module m_i can support. This parameter is used to determine other geometric characteristics like cross-section, area, volume, and periphery.

- B_i: the *bisection bandwidth* available in a module m_i. The cross-section size and the cross-section wire-density at level-i decide the available B_i, which is the maximum bisection bandwidth available to the sub-topology of m_{i-1} modules that can be contained within a m_i module.

- P_i: the *pincount* available with a m_i module. These pins are used to interconnect various modules at level-i to comprise a sub-topology to be contained within a m_{i+1} module. This parameter P_i depends on the pinout technology, physical dimensions of m_i, and the pinout densities supportable at m_i. In Sec. 4.2 we analyze different pinout technologies and their impact on our design guidelines.

- t_i: the *time* to transfer a bit over a level-i wire.

This generalized packaging model can capture and characterize a wide range of packaging hierarchies and technologies by appropriately choosing the parameters. With respect to our two-level k-ary n-cube cluster-c organization shown in Fig. 2, we consider a two-level packaging hierarchy. The lower level modules consists of processors inside a cluster. The dimensions of this cluster-module depends on the size of the cluster and the level of processor-memory integration. Boards represent the upper-level modules. Depending on the cluster size and board area, multiple clusters may be put onto a single board. The inter-cluster network may therefore be partly intra-board and partly across boards. We do not allow a single cluster to span across multiple boards. In this research we study the interplay of the cluster-modules and board-modules on overall system design as a function of cluster size, board size, and technological advances.

4.2 Parameterizing Packaging Technologies

In this section we discuss current trends in board and pinout technologies. Then we present a general scheme in which architectural characteristics like bisection bandwidth and channel width are parameterized with respect to the board and pinout technologies.

4.2.1 Processor board technology

Processor boards cannot be arbitrarily large in size. Their size is restricted by electrical, mechanical, and board fabrication constraints. Let us define the area required by a processor chip, its local memory, and a per-processor fraction of the network and other interface logic, to be a. We treat this area as a square with a perimeter of $4\sqrt{a}$ and measure all board sizes in terms of a. Given a board of size (ba), we can accommodate up to b processors on it. This can represent $b' = b/c$ clusters with c processors in each cluster. For sake of simplicity, in further discussions we drop the area unit a and refer to a board area of (ba) as simply b.

Over the years board sizes have grown in physical dimensions. We now have systems like MIT J-machine [19] which put 64 processors on a board. However, the J-machine uses fine-grained nodes [9] with integrated processor and router organization. The memory provided per processor is also small compared to other contemporary commercial systems and research prototypes. The current trend in building large parallel systems is to use off-the-shelf commodity processors and to provide a few megabyte of memory per processor. Hence, for illustrating our framework we consider board sizes to hold a maximum of 16 processors. To build a system with N processors we need a total board area $\geq N$. However, board area being precious [6], we suggest using a total board area $= N$. All boards used in a system are assumed to be of the same size and $1 \leq b \leq 16$. The total number of boards used in designing an N-processor system is defined as $N_{boards} = N/b$. Another important consideration in the design problem is the on-board bisection bandwidth, B_b. This parameter reflects the maximum number of wires that can be laid across the bisection of the board. Multi-layered wiring is commonly employed to support a higher bisection bandwidth density.

4.2.2 Pinout Technologies

The pin-count P_b out of a board has a direct influence on the data volume that can flow in/out of a given processor board. Currently two different types of technologies are being employed by the computer industry:

a) Peripheral pinouts: This is the traditional technology [27, 28]. 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 p_p to be the pin-count that can be supported from a perimeter of length $4\sqrt{a}$ (corresponding to the perimeter of a square board area a) using this technology. This implies that a board area of (ja) would have a peripheral pinout of $P_b = p_p\sqrt{j}$.

b) Surface pinouts: In this technology the surface of the board is utilized for external connections. Representative examples are electronic interconnections using elastomeric connectors [19] and optical interconnections [18, 26]. Under this technology the pin-count from a board of area A is $O(A)$. Let us define parameter p_s to be the pin-count that can be supported on a given board area a using surface pinout technology. This implies that a board area of (ja) would have a surface pin-count of $P_b = jp_s$.

It is natural to expect that the future trend will be towards a mixed pinout technology employing a convenient combination of surface and peripheral pinouts. The pincount under mixed technology may be expressed as a function $f(p_p, p_s)$ of the periphery and surface pinout densities. The exact nature of this function is difficult to predict as it depends on specific fabrication advancements. However, reasonable suggestions may be a *max* function or a *weighted mean.* Without loss of generality, in this paper we present our results with respect to the peripheral and surface technologies. The results for a mixed pinout technology can be derived by simple extension.

Figure 5 shows the growths of pincount with board area under the respective surface and peripheral technologies for different relationships between p_s and p_p. The vertical clipping line in each graph reflects that the board size cannot grow continuously and is limited by some maximum size.

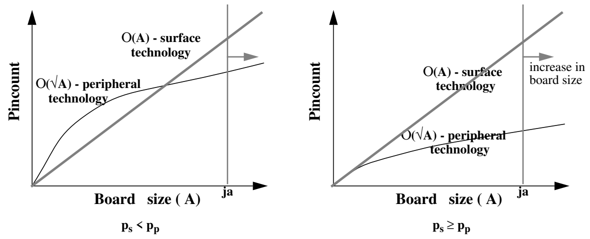

Figure 5: Growth of pincount with board area under peripheral and surface technologies shown for two different relationships between p_p and p_s. The parameters p_p and p_s reflect the pincount that can be supported from a unit board area a under periphery and surface pinout technologies respectively.

As expected, for large board sizes the surface technology always supersedes the peripheral technology. However if $p_s < p_p$, then for smaller board sizes, the peripheral technology can beat the surface technology. Representative values of p_p and p_s derived from the current design trends of CRAY T3D [10] and J-machine [19] are 256 and 64, respectively. Future values of these parameters will depend on the advances made in the respective technologies. To cover a wide spectrum of possibilities and show the impact under different situations, we also use projected values of $p_p = 512$ and $p_s = 256$.

4.2.3 Channel Width Technology

Most current machines have 16-bit data channels. This corresponds to a channel width of $W \approx 24$ including control, acknowledgment, and parity wires. Many factors like path-width inside routers and connector technology restrict channel widths from being arbitrarily large. In the near future it is expected that technologies would allow channels to carry 32-bit and 64-bit data [6, 22], corresponding to $W \approx 40$ and 72, respectively. Larger channel widths would make effective message lengths shorter leading to lower message latencies and lesser contention in routing [8, 11] thus yielding better system performance. In this paper we consider design under all three channel widths.

4.3 Under-Utilization of Board Area and Pinout Capacity

It is not always necessary to either fill up a processor board to its maximum capacity or to utilize all the pins coming out from it. In our k-ary n-cube cluster-c organization, consider the $N' = N/c$ clusters to be interconnected by a generalized k-ary n-cube topology with different radix in each dimension. This leads to $N' = k_1$ x k_2 x... x k_n. The interconnection is assumed to be laid out across the boards. Consider a sub-topology of the inter-cluster network with γ clusters, requiring an area of $c\gamma$, to be placed on a board. Let P wires be required to connect the chosen sub-topology to other clusters outside the board. This requires $b \geq c\gamma$ and $P_b \geq P$. A perfect matching of $b = c\gamma$ and $P_b = P$, where both board capacity and pinout are fully used, is usually difficult to achieve. On choosing an optimal board size with capacity $b = c\gamma$ may imply $P_b > P$, leading to *under-utilization of pin-capacity.* Defining $u_p = P/P_b$, this under-utilization factor can be expressed as $(1 - u_p)$. Similarly choosing a board with optimal pincount $P_b = P$ may lead to $(b > c\gamma)$, which reflects *board area under-utilization* or *under-population* of board estate. Defining $u_b = c\gamma/b$, the factor of under-population can be expressed as $(1 - u_b)$.

It is to be noted that under-population of board area allows fewer nodes to share the pincount from a board making wider channels possible. The disadvantage is clearly the wastage of precious board estate leading to larger number of boards, greater system volume, and higher cost. In the discussion that follows we assume board area to be fully utilized unless mentioned otherwise. Under-utilization of pin-capacity is used if larger board sizes are required to fit the desired topology but the resulting larger pincount cannot be utilized with wider channels as discussed in sec. 4.2.3. It is also possible to choose a design where under-utilization of both board and pinout-capacity is used. This is obviously not required, because we can always select a board size where one of the capacities is optimally utilized but not the other. Such a design rule can be captured as $(u_b = 1) \vee (u_p = 1)$.

