Designing Clustered Multiprocessor Systems under Packaging and Technological Advancements

Debashis Basak, *Member, IEEE* and Dhabaleswar K. Panda, *Member, IEEE*

Abstract—

Clustered or hierarchical interconnections demonstrate advantage in designing large scale multiprocessor systems. Earlier studies in literature have either focused on only flat interconnections or proposed hierarchical/clustered interconnections with limited packaging and demanded performance constraints. Large systems require several levels of packaging. Packaging technologies impose various physical constraints on bisection bandwidth and channel width of a system. Pinout technologies and capacity of packaging modules have been ignored in earlier studies, often leading to configurations that are not design-feasible. Similarly, the impact of processor and interconnect technologies on demanded performance has also not been considered. In this paper, we propose a new *supply-demand framework* for multiprocessor system design by considering packaging, processor, and interconnect technologies in an integrated manner. The elegance of this framework lies in its parameterized representation of different technologies. For a given set of technological parameters the framework derives the best configuration while considering practical design aspects like maximum board area, maximum available pinout, fixed channel width, and scalability. In order to build a scalable parallel system with a given number of processors, the framework explores the design space of flat k-ary n-cube topologies and their clustered variations (k-ary n-cube cluster-c) to derive design-feasible configurations with best system performance. The study identifies processor board area, supported channel width, board pinout density, and router pinout as critical parameters and analyzes their impact on deriving design-feasible and best configurations. For a wide range of parameters, it is shown that best configurations are achieved with cluster-based systems with up to 8 processors per cluster and 3D-5D inter-cluster interconnection.

Keywords— **Multiprocessor Systems, Scalable Systems, Clustered Architectures, Hierarchical Organization, k-ary n-cube Interconnection, Packaging Constraints, Parallel Architectures, Interconnection Networks.**

I. 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 [9], [11], [22]. Design guidelines need to take into account technological changes to yield system configurations delivering best performance. Several previous studies have considered packaging constraints [3] while selecting the best system configuration. These studies include Dally's [8] 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 k-ary n-cube networks considers three different constraints: constant bisection width, constant channel width, and constant pinout while considering node and wire delays. However, Ranade [18], [19] and Yew [12] have argued that neither Dally's VLSI model with limited bisection bandwidth nor the limited pinout model as proposed in [1], [2] is adequate while designing 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 [19] that while designing a large multiprocessor system with multiple levels of packaging, multi-level/hierarchical/clustered architecture can be an advantage. The architectural levels can be chosen to closely match the packaging hierarchy leading to better designs. A multi-level architecture also widens the design space, in terms of alternative configurations possible to build a system with a given number of processors. A variety of two-level hierarchical configurations have been proposed by researchers in the past to build scalable systems. Though a system design with multiple levels is a more generalized approach, it is commonly believed that two levels are sufficient to build parallel systems in the near future [19]. Moreover the design techniques which work for two-level hierarchies can be easily extended to accommodate more levels. Examples of previous work in this area include two-level systems based on hypercube and other network topologies[9], [15], [19], MINs and n-hop networks [19], and combination of bus and mesh/hypercube networks[11]. Though these designs provide alternative ways to build parallel systems, most of them do not take any packaging constraints into account. Thus, it is difficult to use these configurations to build realistic systems under varying technological and packaging constraints.

A typical hierarchy used in packaging a large system consists of multiple chips on a board and multiple such boards in a card-cage. A larger system may require multiple card-cages, multiple cabinets and so on. The modules at each level of this packaging hierarchy: chips, boards, card-cages etc. have their own characteristics in terms of maximum capacity, bisection size, available pinout, and channel width. For example, the maximum board size available may be limited to a size of 12"×12". The pinout from a board may be limited to 256-512 pins. The pinout from a router chip may be limited to a maximum of 250-300 pins. Factors like path-width inside routers and connector technology restrict channel widths from being arbitrarily large. For designing a simpler communication interface, the width of data lines in a channel is also expected to maintain an integral relationship with that of processor and memory which are typically in multiples of a byte. Such packaging limitations impose constraints on the design space. For example, a

The authors are with the Department of Computer and Information Science, The Ohio State University, Columbus, OH 43210-1277. E-mail:{basak,panda}@cis.ohio-state.edu.

configuration requiring a pinout of 500 from each router chip can not be supported under the router pinout limitation mentioned above. Similarly, a configuration with sixty-four 16 bit channels from a board requiring 1024 pins can not be supported under the above board pinout limitation. Based on such packaging constraints, only a small set of theoretically possible configurations are *package-able* or *design-feasible*. Only such configurations can translate into real machines. Thus, a system architect needs to first identify a set of design-feasible configurations, offered (*supplied*) by a given set of packaging technologies, before choosing on a configuration to build a system.