5 Impact of Packaging on Architectural Parameters

In this section we determine possible channel widths and bisection bandwidths to design k-ary n-cube cluster-c systems for a given board, pinout, and channel width technologies. From the expressions we derive in this section for arbitrary cluster sizes, the results for flat architectures can be obtained by choosing $c = 1$.

5.1 Offered Inter-Cluster Channel Width

Let the sub-topology of the inter-cluster network $N' = N/c = k_1$ x k_2 x $\cdots k_n$ being placed on a board be ($b' = b_1$ x b_2 x$\cdots$x b_n), where $b_i \leq k_i, \forall i$ and some b_i's may be 1 to depict that only one processor exists on the board along that dimension. This implies that no inter-cluster dimension can be fully contained within a board. This is reasonable to expect for large system sizes. Figure 6 shows an example inter-cluster interconnection ($N' = 4$ x 4 x 3) being placed on 12 boards each with a capacity for holding four processor clusters. The sub-topology placed on each board is $b' =$ 2 x 2 x 1. Let the number of rows in dimension i of the sub-topology be defined as r_i. This can

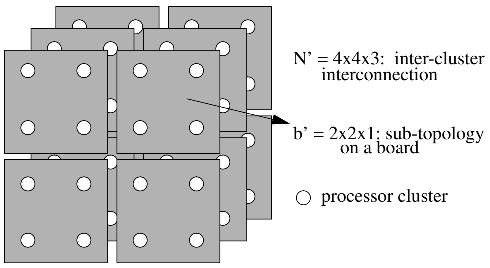

Figure 6: An example 4x4x3 inter-cluster interconnection being placed on 12 boards each with a capacity of holding 4 processor clusters. The sub-topology placed on each board is 2x2x1.

be derived as $r_i = b'/b_i$. Note that inter-board channels go out/come in from the two ends of each row leading to $4r_i$ inter-board channels along dimension i. Hence, the total number of inter-board channels from all n dimensions is derived as $4\sum_{i=1}^{n} r_i = 4b'\sum_{i=1}^{n}(1/b_i)$. To support a channel width of W, a total pincount of $4b'W\sum_{i=1}^{n}(1/b_i)$ is required from each board. Thus, the maximum supportable inter-board channel width from pinout restrictions, denoted by W_p, is derived as:

$$W_p = \frac{P_b}{4b'\sum_{i=1}^{n}(1/b_i)} \tag{5}$$

Let us now consider the required on-board bisection size to support W width channels in the (b_1 x b_2 x... x b_n) sub-topology on a board. Clearly, the available board bisection should be able to support the largest bisection in the sub-topology. The largest bisection in the sub-topology is the one orthogonal to the smallest dimension. However, a smallest dimension j with $b_j = 1$, implying only one processor along that dimension, does not require any on-board bisection. Thus, the largest bisection occurs orthogonal to dimension j such that b_j is the smallest dimension not equal to one. There are b'/b_j nodes nearer to this bisection and assuming bidirectional channels we have $2b'/b_j$ channels crossing this bisection. With channel width of W, it requires $2Wb'/b_j$ wires across this bisection. With a board bisection B_b, the maximum supportable intra-board channel width from board bisection restrictions, W_b, is given by,

$$W_b = \frac{B_b}{2b'/b_j} = \frac{B_b b_j}{2b'} \tag{6}$$

Assuming that the sub-topology on the board is regular, i.e. $(\forall i : b_i = b'^{1/n})$, this can be simplified to,

$$W_b = \frac{B_b}{2b'^{(n-1)/n}} \tag{7}$$

Similarly, Eqn. 5 can be simplified to,

$$W_p = \frac{P_b}{4nb'^{(n-1)/n}} = \frac{P_b}{4n(b/c)^{(n-1)/n}} \tag{8}$$

Thus, the channel width supported in a system is restricted by two constraints: W_p from pinout and W_b from board bisection constraints, respectively. This can potentially lead to different inter-board and intra-board channel widths. For designing an architecture with uniform inter-cluster channel width, we choose $W = \min(W_p, W_b)$. With multi-layered interconnection, it is reasonable to assume on-board connection density to be higher than off-board density. This leads to channel width being limited by the pinout constraint resulting in $W \equiv W_p$. Unless otherwise specified we therefore use W and W_p interchangeably. In the following discussion we assume full utilization of board area and pinout resources. Design with under-utilization is analyzed later.

From the expression for W_p in Eqn. 8, it can be observed that the channel width is determined by board size, cluster size, pinout technology from board, and dimensionality of the inter-cluster network. It does not depend on the total system size. For a given pinout technology and board size let us consider the impact of varying the inter-cluster dimensionality while keeping cluster size fixed. In order to support larger dimensional networks, we need more channels. This leads to thinner channels with a fixed pinout from a board. Similarly, for a given pinout technology and board size, decreasing the cluster size while keeping inter-cluster dimensionality fixed also leads to thinner channels. A smaller cluster size results in larger number of clusters on board. This implies that pinout from the board gets shared among more clusters leading to thinner channels. We summarize these as:

Observation 1 *For a given pinout technology (P_b) and board size (b), the offered channel width falls with*

- *Increase in the inter-cluster network dimensionality (n) while keeping cluster size (c) fixed or*
- *Fall in the cluster size (c) while keeping the inter-cluster dimensionality (n) fixed.*

Figure 7 shows the trend as cluster size is increased for different inter-cluster dimensionalities. In the figure each board is assumed to be large enough to hold exactly one cluster ($b' = 1$) and channel widths up to 128 lines have only been shown. For a given cluster size, offered channel width drops with increasing inter-cluster dimensionality. It can be observed that the drop in channel width is more significant at lower dimensions (from 1D to 2D) than at higher dimensions (from 3D to 4D). Therefore for reasonably higher dimensions, the channel widths are expected to be close. Similar results are also obtained for $b' > 1$. For a given message size, a wider channel reduces the effective message length in the inter-cluster network. This has a potential to reduce message contention leading to smaller message latency. Thus, it is desirable to choose the largest channel width supportable by technology at a given time as discussed in sec. 4.2.3.

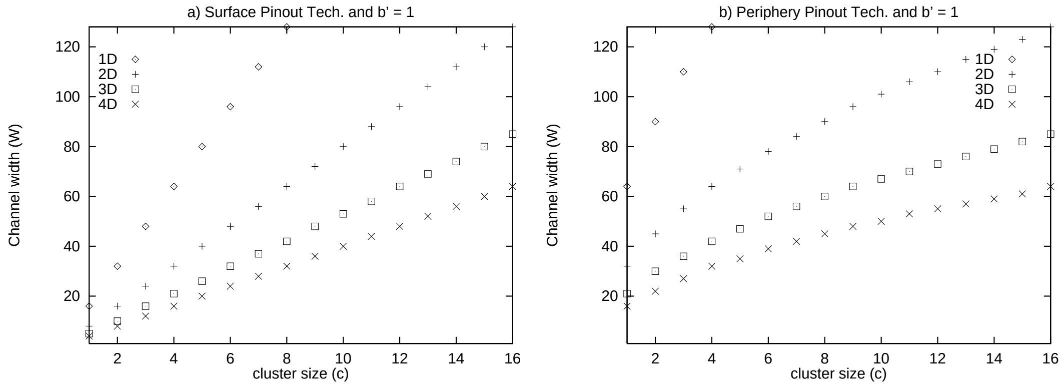

Figure 7: Offered channel width as a function of cluster size and inter-cluster dimensionalities. Board size is not fixed and chosen to hold one cluster with c processors ($b' = 1$). Two different pinout technologies are considered a) surface pinout technology with $p_s = 64$ and b) periphery pinout technology with $p_p = 256$.

Let us consider the impact of board size on channel width W. Under periphery pinout technology, $P_b = p_p\sqrt{b}$. Using $b' = b/c$, we can simplify Eqn. 8 as $W = \frac{p_p c^{(n-1)/n}}{4b^{(n-2)/(2n)}n}$. Thus, for a given cluster size and inter-cluster dimensionality $n > 2$, an increase in board size b leads to a fall in channel width. The channel width in a 2D inter-cluster network remains constant while that in an 1D network rises sharply. This is shown in Fig. 8(b) for $p_p = 256$ and a cluster size of 2. Under surface pinout technology, with $P_b = p_s b$, a similar simplification leads to $W = \frac{p_s c^{(n-1)/n} b^{1/n}}{4n}$. Thus, for any given cluster size and inter-cluster dimensionality, an increase in board size b leads to a rise in channel width. However, this rise is not very significant for higher dimensions. Such trends are shown in Fig. 8(a) for $p_s = 64$ and a cluster size of 2. This leads to:

Observation 2 *Under periphery pinout technology, keeping cluster size c fixed and dimensionality of inter-cluster network $n > 2$, an increase in board size results in channel width to fall. However, under surface pinout technology, for all inter-cluster dimensionalities it leads to a rise in channel width.*

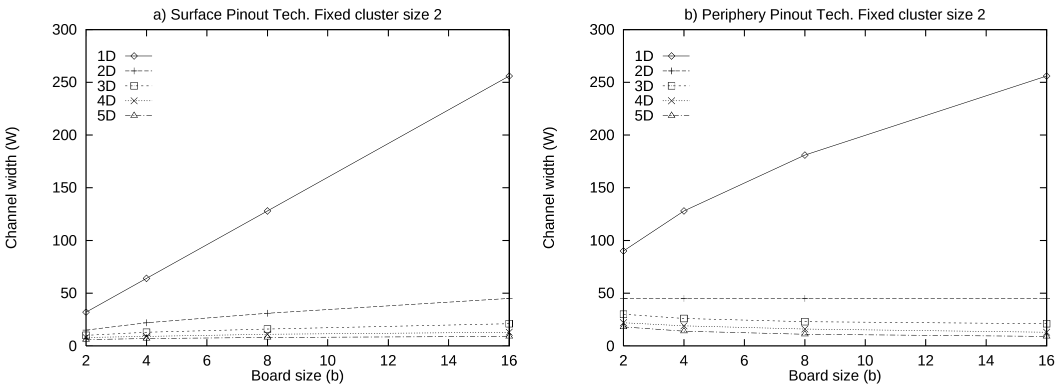

Figure 8: Channel width as a function of board size for different inter-cluster dimensionalities under a) surface pinout technology with $p_s = 64$ and b) periphery pinout technology with $p_p = 256$. Cluster size is kept fixed at $c = 2$.

5.2 Supporting a Fixed Inter-Cluster Channel Width

Most of the prior studies on system design, while proposing guidelines under different constraints like constant bisection bandwidth [11], did not impose any restrictions on the values that the channel width can take while satisfying other constraints. Supporting an arbitrary channel width is not easy because it has impact on router design. Agrawal realized the importance of this constraint and analyzed system designs under fixed channel width [2]. For a simpler interface design, the width of *data* lines in a channel is expected to maintain an integral relationship with that of processor and memory which are typically in multiples of a byte.

In Eqn. 8, we showed that $W(W_p)$ gets determined once other system parameters like n, b, and c get specified. Given a channel width technology with W' (W' = 24, 40, 72...), an obvious design objective while selecting values for n, b, and c is to ensure that the offered channel width (W) is equal to the supportable channel width. Together with Eqn. 8 this leads us to the following set of observations on the cluster size that can be placed on a board while other parameters like inter-cluster dimensionality, pinout density, and channel width are varied:

Observation 3 .

- *For a fixed channel width (W'), board size (b), and a pinout technology, an increase in inter-cluster network dimensionality (n) leads to an increase in the cluster size (c), resulting in fewer clusters on the same board area.*
- *For a fixed channel width (W'), board size (b), and inter-cluster dimensionality (n), an increase in pinout density (p_p or p_s, and hence P_b) leads to a fall in the cluster size (c), resulting in more clusters on the same board area.*
- *Increasing the channel width (W') leads to larger cluster size (c) while other parameters are kept constant. This leads to fewer clusters on a given board area.*

While maintaining a fixed channel width (W'), it may also be desired to maintain a given fixed number of clusters ($b' = b/c$) on each board. Under such a condition, supporting larger clusters also implies using larger boards to maintain a fixed b'. Let us consider the effect of increasing board size (or equivalently cluster size) on the inter-cluster dimensionality in Eqn. 8 while maintaining a fixed W' and b'. Larger board area allows more pinout per board thus leading to an increase in P_b. This implies that n should also increase to maintain the same channel width. This can be stated as the following observation:

Observation 4 *For fixed number of clusters on a board (b') and supportable channel width (W'), an increase in board (cluster) size leads to an increase in the dimensionality of the inter-cluster network.*

Let us consider the impact of pinout technology on supporting a fixed channel width. Under periphery pinout technology, as discussed earlier, we need to satisfy $W' = \frac{p_p c^{(n-1)/n}}{4 b^{(n-2)/(2n)} n}$. Thus, for a given inter-cluster dimensionality $n > 2$ and pinout density (p_p), supporting larger cluster size requires an increase in the size of the boards used. For $n = 2$, cluster size can not be varied because changing board sizes has no impact. Similarly, for $n = 1$, board size can not be varied because changing cluster size has no impact. Under surface pinout technology, we need to satisfy $W' = \frac{p_s c^{(n-1)/n} b^{1/n}}{4n}$. Thus, for a given $n > 1$ and pinout density (p_s) , supporting larger cluster sizes requires a fall in the size of the boards used. For $n = 1$, board size can not be varied because changing cluster size has no impact. The above interplay can be summarized as the following observation:

Observation 5 *In supporting a fixed channel width (W') in an inter-cluster network with a given dimensionality (n), for periphery pinout technology with $n > 2$, supporting a larger cluster size requires an increase in board size. In comparison, under surface pinout technology with $n > 1$, supporting a larger cluster size requires a fall in board size.*

5.3 Realizing Designs with Under Utilization

The expression in Eqn. 8 yields a channel width W assuming full utilization of board area and pinout resources. However, while building a real machine it is unlikely that both board area and pinout resources get fully utilized. To make the study more realistic and interesting, we extend the expression in Eqn. 8 to allow for under-utilization of board and pinout capacities as discussed in sec. 4.3. Now Eqn. 8 can be rewritten as,

$$W' = \frac{P_b u_p}{4 n u_b (b/c)^{(n-1)/n}} = W \frac{u_p}{u_b} \tag{9}$$

where $P_b u_p$ is the utilized pincount from board, $u_b b'$ is the actual number of clusters placed on a board (out of the maximum $b' = b/c$), and W is the channel width value obtained assuming full-utilization of resources. We assume a reasonable bound on both under-utilizations such that $u_p, u_b \geq u_{min}$ for some u_{min} closer to 1. Thus, observations made earlier assuming full-utilization ($u_{min} = 1$) continue to hold. For illustrative purposes, in this paper, we choose $u_{min} = 0.9$. Given a pinout density and a channel width technology, we formulate the following search problem to determine (n, b, c) tuples which satisfy the following inequalities.

Allowing *only* under-utilization of pinout, we have $0.9 \leq u_p \leq 1$ and $u_b = 1$. This leads to:

$$W = W'(u_b/u_p) \Rightarrow W \leq 1.1W' \tag{10}$$

Allowing *only* under-utilization of board area, we have $0.9 \leq u_b \leq 1$ and $u_p = 1$. This leads to:

$$W = W'(u_b/u_p) \Rightarrow W \geq 0.9W' \tag{11}$$

Figure 9 shows the plots in Fig. 7 with such under-utilization tolerance lines, which provide flexibility of 10% around the channel widths of 24, 40, and 72. The configurations which fall within these tolerance boundaries are *valid* or *design-feasible* configurations. We summarize the solutions to this search problem in Table 3 for surface pinout technology of $p_s = 64, 256$ and in Table 4 for periphery pinout technology of p_p =256, 512. In both tables three different supportable channel width technologies corresponding to W = 24, 40, and 72 and different allowable number of clusters per board are considered. In the examples shown in Fig. 9 and Tables 3 and 4, we ensure that the maximum board size ($b = b'c$) does not exceed 16 which was assumed in sec. 4.2.1 as the largest board size. Blank entries in the tables indicate that there is no valid configuration for that inter-cluster dimensionality, pinout technology, channel width, and board area. Note that the majority of the valid configurations fall under $b' = 1$ or 2. Solutions become fewer for larger values of b'. In fact for the technological parameters chosen, no configurations are possible for $b' = 16$ (the columns corresponding to $b' = 16$ have therefore not been considered in these tables). This suggests that