Besides design-feasibility, processor and application characteristics also lead to an expected *minimal demand* on performance from the system in terms of sustained average throughput and maximum allowable message latency. Sustaining a desired performance places a demand on the latency-throughput characteristics of a system configuration [4], [19]. Thus, an architect should identify such *demand* on performance and use it as a criterion to select *good* configurations satisfying the demand. However, the demanded performance is not a fixed parameter. It is a function of processor and interconnection speeds. For example, faster processors can compute quicker leading to faster injection of messages into the network. To sustain the faster computation rate, the demanded latency and sustained average throughput need to be supportable by the underlying communication network. For example, a given *good* design-feasible configuration for 100 MHz processor system need not remain good when processor technology changes to 200 MHz. However, by doubling the channel speed, the configuration can be made good for 200 MHz processors. Design results presented in previous studies have not taken into account such impact of processor and interconnect technology on the demanded performance. Thus, a design framework must consider packaging, processor, and interconnect technologies in an integrated manner to propose realistic design solutions.

In this paper, we propose such an integrated *supply-demand framework* for multiprocessor system design. The overall framework is summarized in Fig. 1. The basic objective of the framework is to design the best system configuration for a given number (N) of processors using a two-level architecture. The family of flat k-ary n-cube topologies and their clustered variations (k-ary n-cube cluster-c) are considered to derive scalable configurations. Theoretically a large number of alternate configurations are possible to build such a system. We consider several packaging constraints such as: varying board sizes, reasonable channel widths in multiples of a byte, limited pins from a router chip, and limited pinout from board depending on its size. The set of design-feasible configurations conforming to the packaging constraints are first derived. The demanded performance (latency and throughput) are then derived for a given processor and interconnect technology. The latency-throughput performance characteristics offered by each design-feasible configuration is estimated through analytical modeling and compared to the demanded performance. Among the good design-feasible configurations satisfying the demanded performance, the *best* configuration is decided on the basis of cost-effectiveness and scope for scalability. The elegance of the framework lies in its parameterized representation of different technologies. The best configuration can be derived for any set of technologies and constraints by choosing appropriate values for the parameters. We illustrate the framework by deriving best configurations to design a 1024 processor system. Using the framework we also study the impact of various packaging parameters and demanded performance on the overall design process. Among others the effect of varying maximum board size, higher router pinout, and alternate board pinout technology on the set of design-feasible/good/best configuration is analyzed and suitable design guidelines are derived to aid an architect in the design process. For a wide range of parameters, it is shown that best configurations are achieved with up to 8 processors per cluster and 3D-5D inter-cluster interconnection. The proposed guidelines and results are verified through accurate simulation modeling.

The paper is organized as follows. In Sec. II we present the two-level k-ary n-cube cluster-c architecture. The impact of processor speed and communication link speed on the demanded performance is discussed in Sec. III. In Sec. IV we present a representative multi-level packaging model for clustered systems. Section V discusses the trends in growth of processor board sizes, pinout, channel width, and router pinout technologies. In Sec. VI we derive expressions for offered channel width and bisection bandwidth under different packaging, processor, and interconnect technologies. Section VII presents a latency-throughput performance model for k-ary n-cube cluster-c clustered systems. In Sec. VIII we present the integrated framework and discuss important considerations in choosing the best configuration. Section IX illustrates the framework to design a 1024 processor system. Section X demonstrates the impact of varying packaging and demand parameters. Finally, concluding remarks and future work are presented.

II. DESIGNING SYSTEMS WITH k-ARY n-CUBE CLUSTER-c ORGANIZATION

A. k-ary n-cube cluster-c Organization

Many current parallel systems like the CRAY T3D [7], Intel Paragon [13], and the Stanford DASH [10] are taking a two-level clustering approach. Recently, we have introduced a new k-ary n-cube cluster-c organization [4], [5], [16] to capture this upcoming trend in building scalable parallel systems. In this organization, the lower level consists of k^n processor clusters. 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 network-based internet providing easy scalability and b) processor clusters providing the convenience of packaging modularity and potential for better exploitation of communication locality. Figure 2 shows the overall configuration of such a system.