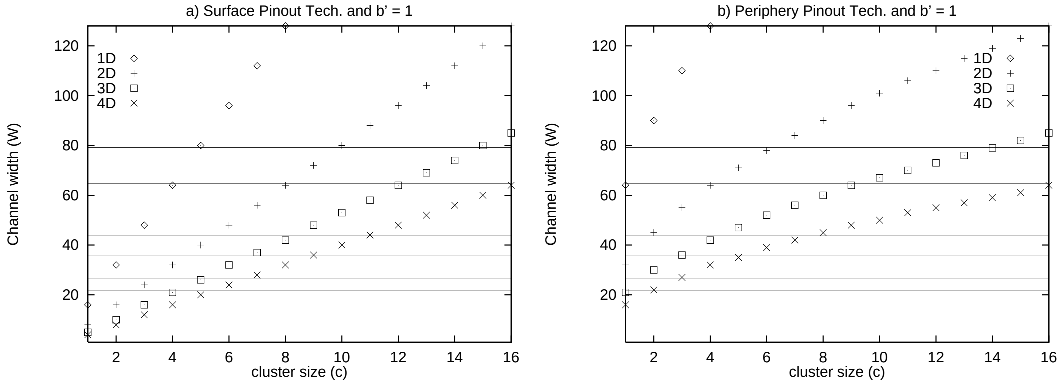

Figure 9: Valid configurations offering channel width (W') closer to 24, 40, or 72 with a reasonable (10%) under-utilization of board or pinout resources. Two different pinout technologies are considered: a) surface with $p_s = 64$ and b) periphery with $p_p = 256$.

most current and near future systems will have 1 or 2 clusters on each board. With increase in pinout densities, under both surface and periphery technologies, it may also be observed that the number of valid configurations with $b' = 4, 8$ increase. This suggests that future systems with higher pinout densities will be able to better support 4 to 8 clusters on each board.

From Table 3 note that for any given value of $p_s(64)$, $W(24)$, and $b'(1)$ as cluster size c is increased from 3 to 13 the required dimensionality of the inter-cluster network also rises from 2 to 8. This confirms Obs. 4. Next consider the board sizes required for configurations under surface pinout technology corresponding to $p_s = 64$, $W = 24$, $n = 2$ with two different cluster sizes $c = 1$ and 3. The configuration with the smaller cluster size ($c = 1$) and $b' = 8$ requires boards of size $b = 8$, while the one with larger cluster size ($c = 3$) and $b' = 1$ requires a smaller $b = 3$. Under periphery pinout technology a different trend is noted. In Table 4 the board sizes required for configurations corresponding to $p_s = 512$, $W = 24$, $n = 4$ with two different cluster sizes $c = 1$ and 2 are $b = 2$ and 16, respectively. Therefor as cluster size increases so does board size. This confirms the observations made in Obs. 5.

5.4 Offered Inter-Cluster Bisection

Now let us consider the impact of packaging on offered inter-cluster bisection. Given the inter-cluster network as a (k_1 x k_2 x... x k_n) mesh/torus and a fixed channel width, the size of the inter-cluster bisection can be computed in the following manner. A (k_1 x k_2 x... x k_n) mesh/torus can have various bisections which divide this network into two halves. For example, in a 3D torus with x, y, and z dimensions, we can have three possible bisections: one orthogonal to x dimension along yz plane, and so on. We are interested in the size of the smallest bisection in the system

Table 3: Valid Ranges of Cluster Size (c) from Packaging Constraints. Surface pinout technology with $p_s =$ 64 and 256 is assumed. Supported channel widths are W = 24, 40, and 72. Inter-cluster dimensionality (n) varies from 1 - 8. Number of clusters/board (b') is varied from 1 - 8. Maximum board size ($b = b'c$) is assumed to be ≤ 16.

	$p_s = 64$					$p_s = 256$				
	n	c				n	c			
		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$
W=24	2	3			1	2				
	3		4	3		3		1		
	4	6	5			4				1
	5	7 - 8	7			5	2			
	6	9	8			6		2	2	
	7	10 - 11				7				2
	8	11 - 13				8	3	3		
W=40	2	5	4			2		1		
	3	7 - 8	6			3	2			1
	4	9 - 11	8			4		2	2	
	5	12 - 13				5	3	3		2
	6	14 - 16				6	4		3	
	7	16				7	4	4	4	
	8					8	5	5	4	
W=72	2	9	7			2				
	3	13 - 14				3		3		
	4					4		4		
	5					5	6	5		
	6					6	7	6		
	7					7	8	7 - 8		
	8					8	9	8		

because it maximally constrains the performance of the system under random traffic. Clearly, the smallest bisection has to be orthogonal to the dimension having the largest radix, given by $k_{max} = \max_{i=1..n}(k_i)$. The number of nodes in such a bisection is given by N'/k_{max}. Assume each of these nodes is connected to another across this bisection using bidirectional channels of width W. This leads to the offered bisection size, β_u^s, as $\frac{2N'W}{k_{max}}$ wires. The presence of wrap-around channels in torus doubles this number. This leads to:

$$\begin{aligned}\beta_u^s &= \frac{4N'W}{k_{max}} = \frac{4N'W}{(N/c)^{1/n}} = 4(N/c)^{(n-1)/n}W \\ &= 4(N/c)^{(n-1)/n}\frac{P_b}{4nb'^{(n-1)/n}} \\ &= \frac{{N_{boards}}^{(n-1)/n}P_b}{n}\end{aligned} \tag{12}$$

which is simplified by assuming the inter-cluster network is regular i.e. ($k_{max} = (N/c)^{1/n}$) and using Eqn. 8 to replace W and $N_{boards} = N/b$.

Table 4: Valid Ranges of Cluster Size (c) from Packaging Constraints. Periphery pinout technology with $p_p = 256$ and 512 is assumed. Supported channel widths are $W = 24$, 40, and 72. Inter-cluster dimensionality (n) varies from 1 - 8. Number of clusters/board (b') is varied from 1 - 8. Maximum board size ($b = b'c$) is assumed to be ≤ 16.

	$p_p = 256$					$p_p = 512$				
	n	c				n	c			
		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$
W=24	3		2	2		3				
	4	2	3 - 4			4		1		2
	5	3 - 4	5 - 6			5	1		2	
	6	5 - 6	7 - 8			6		2	3 - 4	
	7	6 - 8				7	2	3		
	8	8 - 10				8	2	4		
W=40	2		2			2				
	3	3 - 4	4 - 5			3	1	1		2
	4	6 - 7	8			4		2	3 - 4	
	5	8 - 11				5	2	4		
	6	12 - 16				6	3 - 4	5 - 7		
	7	16				7	4 - 5	7 - 8		
	8					8	6 - 7			
W=72	2	5 - 6	5 - 6			2				1
	3	10 - 13				3	3	4		
	4					4	5 - 6	7 - 8		
	5					5	7 - 9			
	6					6	10 - 13			
	7					7	13 - 16			

5.4.1 Impact of board size

Figure 10 shows the plots of offered bisection bandwidth B in the inter-cluster network with respect to different pinout technologies as board size is increased. Under periphery pinout technology with $P_b = p_p\sqrt{b}$ and $N_{boards} = N/b$, Eqn. 12 reduces to $\beta_u^s = \frac{(N/b)^{(n-1)/n} p_p \sqrt{b}}{n} = \frac{N^{(n-1)/n} p_p}{b^{(n-2)/(2n)} n}$. Thus, for a given system size (N) and inter-cluster network dimensionality (n), the size of the bisection falls with increase in board size for $n > 2$. This can be explained by the fact that the total number of board pincount reduces as board sizes are increased. For $n = 2$, the value of β_u^s remains fixed and for $n = 1$ the value rises slowly. The key point is that under this pinout technology, it is ideal to work with smaller boards as they offer higher bisection sizes and hence performance. Under surface pinout technology with $P_b = p_s b$, Eqn. 12 reduces to $B = \frac{(N/b)^{(n-1)/n} p_s b}{n} = \frac{N^{(n-1)/n} p_s b^{1/n}}{n}$. Thus, we see here that for a given system size (N) and inter-cluster network dimensionality (n), the size of the bisection increases for n. These observations can be summarized as,

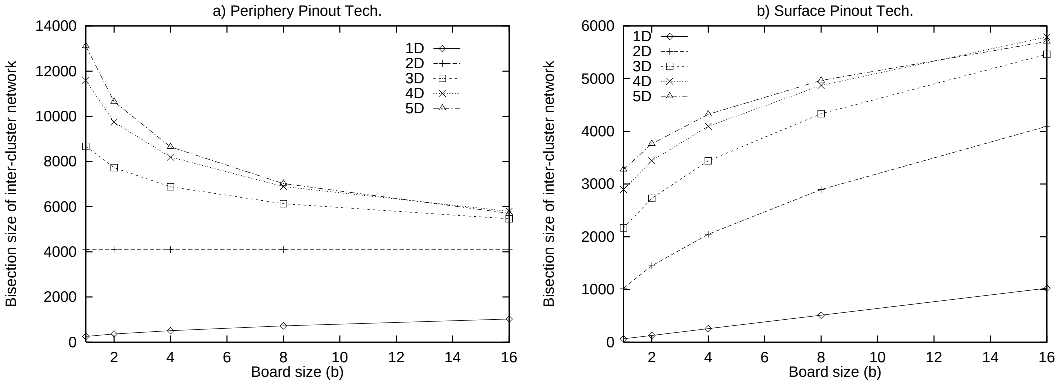

Figure 10: Offered inter-cluster bisection β_u^s in a system of 1024 processors as a function of board size and different inter-cluster dimensionalities under two different pinout technologies a) periphery pinout with $p_p = 256$ and b) surface pinout with $p_s = 64$.