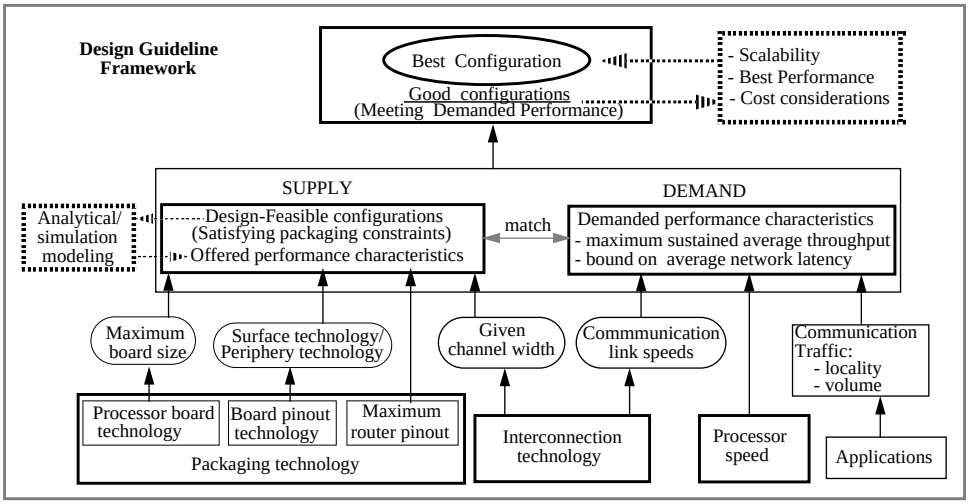


Fig. 1. A supply-demand optimization framework for designing and developing scalable architectures under varying processor, interconnection, and packaging technologies.

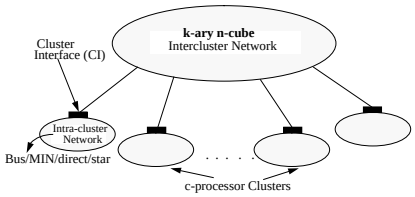


Fig. 2. Two-level clustering with k-ary n-cube cluster-c organization.

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 [10], [16]. However, such discussion is beyond the scope of this paper and we emphasize only on the design framework.

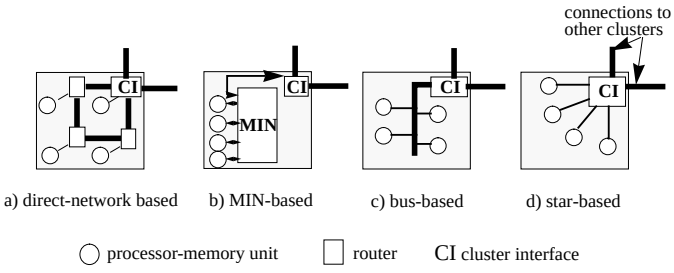


Fig. 3. Four possible cluster interconnections under k-ary n-cube cluster-c organization.

The memory in such systems is distributed physically across the clusters. 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 design and analysis presented in this paper. Therefore, without loss of generality we assume the cluster memory to be distributed uniformly amongst the processors in the cluster.

B. Architectural Alternatives and Choices

To build an N-processor system, there are vast number of possible alternatives with clustered configurations. 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, a) satisfies packaging constraints, b) meets desired performance of latency and throughput, and c) is easily scalable to larger sizes.

It is easy to observe that the flat k-ary n-cube systems 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 [19], 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 I provides a summary of the main symbols and notations used in this paper. In the remaining discussion we represent a k-ary n-cube cluster-c organization using a compact notation of (k^n, c). For example, an 8x8 inter-cluster network with clusters of four processors each is represented as $(8^2, 4)$ or $(8\times8, 4)$. Similarly, a configuration with mixed radix inter-cluster network with 4x4x3x3 topol-

ogy and cluster size 2 is represented as $(4^2 \times 3^2, 2)$.

TABLE I
SUMMARY OF IMPORTANT SYMBOLS AND NOTATIONS USED IN PAPER.

a	Unit area to hold a processor chip, associated memory, and router logic
b	Capacity 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
(k^n, c)	Shorthand representation for a k-ary n-cube cluster-c organization
n	Dimensionality of inter-cluster network
p_p	Pinout density under periphery pinout technology
p_s	Pinout density under surface pinout technology
r	Computation to communication ratio (indicates application characteristics)
t	Cycle 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	Bisection size in the inter-cluster network
D	Computational speed of a processor (indicates processor technology)
L	Message length in bits
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)
R	Maximum pinout supported from a router
T_c	Average message latency in presence of contention
T_{max}	Upper bound on T_c
W	Inter-cluster channel width (number of wires)
W'	Channel width supported by technology
W_p	Maximum value of W constrained by board pinout P_b
λ	Demanded average throughput per processor reflecting both processor and communication link technologies. $\lambda = Dt/r$

III. DEMANDED NETWORK PERFORMANCE

In this section we characterize the demands made on the average system throughput and average network latency of an interconnection network. These demands are characterized by parameterizing processor and interconnection speed and communication characteristics of applications. We first analyze the impact of processor speed and communication link technology on the demanded average throughput in the system. The importance of a bound on the average network latency of a message is discussed next.

A. Effect of Processor and Interconnect Speed on Demanded System Throughput

With advances in processor technology, the rate at which processors can execute instructions is going up. Let us denote the computational speed of a processor as D MFLOPS. An increase in processing power leads to a higher value of D. As a general observation in applications, an amount of computation is associated with a quota of necessary communication, expressed as the *computation to communication ratio* (r FLOPs/bit) [20]. Therefore while computing at the rate D MFLOPS a processor demands a sustained communication rate or *average throughput* of (D/r) Mbits/sec. To allow such a demanded traffic rate the underlying interconnection network should be able to sustain a minimum of (D/r) Mbits/sec per processor. A typical estimate of the range of computation to communication ratio, as suggested in [20], is $r = 0.125$ to 1.25 FLOPs/bit. For example, in a system with $D = 100$MFLOPS and $r = 0.50$ FLOPs/bit the expected throughput demand on the network is (100/0.50)Mbits/sec or 200×10^6bits/sec per processor.

To capture the competitive growth of the processor and interconnect technologies, we represent the above processor demand on average throughput in terms of (bits/network cycle) instead of (bits/sec). Let t denote the network *cycle* time (seconds), the time to transfer a bit across a network wire. A reduction in t allows more data to be sent across any wire in a given time leading to higher channel bandwidth. The average throughput of (D/r) bits/sec can be rewritten as $(D/r)t$ bits/cycle. Observe that an increase in the value of D captures the advancement in processor speed while a decrease in the value of t captures that in link speed. To capture both these technological advancements together, let us introduce a parameter $\lambda = (D/r)t$ for demanded average throughput in bits/cycle. It is to be noted that if processor and link speeds increase in the same proportion then λ does not change. For deriving representative values of λ in current and future systems, we consider predictions made by Patterson [17]. As shown in Table II, for a computation to communication ratio $r = 0.5$ FLOPs/bit the values of λ broadly lie in the range of 0.5-2.5 bits/cycle. For a lower value of r the corresponding values in the range of λ would be higher.

TABLE II
SAMPLE VALUES OF λ FOR VARIOUS REPRESENTATIVE COMBINATIONS OF PROCESSOR AND INTERCONNECTION LINK SPEEDS. A COMPUTATION TO COMMUNICATION RATIO OF $r = 0.5$ FLOPS/BIT[20] IS ASSUMED.

Processor		Interconnection Link		$\lambda = (D/r)t$ (bits/cycle)
Speed D (MFLOPS)	Demand on Network D/r (bits/sec)	Speed (MHz)	Channel Cycle Time t (sec/cycle)	
50	1×10^8	150	6.6×10^{-9}	0.66
50	1×10^8	100	10.0×10^{-9}	1.0
100	2×10^8	150	6.6×10^{-9}	1.3
250	5×10^8	200	5.0×10^{-9}	2.5
250	5×10^8	500	2.0×10^{-9}	1.0
250	5×10^8	1000	1.0×10^{-9}	0.5
1000	2×10^9	1000	1.0×10^{-9}	2.0

The *bisection size* 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 [8]. It limits the number of bits that can cross from one half to another half of the network. Let us analyze the impact of sustaining an average throughput of λ bits/cycle per processor on the network bisection size in the inter-cluster network. The total traffic injected by all processors per cycle is $N\lambda$ bits. For uniform traffic, the probability of a generated message being intra-

cluster is c/N. Similarly, the probability of a message being inter-cluster is $(1-c/N)$. Thus, on the average $N\lambda(1-c/N)$ bits get injected into the inter-cluster network every cycle. On the average half of this inter-cluster traffic is destined to processors on the other half of the system. These messages require to cross the inter-cluster bisection indicating a demand of $N\lambda(1-c/N)/2$ bits across the bisection every cycle. Denoting the inter-cluster bisection size as B wires, it is clear that $B = N\lambda(1-c/N)/2$ wires are needed to sustain the demanded traffic. However, due to contention in the network the utilization of the network channels are only a fraction of the maximum capacity [8]. To compensate for such loss in bandwidth due to contention, the actual value of B required to support a throughput of λ bits/cycle has to be greater than $N\lambda(1-c/N)/2$.

Similar to the demand on the inter-cluster bisection size to sustain a given throughput, the traffic going in/coming out/traversing from one processor to another inside a cluster also imposes a demand on the intra-cluster bisection size. This demand is clearly proportional to the size of a cluster. Thus, to support larger sized clusters, the intra-cluster bisection size should scale linearly with size c. Since packaging constraints are less rigid in lower hierarchies, it is possible to provide thicker channels/buses to achieve higher bandwidth inside a cluster [19]. Hence in this paper, we focus only on the bisection size of the inter-cluster network.