Observation 6 *Under periphery pinout technology, for a given system size (N), keeping the dimensionality of inter-cluster network (n) fixed at a value greater than 2, an increase in board size (b) leads to a fall in the inter-cluster bisection size. However, under surface pinout technology, as board size (b) is increased, bisection size increases for all inter-cluster dimensionalities.*

Next, let us define V_w as the volume (number) of inter-board wires required to interconnect a desired inter-cluster configuration. This parameter V_w can be derived as $N_{boards}P_b/2$, where $N_{boards} = N/b$, P_b is the pinout from each board, and each wire connects two pins. Under periphery pinout technology, V_w can be simplified as $\frac{Np_p}{2\sqrt{b}}$ while under surface pinout technology it simplifies to $\frac{Np_s}{2}$. From these two expressions we conclude,

Observation 7 *For a given system size, the volume of inter-board wires falls with an increase in board size under periphery pinout technology, while under surface pinout technology it remains fixed.*

5.4.2 Indirect impact of cluster size

From Eqn. 12 note that $\beta_u^s = 4(N/c)^{(n-1)/n}W$. Thus, keeping channel width W fixed at W', an increase in the cluster size c (maintaining $W = W'$ may require board size to be varied also) causes the offered bisection to fall. Fig. 11 shows this for a system size $N = 1024$ processors and a given channel width $W' = 24$, and different inter-cluster dimensionality ($n = 1 - 4$).

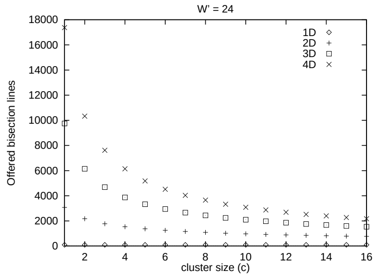

Figure 11: Offered inter-cluster bisection β_u^s as a function of cluster size and inter-cluster dimensionalities while maintaining a channel width $W' = 24$ for a 1024 processor system. Note that for a given pinout technology, the board sizes are not fixed and changed with cluster size to maintain $W' = 24$.

6 Deriving Optimal System Configurations

6.1 Demand vs Supply: An Optimization Problem

For a given system size, our goal is to derive the optimal configuration. This configuration should satisfy packaging and bisection demand constraints and offer best performance. We formulate this as an optimization problem of selecting the *best* configuration from amongst the various configurations. The problem is solved in three phases. In the first phase we derive the set of design-feasible configurations which adhere to the packaging constraints as discussed earlier,

- Maximum board size constraint ($b \leq 16$)
- Pinout constraint from board ($P_b = p_s b$, under surface pinout technology or $P_b = p_p\sqrt{b}$, under periphery pinout technology)
- Channel width constraint: $(W = \min(W_p, W_b)) \wedge (W \in \{24, 40, 72\})$

System configurations which are possible to be synthesized under the above constraints are sorted out in this process. This was studied in sec. 5 and *valid* or design-feasible configurations derived under various technological parameters in Table 3 and 4. For each design-feasible configuration we then compute the offered bisection in the inter-cluster (β_u^s) and intra-cluster networks (β_l^s). The set of design feasible configurations and their offered characteristics are referred by us as the supply side. The demand side refers to the demanded bisection sizes β_u^d and β_l^d in the inter-cluster and intra-cluster networks of such design-feasible configurations, derived earlier in sec. 3.3 on the basis of processor and interconnect speeds. The second phase of the optimization problem sorts out the design-feasible configurations which meet the demanded bisections, i.e.

- Inter-cluster bisection constraint: $\beta_u^s \geq \beta_u^d$

- Intra-cluster bisection constraint: $\beta_l^s \geq \beta_l^d$

The *Inter-cluster Bisection Constraint* states that the offered inter-cluster bisection from packaging constraints should be greater than the demanded inter-cluster bisection from processor and interconnect speed considerations. Using Eqn. 3 and Eqn. 12 in the above inequality we obtain the following:

$$4(N/c)^{(n-1)/n} W \geq \frac{\lambda N}{2} \tag{13}$$

bisection β_u^s with respect to cluster size c. We augment that plot by a horizontal line as shown in Fig. 12. The horizontal line corresponds to the demanded bisection size β_u^d of $N\lambda/2$ as derived in Sec. 3.3. A representative value of $\lambda = 6.0$ is assumed here yielding $\beta_u^d = 3072$ for a system with $N = 1024$ processors. It is very clear that only those configurations above the β_u^d line meet the bisection demand. In Table 5 we summarize the range of cluster sizes, which meet the demanded bisection bandwidth under $\lambda = 6.0$ for each combination of system size ($N = 1024$, 2048, and 8192), channel width technology ($W = 24$, 40, and 72), and inter-cluster dimensionality ($n = 1$ to 8).

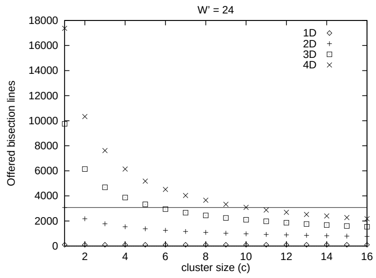

Figure 12: Offered inter-cluster bisection β_u^s as a function of cluster size and inter-cluster dimensionalities while maintaining a channel width $W' = 24$. The demanded bisection bandwidth corresponding to $\lambda = 6.0$ is shown as a horizontal line.

The *intra-cluster bisection constraint* states that the offered intra-cluster bisection from packaging constraints should be greater than the demanded intra-cluster bisection from processor and interconnect speed considerations. We have not considered any limitations or restrictions on the size and scalability of the intra-cluster network in this work and assume the intra-cluster bisection demand to be satisfied. However, in bus-based clusters the size of clusters do get limited by the bus-bandwidth. Contemporary bus-based clusters support up to 4 processors [13, 16]. Future bus technology is expected to support larger sized clusters. In current technology large clusters with higher intra-cluster bisection can be supported by using MIN-based or direct network-based clusters.

The second phase of the optimization leads us to the set of *good* configurations. The final optimization step is to select the *best* configuration from the good configurations. Next, we take up an example to show how to derive the set of good configurations and the best configuration under a given set of current technological parameters.

Table 5: Range of Cluster Sizes Satisfying Bisection Demands for Different System Sizes (N = 1024, 2048, and 8192), Channel Widths (W =24, 40, and 72), and Inter-cluster Dimensionalities (n = $1-8$). The Maximum Cluster Size is Restricted to 16.