B. Demand on Average Network Latency

The achievable average message latency can have a direct impact on the processor efficiency in a multiprocessor system. For example, consider a parallel application in which each processor in the system executes a series of computation blocks of 400 cycles each. At the start of a block each processor sends out request messages to other processors, the reply to which is checked at the end of the block execution (can be considered as *prefetching* data). Assume a processor continues execution of its next block only after it receives the reply to the previous request. Let the amount of time spent by a remote processor in sending out the reply message be small. Thus, it is critical that the average one-way message latency in the network be less than 200 cycles for a round-trip latency of less than 400 cycles. Otherwise, a processor is forced to idle while waiting for a reply message leading to a fall in performance. Let T_{max} denote such an upper bound on the average network latency expected/demanded in a system. Clearly the value of T_{max} depends on many factors like nature of the application and state of compiler technology. For example, the extent to which prefetching of data can be employed varies from application to application. Similarly, the state of compiler technology dictates the degree to which prefetching can be exploited. For illustrative purposes in this paper we assume typical values of T_{max} to be 100-200 cycles [10].

In this section we have defined demanded network performance in terms of a) sustained average throughput (λ bits/cycle) and b) bound on average message latency (T_{max} cycles). We refer to this as the *demand side* to our design framework. In the following sections we develop the *supply side* to our framework. The next section presents a typical multi-level packaging model for designing large multiprocessor systems. The packaging technologies and constraints of these levels are discussed in Section V and their impact on the set of feasible (or supplied) configurations derived in Section VIII.

IV. PACKAGING MODEL FOR CLUSTERED SYSTEMS

Some earlier models, proposed to capture packaging constraints, like the VLSI model with limited bisection bandwidth as proposed by Dally [8] 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 factors like VLSI die sizes, chip pinout, board areas, board pinout, and number of boards per card-cage.

Figure 4 shows a typical multi-level hierarchy encountered while packaging a large clustered multiprocessor system. At the lowest level are processor, memory, and router VLSI chip modules. A single processor chip, its associated share of memory, routing, and other interface logic are usually referred together as a *processing node*. Multiple processing nodes are organized into *cluster modules* and placed on processor boards. A cluster module is usually compact and does not span across multiple boards. Figure 4 shows each board in the system having four cluster modules placed on it. Multiple boards are organized into a card-cage and a system may comprise of multiple such card-cages. For example, the system in Fig. 4 requires two card-cages each containing four boards. A larger system would require more boards and card-cages.

The modules at each level of this packaging hierarchy: chips, clusters, boards, card-cages have their own characteristics in terms of maximum capacity, bisection size, available pinout, and channel width. For example, the pinout from a cluster module may be limited to a maximum of 250-300 pins. Similarly, the maximum allowable area of a processor board, depending on the size of a cluster module, may be able to accommodate only up to 4-8 cluster modules. Such packaging characteristics are analyzed in more detail in the next section.

The above packaging hierarchy has a direct impact on the connections of the inter-cluster channels. For example, let us consider two clusters connected by an inter-cluster channel. This channel utilizes the pinout from the associated cluster modules. The two clusters may lie on the same or on different boards. A channel between two clusters on the same board is connected through wires on the board. Such wires directly connect pins of one cluster module to another. However, connecting a channel between clusters on two different boards is more difficult. Pins from the cluster modules need to be first connected to pins of the respective boards. Figure 4 shows such board pins along the periphery of each board. The channel is completed by connecting the pins between the two board modules. This inter-board part of the channel is implemented either

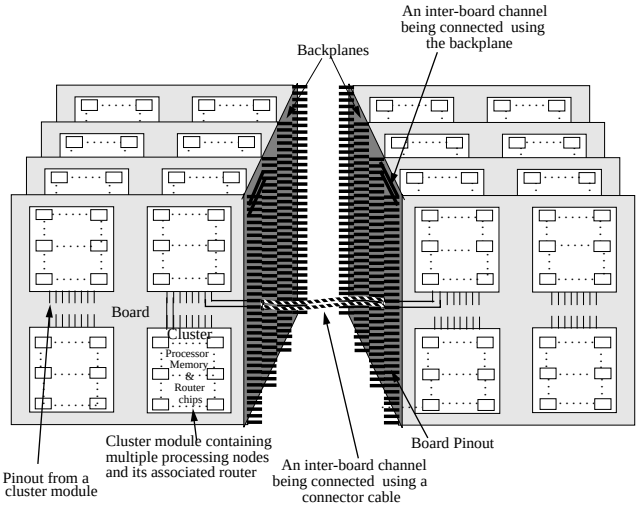


Fig. 4. Typical multi-level hierarchy used in packaging a large clustered multiprocessor system.

through a) shared backplane, for boards within the same card-cage or b) connector cable, for boards across different card-cages. Figure 4 shows examples of these two types of inter-board channels.

Large system size and multiple levels of packaging have significant impact on the length of connecting wires in the system. The length of wires to connect channels depends on the distance between the connecting boards. Longer wires can lead to longer propagation delays and hence longer channel cycle times. However, this problem is alleviated by applying pipelining techniques over long wires [22]. In this study we assume such techniques being used to limit channel cycle time.

V. PARAMETERIZING PACKAGING TECHNOLOGIES

The characteristics and limitations of each level of packaging has an extreme impact on the set of achievable or feasible configurations and their cost and performance. In this section we discuss important trends in board capacity, board pinout, channel width, and router pinout technologies. These technologies are parameterized and their impact on design constraints are analyzed.

A. Processor Board Technology

Processor boards cannot be arbitrarily large in size. The physical size of a board is restricted by electrical, mechanical, and board fabrication constraints. In terms of physical dimensions board sizes, being used in recent multiprocessor systems, vary from 6"×4" to an aggressive 26"×21" used in the J-machine [14]. The largest board size available to a system designer usually varies with technology and over a period of time. In this study we therefore do not present guidelines restricted to a particular largest board size. The available maximum board size is treated as a parameter in the framework. Depending on the size of a given board we can fit only a limited number of cluster modules or processing nodes on it. For example, consider a design problem with board sizes of 12"×6" and two processing nodes per cluster module. Let the size of a processor chip be 2"×2", the area occupied by associated local memory of 16MB (say) be 6"×4", and required intra-board estate for router, memory and address buses and other support logic be 8 square inches. In such a design we require around 36 square inches per processing node leading to 72 square inches per cluster module. This implies that we can fit exactly one cluster module or two processing nodes on each board.

It is natural to express board area or capacity in terms of absolute units such as square inches. However, for ease of discussion, it is more convenient to express it in relation to the area of a processing node. For example, consider a given board with capacity specified as 8 processing nodes. This implies that such a board is also capable of holding 4 clusters of 2 processing nodes each, or similarly, 2 clusters of 4 processing nodes each. As discussed earlier we define the size of a processing node to include the area required by a processor chip, its local memory, router, and other associated interface logic. Such a combined area is denoted as parameter a. The magnitude of a depends mainly on two factors: a) level of integration in the VLSI chips and b) the size of memory associated with each processor. A higher VLSI technology leads to smaller a. Similarly, supporting larger memory per processor leads to larger a. A designer can choose an appropriate value for a depending on chip sizes and amount of memory per processor and use this as an unit of board capacity. A board with physical area (ba) is defined to have a capacity of (b). Thus, in the example in the last paragraph $a = 36$ and a 12"×6" board has a capacity of $b = 2$. Future advancements in VLSI technology can lead to higher processor and memory integration. Let this be reflected in our design problem by choosing a suitably smaller value of $a = 24$. This implies that the same 12"×6" board in terms of the new value of a has a capacity of $b = 3$ processing nodes. In further discussion, unless otherwise mentioned, we use the terms board size, board area, and board capacity in an interchangeable manner.

Assume a cluster module containing c processing nodes require an area ca. Thus, a board with capacity b can hold up to $b' = b/c$ clusters on it. To build a system with N processing nodes a total board area of Na is necessary. A total board area larger than Na may be used but this clearly leads to wastage of precious board estate. Thus, we suggest using a total board area close to Na. We denote the maximum board size discussed earlier as parameter b_{max}. For illustrative purposes, we later consider a b_{max} of 8 and smaller in this paper. However, the framework is valid for any value of b_{max} being offered by technology. All boards used in a given system are assumed to be of the same size and board capacity satisfies the constraint, $1 \leq b \leq b_{max}$. Total number of boards used in designing an N-processor system is defined as $N_{boards} = N/b$.

B. Board 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 pinout: This is the traditional technology [23], [24] where the periphery of a board is used for external connections. The exact relationship between the size of board periphery and pin-count depends on the pin connector technology. However, it is reasonable to expect this relationship to be linear. Thus, a larger board, having a longer periphery, can support more pins. Typically only one or two sides of a board are used for periphery pinout. Without loss of generality let us assume only one side being used for pinout as shown earlier in Fig. 4. Let p_p denote the peripheral pinout density, the pin-count that can be supported from an unit board length. Thus, a larger square board with capacity b having each side of length $\sqrt{b}$ units, can support a total of $P_b = p_p\sqrt{b}$ pins.

b) Surface pinout: This is representative of a more aggressive pinout technology. The surface of the board is utilized for external connections. Representative examples are electronic interconnections using elastomeric connectors [14] and optical interconnections [21]. Let p_s denote the surface pinout density, the pin-count that can be supported from a board of unit capacity. Assuming a linear relation between board area and surface pinout, the pin-count supportable from a board of capacity b is $P_b = bp_s$.