		Cluster Sizes		
		$N = 1024$	$N = 2048$	$N = 8192$
$W = 24$	n= 1			
	n= 2	1		
	n= 3	1 - 5	1 - 3	1
	n= 4	1 - 10	1 - 8	1 - 5
	n= 5	1 - 13	1 - 11	1 - 8
	n= 6	1 - 16	1 - 13	1 - 10
	n= 7	1 - 16	1 - 15	1 - 12
	n= 8	1 - 16	1 - 16	1 - 14
$W = 40$	n= 1			
	n= 2	1 - 2	1	
	n= 3	1 - 12	1 - 8	1 - 4
	n= 4	1 - 16	1 - 15	1 - 9
	n= 5	1 - 16	1 - 16	1 - 15
	n= 6	1 - 16	1 - 16	1 - 16
	n= 7	1 - 16	1 - 16	1 - 16
	n= 8	1 - 16	1 - 16	1 - 16
$W = 72$	n= 1			
	n= 2	1 - 8	1 - 4	1
	n= 3	1 - 16	1 - 16	1 - 10
	n= 4	1 - 16	1 - 16	1 - 16
	n= 5	1 - 16	1 - 16	1 - 16
	n= 6	1 - 16	1 - 16	1 - 16
	n= 7	1 - 16	1 - 16	1 - 16
	n= 8	1 - 16	1 - 16	1 - 16

6.2 Design Choices under Various Technologies

6.2.1 Deciding on Good Configurations

Tables 3 and 4 provided a summary of the design-feasible configurations which satisfy the packaging constraints (including the channel width constraint) under surface and periphery pinout technologies respectively. Table 5 provided a range of configurations which satisfy the inter-cluster bisection constraint. The intersection of these two tables identify good configurations: Tables 6 and 7 show the offered good configurations for a system size of N = 1024 processors under surface and periphery pinout technologies, respectively. Similar tables can be generated for other system sizes of $N = 2048$ and 8192 processors.

Table 6: Good Ranges of Cluster Size (c) from Packaging and Bisection Demand Constraints. Surface pinout technology with $p_s = 64$ and 256 is assumed. Supported channel widths are $W = 24$, 40, and 72. Inter-cluster dimensionality (n) varies from 1 - 8. Number of clusters/board (b') is varied from 1 - 8. Maximum board size ($b = b'c$) is assumed to be ≤ 16.

	$p_s = 64$					$p_s = 256$				
	n	c				n	c			
		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$
W=24	2				1	2				
	3		4	3		3		1		
	4	6	5			4				1
	5	7 - 8	7			5	2			
	6	9	8			6		2	2	
	7	10 - 11				7				2
	8	11 - 13				8	3	3		
W=40	2					2		1		
	3	7 - 8	6			3	2			1
	4	9 - 11	8			4		2	2	
	5	12 - 13				5	3	3		2
	6	14 - 16				6	4		3	
	7	16				7	4	4	4	
	8					8	5	5	4	
W=72	2		7			2				
	3	13 - 14				3		3		
	4					4		4		
	5					5	6	5		
	6					6	7	6		
	7					7	8	7 - 8		
	8					8	9	8		

Note that most design-feasible configurations in Tables 3 and 4 meet the bisection demands corresponding to a system size of $N = 1024$ processors and $\lambda = 6$ and qualify as good configurations in Tables 6 and 7. However, for larger system sizes with greater processor demand on network it can be shown that fewer design-feasible configurations would meet the higher bisection demands and qualify as good configurations.

Table 7: Good Ranges of Cluster Size (c) from Packaging and Bisection Demand Constraints. Periphery pinout technology with $p_p = 256$ and 512 is assumed. Supported channel widths are $W = 24$, 40, and 72. Inter-cluster dimensionality (n) varies from 1 - 8. Number of clusters/board (b') is varied from 1 - 8. Maximum board size ($b = b'c$) is assumed to be ≤ 16.

	$p_p = 256$					$p_p = 512$				
	n	c				n	c			
		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$		$b' = 1$	$b' = 2$	$b' = 4$	$b' = 8$
	3		2	2		3				
	4	2	3 - 4			4		1		2
W=24	5	3 - 4	5 - 6			5	1		2	
	6	5 - 6	7 - 8			6		2	3 - 4	
	7	6 - 8				7	2	3		
	8	8 - 10				8	2	4		
	2		2			2				
	3	3 - 4	4 - 5			3	1	1		2
	4	6 - 7	8			4		2	3 - 4	
W=40	5	8 - 11				5	2	4		
	6	12 - 16				6	3 - 4	5 - 7		
	7					7	4 - 5	7 - 8		
	8					8	6 - 7			
	2	5 - 6	5 - 6			2				1
	3	10 - 13				3	3	4		
	4					4	5 - 6	7 - 8		
W=72	5					5	7 - 9			
	6					6	10 - 13			
	7					7	13 - 16			

6.2.2 Deriving the Best Configuration

For any given set of technological parameters, the set of good configurations narrow down the design-choice space. Using any of these good configurations it is possible to actually fabricate a machine under packaging constraints and such a machine is guaranteed to offer more than or equal the desired bisection bandwidth. However, there are two other important considerations while selecting a configuration. It should offer low average message latency and it should have low cost. Thus, among the good configurations we are interested in the *best configuration* [23] which provides best performance at lowest cost.

From Tables 6 and 7, it is to be noted that under most combinations of pinout density and channel width technology, we are left with around 15-20 good choices from which we have to select the best one. This search, however, is not a very clear-cut process. Unknown and inestimable factors like exact link cost, connector cost, cluster technology, delay due to longer wires and other packaging costs make this a hard decision. The scalability of a configuration to larger sizes is also very important. In this section, we use a combined qualitative, quantitative, and simulation approach to derive the best configurations.

We first demonstrate the design choices under the current technologies of channel width ($W = 24$), surface pinout density ($p_s = 64$), and periphery pinout density ($p_p = 256$). Each entry in Table 6, corresponding to a good configuration, can be expanded to show detailed characteristics about the connector and wiring costs together with its potential for future scalability. Table 8 shows such detailed characteristics. For example, the entry corresponding to $n = 4$, $p_s = 64$, $W = 24$, $b' = 1$ and cluster size of 6 in Table 6 gets expanded to the first entry in Table 8. The desired number of clusters in this system with 1024 processors should be $1024/6 \approx 171$. However, the closest 4D internet topology is 4D:4x4x3x4 having 192 clusters. This leads to an actual system size of 192x6 = 1152 processors, which is larger than the desired size of 1024 by 12.5%. We allow such deviations in system size up to 15%. The good configurations which yield actual system sizes outside this range are dropped. The actual bisection size offered by this configuration is 4608 (obtained using Eqn. 5.4). The average distance to be traveled in the internet is 3.75, obtained as $\Sigma_{i=1}^{n} k_i/4$ [11]. The total number of internet channels can be obtained as $2n(N/c)$ yielding a value of 1536 for this configuration. This is a measure of the number of inter-cluster connectors required. Connector cost being quite high [23], a natural design guideline would be to opt for configurations which require less number of connectors. The board size of 6, used to place the clusters in this configuration, is the same as the cluster size. Therefore only one cluster is placed on each board. The number of inter-board wires, which is a measure of the volume of wiring, is derived as 32K (based on the discussion in Sec. 5.4). Other entries in Table 6 are similarly expanded.

Table 9 shows similar characteristics for entries in Table 7 under periphery pinout of $p_p = 256$. Both Tables 8 and 9 are split into three parts for ease of presentation: a) one cluster is put on one board implying $b' = 1$ i.e. the board size is chosen to fit the cluster size, b) two clusters are put on a board leading to $b' = 2$, and c) three or more clusters are put on a board leading to $b' \geq 3$.

The average distance to be traveled in the internet is a measure of the average latency a message incurs. While choosing the best configuration, a design objective should be to choose a configuration that offers low average message latency. Since messages can travel across clusters, this average latency measure should include both the inter-cluster network delay as well as the delay incurred inside the cluster. However, not having restricted our study to a particular intra-cluster topology, we primarily emphasize on the inter-cluster delay.

Let us consider the *scalability* aspect in choosing a good configuration. While building a system with $1K$ processors, once a good configuration (say n =4D, $c = 5$, $b = 10$, and $W = 24$) is chosen to build it, these parameters get fixed. Now assume that it is required to scale this system to $2K$ processors. To maintain a homogeneous system the addition of new processors to the system should be done while keeping the parameters n, b, c, and W fixed to their prior values. Otherwise, the system requires complete redesign and refabrication. Let $(n, b, c, W)_{1K}$ denote the set of all n, b, c, and W values that represent good configurations to build a system of $N = 1K$ processors. Similarly, let $(n, b, c, W)_{2K}$ and $(n, b, c, W)_{8K}$ denote such sets for system sizes of $N = 2K$ and $8K$ processors, respectively. The intersection set $(n, b, c, W)_{1K} \bigcap (n, b, c, W)_{2K}$ therefore denotes good configurations under both $N = 1K$ and $2K$ processors. This implies that these configurations used to build a system with $1K$ processors can be scaled up to $2K$ processors. Similarly, the intersection set $(n, b, c, W)_{1K} \bigcap (n, b, c, W)_{2K} \bigcap (n, b, c, W)_{8K}$ represents configurations possible for systems with $1K$, $2K$, and $8K$ processors. Therefore, such configurations can be scaled up to $8K$ processors. The last column in Tables 8 and 9 shows the maximum system size to which the configuration being considered can be scaled. Note that not all good configurations for $1K$ processors scale up to $2K$ or $8K$ processors.