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. As expected, for large board sizes the surface technology

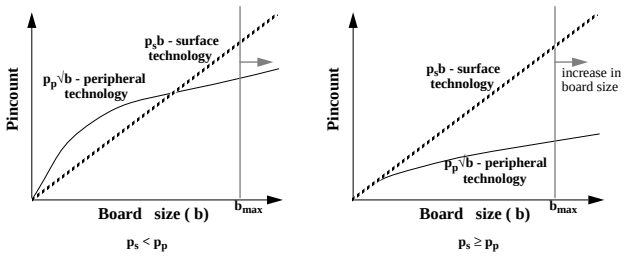


Fig. 5. Growth of pincount with board area under peripheral and surface technologies shown for two different relationships between p_p and p_s. The parameters p_p and p_s reflect the pincount that can be supported from a board of unit capacity under periphery and surface pinout technologies, respectively.

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 [7], [14] are: $p_p = 128 - 256$ and $p_s = 64 - 128$. Future values of these parameters will depend on the advances made in the respective technologies. In the next section we compare the impact on designing clustered systems under both these pinout technologies.

C. Channel Width Technology

Most current parallel machines have 8 and 16-bit data channels. This corresponds to a channel width of $W \approx 12$ and 24, respectively, including control, acknowledgment, and parity wires. 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. Further, factors like path-width inside routers, and connector technology restrict channel widths from being arbitrarily large. Most previous studies while proposing design guidelines, did not consider such constraints on channel width. In our framework we account for these and denote a channel width supportable by technology as W'. For example, current technology supports $W' = 12$ and 24. In the near future it is expected that a wider channel width technology with ability to carry 32-bit data [6], [17], corresponding to $W' \approx 40$, would be feasible. In the next section we investigate the impact of such supportable fixed channel width (W') on the set of design-feasible configurations.

D. Router Pinout Technology

Similar to the restriction on maximum pinout from a processor board, the pinout from a router is also limited. The available number of pins from a router restricts the number of channels and channel widths that can be supported[2]. It is to be noted that as discussed in Sec. IV the router is a part of the cluster module. Thus, in this organization router pinout is synonymous with cluster module pinout. Let R denote the maximum pinout from a router (cluster module) being available for inter-cluster channels. Let us consider the required pinout for inter-cluster channels from a router in a (k^n, c) system. In a multi-dimensional system a router has to support at least two channels along any dimension, one for an incoming and another for an outgoing channel. For a system with bidirectional channels this number is four per dimension. The number of channels to be supported from a router assuming unidirectional channels is $2n$. Supporting W bit wide channels in such a system requires a pinout of $2nW$ from each router. Clearly, this imposes a constraint of

$$2nW \leq R. \tag{1}$$

As an illustration consider a representative value of $R \approx 250$. Using the above equation let us derive the maximum dimension (n) that can be supported in an interconnection. For various current and future channel width technologies: $W' = 12, 24$, and 40, the maximum dimension supported are $n = 10, 5$, and 3, respectively. Thus, it can be observed that even with a conservative pinout technology and a channel width technology of $W = 40$, up to 3D systems are feasible. For thinner channels the maximum supportable dimension can be even higher.

VI. IMPACT OF PACKAGING ON ARCHITECTURAL PARAMETERS

In this section we analyze the impact of packaging constraints on the channel width and bisection size of k-ary n-cube cluster-c configurations. The maximum offered channel width is derived based on board size and pinout constraints. We then analyze the impact of maintaining a

fixed supportable channel width on system dimensionality and cluster size. It is then shown that a realistic design can lead to under-utilization of board resources. Minimizing such under-utilization is necessary to minimize system cost. Based on this, a process of deriving system configurations minimizing under-utilization is proposed. We then analyze the impact of increasing system and cluster size on the bisection size of a system while maintaining a fixed supportable channel width. Most results presented in this section hold for both periphery and surface pinout technologies. In case of differences these are specifically mentioned. All results are derived for arbitrary cluster sizes ($c \geq 1$) and the corresponding results for flat architectures can be obtained by choosing $c = 1$.