Choosing the *best* configuration

A) Surface pinout O(A) technology: In this technology, irrespective of internet configuration, the total volume of inter-board wires (total number of pins) is fixed. Larger boards offer greater bisection bandwidth under this technology. Thus a reasonable approach is to go for larger board sizes for any given internet dimensionality. To decide among the good configurations we should also keep in mind factors like minimal latency, connector costs, cost of inter-board wires, and extent of future scalability. Let us consider the good configurations under $p_s = 64$ and $W = 24$ technology in Table 8. We chose the 4D:4x4x4x3 with $c = 5$ as the best topology as it offers reasonably high bisection size (recall that for a system of size 1024 processors demanded bisection size is 3072). It also offers low average latency, with low connector cost, while being scalable to around 8K processors.

B) Under Peripheral Pinout $O(\sqrt{A})$ technology: Under this technology, the total volume of inter-board wires in the system falls with increasing board size. This decreasing system volume could translate to a fall in packaging costs. Consider the good configurations under $p_p = 256$ and $W = 24$ technology in Table 8. Note the 3D configurations are scalable up to only 2K processors. For scalability beyond 2K we need to choose amongst the higher dimensional systems. The 4D with $c = 4$ and 5D with $c = 6$ under $b' = 2$ and 4D with $c = 4$ under $b' \geq 3$ provide such scalability while offering reasonably high bisection bandwidth at low connector and inter-board wire volume. Amongst these three configurations the 4D under $b' \geq 3$ is clearly better than the 4D under $b' = 2$ as it offers similar characteristics with lower inter-board volume of wires. We therefore consider the two configurations: 5D with $c = 6$ and $b' = 2$ and 4D with $c = 4$ and $b' = 3$. Simulation modeling was used to decide the best configuration by comparing the offered throughput versus average message latency. For this experiment bus-based clusters were assumed. We observed as shown in Fig. 13 that at high system loads the cluster-bus became a greater bottleneck in the 5D system with larger clusters. Message latencies in both systems at lower latencies were comparable. These observations lead us to conclude that the 4D:4x4x4x4 with $c = 4$ will be the best system configuration.

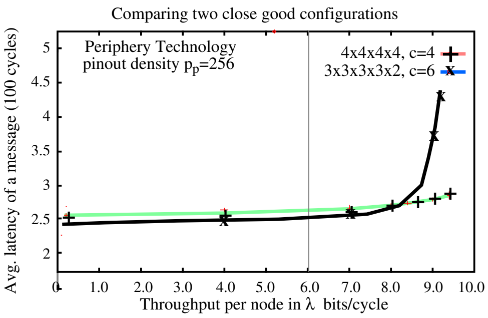

Figure 13: Comparing the performance of two very close good choices under periphery pinout technology ($p_p = 256$) and channel width ($W = 24 \equiv 16$ data lines). Wormhole routing with message length = 200 bits and message start up time of 1μ sec (100 cycles) were used.

6.2.3 Applying the Framework to Contemporary Design Trends

From Tables 8 and 9 the following interesting comments can be made.

Cray T3D example

Employing the technology combination of $p_p = 256$ and $W = 24$ the Cray T3D uses boards of size 4 containing 2 clusters (nodes) each of size 2 interconnected through a 3D inter-cluster network [10]. Our framework derives this configuration as the first entry in Table 9 under $b' = 2$. This may be treated as a validation of our framework in predicting real system design. Also note the first entry in Table 9 under $b' \geq 3$. This predicts that if in a similar system if boards of size 14 are used instead of 4 then by placing 7 clusters on each board the volume of inter-board wiring in a Cray T3D system could be reduced from 64K to 34K i.e almost halved leading to a much less expensive system.

Flat systems are not always possible

Our framework is general enough to allow flat configurations ($c = 1$). However, it is interesting to note that while designing system under some technology combination it may happen that the flat configuration is not possible at all because it is not a good configuration. This can be seen in Table 9 where none of the good configurations offer $c = 1$.

Higher inter-cluster dimensionality may be required

Table 8: Comparing Various Performance and Cost Characteristics of Good Configurations to Design a System with $N \approx 1024$ Processors Under Current Surface Pinout Technology ($p_s = 64$) and Channel Width ($W = 24$). Maximum Cluster size and Board Size are limited to ≤ 16 Processors.

Internet topology	cluster size c	Actual number of processors	Actual bisection size	Average distance traveled in internet	Number of internet channels (connectors)	Largest board size usable	Volume (number) of inter-board wires	Number of processors up to which scalable
Number of clusters on a board $b' = 1$								
4D: 4x4x3x4	6	1152	4608	3.75	1536	6	32K	2K
5D: 3x3x3x2x3	7	1134	5184	3.50	1620	7	32K	8K
Number of clusters on a board $b' = 2$								
3D: 6x7x6	4	1008	3456	4.75	1512	8	32K	1K
4D: 4x4x4x3	5	960	4608	3.75	1536	10	32K	8K
5D: 3x3x3x2x3	7	1134	5184	3.50	1620	14	32K	8K
6D: 2x2x2x3x2x3	8	1152	4608	3.50	1728	16	32K	8K
Number of clusters on a board $b' \geq 3$								
2D: 32x32	1	1024	3072	16.00	4096	10	32K	1K
3D: 7x7x7	3	1029	4704	5.25	2058	12	32K	2K

Table 9: Comparing Various Performance and Cost Characteristics of Good Configurations to Design a System with $N \approx 1024$ Processors Under Current Periphery Pinout Technology ($p_p = 256$) and Channel Width ($W = 24$). Maximum Cluster size and Board Size are limited to ≤ 16 Processors.

Internet topology	cluster size c	Actual number of processors	Actual bisection size	Average distance traveled in internet	Number of internet channels (connectors)	Largest board size usable	Volume (number) of inter-board wires	Number of processors up to which scalable
Number of clusters on a board $b' = 1$								
4D: 5x5x4x5	2	1000	9600	4.75	4000	2	90K	8K
5D: 3x3x3x4x3	3	972	7776	4.00	3240	3	73K	8K
5D: 3x3x3x3x3	4	972	7776	3.75	2430	4	64K	8K
6D: 2x3x2x3x2x3	5	1080	6912	3.75	2592	5	57K	8K
7D: 2x2x2x2x2x2x3	6	1152	6144	3.75	2688	6	52K	8K
7D: 2x2x2x2x2x2x2	8	1024	6144	3.50	1792	8	45K	8K
Number of clusters on a board $b' = 2$								
3D: 8x8x8	2	1024	6144	6.00	3072	4	64K	2K
4D: 4x4x5x4	3	960	6144	4.25	2560	6	52K	8K
4D: 4x4x4x4	4	1024	6144	4.00	2048	8	45K	8K
5D: 3x3x3x2x3	6	972	5184	3.50	1620	12	36K	8K
6D: 2x2x2x3x2x3	7	1008	4608	3.50	1728	14	34K	8K
6D: 2x2x2x3x2x3	8	1152	4608	3.50	1728	16	32K	8K
Number of clusters on a board $b' \geq 3$								
3D: 8x8x8	2	1024	6144	6.00	3072	14	34K	2K
4D: 4x4x4x4	4	1024	6144	4.00	2048	12	36K	8K

It has been recently demonstrated that that it is easier to build lower dimensional systems with 2D/3D interconnections [10, 11]. Higher dimensional networks require longer wires which may lead to higher wiring costs and packing effort. However, it is possible that system design under a given set of packaging constraints and technologies may not offer any good configuration with 2D/3D interconnection. For example, consider the design problem of a scalable system with $N = 1024$ processors under current technologies of $p_p = 256$ and $W = 24$. Let the packaging constraints be such that we can only place one cluster on each board ($b' = 1$). The possible good configurations are shown in Table 9 under $b' = 1$. Note that no good configuration is possible to build such a system with a 2D/3D interconnect. Under such constraints a system designer would therefore be forced to choose a higher dimensional system.

6.3 Summary of Best Scalable Configurations

We extended our previous analysis to predict best system configurations for other channel width and pinout density parameters reflecting near future technologies. Combinations of channel width technologies ($W = 24, 40$ and 72) were considered in conjunction with surface pinout technologies of ($p_s = 64$ and 256) and periphery pinout technologies of ($p_p = 256$ and 512). We first studied the problem of choosing the best configurations while restricting design to low-dimensional (2D/3D) inter-cluster networks with small cluster sizes ($c < 8$). Table 10 shows such best configurations to build a system with size of $N = 1024$ processors under various technologies. A best system configuration depicted using the shorthand (3D c-2) represents a (8-ary 3-cube cluster-2) system. Other entries can be similarly expanded to the k-ary n-cube cluster-c notation. Note that the availability of larger channel width under future technology has potential to go for larger cluster sizes under a given surface or peripheral pinout density. However, an increase in pinout density without growth in channel width will force systems with smaller cluster sizes. We also observe that a future periphery pinout density of $p_p = 512$ yields best system choices similar to those by a future surface pinout density of $p_s = 256$. This suggests that although theoretically surface pinout has potential to eject periphery pinout technology as shown in sec. 4.2.2, the state of the art is such that even if surface pinout technology were to improve 4 times while periphery pinout technology only grew twice, they would still be comparable.

We next considered the design of systems with 1024 processors which are scalable up to 8K processors under the same technologies. In this case we did not restrict ourselves to only low-dimensional networks with small cluster sizes. Table 11 shows the obtained best configurations. Note that interestingly all these configurations use 3D/4D/5D inter-cluster networks and cluster sizes from $2 - 10$. These are reasonable values and it is easy to comprehend that such systems will be available in the near future. The entry corresponding to a technology of $p_s = 64$ and $W = 72$ is empty because no system configuration with 1024 processors scales up to $8K$ processors.

Table 10: Summary of Best Configurations Under Various Combinations of Surface/Periphery Pinout and Channel Width Technologies to Build a System of $N = 1024$ Processors while Designing for Low Dimensional (2D/3D) Internets and Cluster Size ($c < 8$). The first value in each entry is the dimension of the internet, the second is the cluster size of the best configuration.

Surface Pinout Technology					
$p_s = 64$			$p_s = 256$		
$W = 24$	$W = 40$	$W = 72$	$W = 24$	$W = 40$	$W = 72$
3D c-4	3D c-9	2D c-7	3D c-1	3D c-2	3D c-3

Periphery Pinout Technology					
$p_p = 256$			$p_p = 512$		
$W = 24$	$W = 40$	$W = 72$	$W = 24$	$W = 40$	$W = 72$
3D c-2	3D c-5	2D c-6	3D c-1	3D c-2	3D c-4

Table 11: Summary of Best Configurations Under Various Combinations of Surface/Periphery Pinout and Channel Width Technologies to Build a System of $N = 1024$ Processors which is Scalable Up to 8K Processors. The first value in each entry is the dimension of the internet, the second is the cluster size of the best configuration.

Surface Pinout Technology					
$p_s = 64$			$p_s = 256$		
$W = 24$	$W = 40$	$W = 72$	$W = 24$	$W = 40$	$W = 72$
4D c-5	4D c-9		5D c-2	5D c-3	5D c-6

Periphery Pinout Technology					
$p_p = 256$			$p_p = 512$		
$W = 24$	$W = 40$	$W = 72$	$W = 24$	$W = 40$	$W = 72$
4D c-4	4D c-8	3D c-10	4D c-2	4D c-4	4D c-8

7 Conclusions and Future Work

In this paper we have presented a comprehensive supply-demand framework to design large multiprocessor systems by taking into account advancements in packaging, processor, and interconnection technologies. 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 while considering practical design aspects like maximum board area, maximum available pinout, fixed channel width, scalability, etc. The significance of it lies in its generality which has never been taken into consideration by earlier researchers. All proposed earlier works have centered around fixed topology, technology, or packaging constraint. This generalized framework can be applied to a wide variety of technological parameters and constraints for years to come.

The proposed framework has been validted with design trends of contemporary systems like Cray T3D. Using this framework, the following design guidelines have also been obtained for building large multiprocessor systems with k-ary n-cube cluster-c organizations:

1. Using a representative set of current technological parameters and low dimensional networks, a 1024-processor system can be optimally built as a 3D (6x6x7) cluster-4 system under surface pinout technology and as a 3D (8x8x8) cluster-2 system under periphery pinout technology.

2. Designing optimal systems with high scalability under various representative future technologies leads to 3D/4D/5D inter-cluster networks with cluster sizes between 2-10.

3. Availability of larger channel width under future technology has potential to go for larger cluster sizes under a given surface or peripheral pinout technology. However, an increase in pinout densities without growth in channel width will force systems with smaller cluster sizes.

4. Clustered systems deliver better performance than flat systems under most technologies. It is therefore expected that clustered systems would be a dominant trend in building future systems.

This research has emphasized on the inter-cluster network and cluster size while designing clustered systems. In the next phase of this research, we are investigating on the design of the intra-cluster interconnection and the cluster interface. The cluster interface may be augmented by adding extra functional capacities to implement various communication and synchronization operations more efficiently. It will be interesting to identify the nature and extent of these extra capacities to be provided as a function of the cluster size and their impact on overall system performance. This will make our understanding of clustered systems more complete.

References

[1] S. Abraham and K. Padmanabhan. Constraint based evaluation of multicomputer networks. In *Proc. of the Int. Conf. on Parallel Processing*, Aug 1990.

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

[3] A. Asthana, H. Jagdish, and B. Mathews. Impact of Advanced VLSI packaging on the design of a large parallel computer. In *Proc. of the Int. Conf. on Parallel Processing*, Aug 1989.

[4] 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*, pages 780–787, 1993.

[5] D. Basak and D. K. Panda. Designing Large Hierarchical Multiprocessor Systems under Pro cessor, Interconnection, and Packaging Advancements. In *Proc. of the Int'l Conference on Parallel Processing*, pages I:63–66, 1994.

[6] D. Basak and D. K. Panda. Designing Large Hierarchical Multiprocessor Systems under Processor, Interconnection, and Packaging Advancements. Technical Report OSU-CIS 1994/TR01, Department of Computer and Information Science, The Ohio State University, Jan 1994.

[7] D. Carlson. The Mesh with a Global Mesh: a Flexible, High-speed Organization for Parallel Computation. In *Proc. of the* 1^{st} *Int. Conf. on Supercomputer Systems.* IEEE Comp. Soc. Press, 1985.

[8] A. A. Chien and J. H. Kim. Planar Adaptive Routing: Low cost adaptive networks for multiprocessors. In *Proc.* 19^{th} *Ann.Int.Symp.on Comp.Arch*, pages 268–277, 1992.

[9] L. Choi and A. A. Chien. Integrating Networks and Memory Hierarchies in a Multicomputer Node Architecture. In *Proceedings of the International Parallel Processing Symposium*, pages 10–17, April 1994.

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

[11] W. J. Dally. Performance Analysis of k-ary n-cube Interconnection Networks. *IEEE Transactions on Computers*, 39(6), June 1990.

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

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

[14] 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.

[15] 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.

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

[17] A. W. Wilson Jr. Hierarchical Cache/Bus Architecture for Shared Memory Multiprocessors. In *Proc. of the Int. Symposium on Computer Architecture*, pages 244–252, 1987.

[18] A. Louri and H. Sung. 3D Optical Interconnects for High-Speed Interchip and Interboard Communications. *IEEE Computer*, pages 27–37, 1994.

[19] M. D. Noakes, D. A. Wallach, and W. J. Dally. The J-Machine Multicomputer: An Architectural Evaluation. In *Proc. of the Int. Symposium on Computer Architecture*, pages 224–235, 1993.

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

[21] D. K. Panda and D. Basak. Issues in Designing Scalable Systems with k-ary n-cube cluster-c Organization. In *Proc. of the First International Workshop on Parallel Processing, India*, 1994.

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

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

[24] M. T. Raghunath and A. Ranade. Designing interconnection networks for multi-level packaging. In *Proc. of the Supercomputing*, pages 772–781, 1993.

[25] E. Rothberg, J. P. Singh, and A. 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.

[26] A. A. Sawchuk, B. K. Jenkins, C. S. Raghavendra, and A. Varma. Optical Crossbar Networks. *IEEE Computer*, pages 50–62, 1987.

[27] Seraphim, Lasky, and Li Eds. *Principles of Electronic Packaging.* McGraw Hill, 1989.

[28] Tummala and Rymaszewski. *Microelectronics Packaging Handbook.* Van Nostrand Reinhold, 1989.