A. Offered Inter-Cluster Channel Width

Let the clusters in a system be placed on multiple boards and interconnected by channels to yield a $N' = N/c = k_1$ x k_2 x $\cdots k_n$ inter-cluster topology as discussed earlier in Section IV. Each board in this system holds b' clusters. The clusters on any board together with the inter-cluster channels between them can be visualized as a sub-topology of the overall inter-cluster topology. Let this sub-topology be represented as ($b' = b_1$ x b_2 x$\cdots$x b_n), where $\forall_{i=1}^{n}(b_i < k_i)$ and some b_i's may be 1 to depict that only one cluster exists on the board along that dimension. The condition $\forall_{i=1}^{n}(b_i < k_i)$ is assumed instead of $\forall_{i=1}^{n}(b_i \leq k_i)$ to reflect a typical constraint that any inter-cluster dimension is large enough such that it can not fit on a single board. This is reasonable to expect for large system sizes. Figure 6 shows an example system with 48 clusters being placed on 12 boards each with a capacity for holding four processor clusters. The overall inter-cluster topology is assumed to be ($N' = 4$ x 4 x 3). The sub-topology on each board is assumed to be ($b' = 2$ x 2 x 1). Let the number of rows of processor clusters in dimension i of the sub-topology on a board be denoted as r_i. This can be derived as $r_i = b'/b_i$. For example, the number of such rows on each board in Fig. 6 along dimensions marked 1, 2, and 3 are 2, 2, and 4, respectively. Note that inter-board channels go out/come in from the two ends of each row leading to $2r_i$ inter-board channels[1]

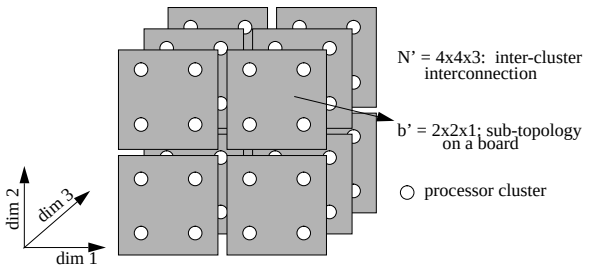


Fig. 6. A system with 48 clusters being placed on 12 boards each with a capacity for holding 4 processor clusters. The overall inter-cluster topology is 4x4x3. The sub-topology placed on each board is 2x2x1.

[1]This is true for unidirectional channels. For full duplex bidirectional channels this number would be $4r_i$.

along dimension i. Hence, the total number of inter-board channels required to go out/come into a board from all n dimensions is derived as $2\sum_{i=1}^{n} r_i = 2b'\sum_{i=1}^{n}(1/b_i)$. As discussed earlier in Section IV inter-board channels between boards in the same card-cage can be connected through a shared backplane. The connections from a board to the backplane are made using the pinout from the board. In large systems with many boards and multiple card-cages a single backplane cannot be used to connect all boards. Thus, cable connectors also need to be employed to connect the inter-board channels across card-cages. Here too, the connections from the board to cable connectors are established using the pinout from the board. Thus, the inter-board channel width is constrained by the available board pinout. To support a channel width of W, a total pincount of $2b'W\sum_{i=1}^{n}(1/b_i)$ is required from each board. Thus, the maximum supportable inter-board channel width from pinout restrictions, is derived as:

$$W = \frac{P_b}{2b'\sum_{i=1}^{n}(1/b_i)} \tag{2}$$

Assuming that the sub-topology on the board is regular[2], i.e. ($\forall i : b_i = b'^{1/n}$) and using $b' = b/c$, this can be simplified to,

$$W = \frac{P_b}{2nb'^{(n-1)/n}} = \frac{P_b}{2n(b/c)^{(n-1)/n}} \tag{3}$$

Similar to the inter-board inter-cluster channels connecting clusters across different boards, intra-board inter-cluster channels are required to connect the clusters on a single board to maintain the (b_1 x b_2 x... x b_n) sub-topology on a board. These channels can be implemented using on-board wires. For an on-board implementation it is intuitive to expect the available board bisection to limit the size of such intra-board channels. However, with multi-layered boards, it is reasonable to assume on-board connection density to be much higher than off-board density. Hence the inter-cluster channel width is mainly limited by the inter-board channel width determined by the pinout constraint as shown in Eqn. 3.

From the expression for W in Eqn. 3, 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. This observation matches with a similar result shown in [2], [8] while designing flat systems. Similarly, for a given pinout technology and board size, decreasing the

[2]Assumption of regular topology wherever made in this section is only for formula simplification and providing a more intuitive understanding of the interplay between various parameters. However, the proposed framework is also valid for the mixed radix topologies. Design solutions together with simulation results are presented later with mixed radix topologies.

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. These observations can be summarized as:

RESULT 1: For a fixed pinout from a board of given size, the offered channel width (W) falls with increase in the inter-cluster network dimensionality (n) while keeping cluster size (c) fixed. Similarly, channel width (W) falls as the the inter-cluster dimensionality (n) remains fixed and the cluster size (c) reduces.

Let us consider the impact of board size on channel width W. Under periphery pinout technology, with $P_b = p_p\sqrt{b}$ we can simplify Eqn. 3 as:

$$W = \frac{p_p c^{(n-1)/n}}{4b^{(n-2)/(2n)}n}. \quad (4)$$

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. For a 2D inter-cluster network ($n = 2$) the channel width remains constant while that in an 1D network rises sharply. Under surface pinout technology, with $P_b = p_s b$, a similar simplification leads to:

$$W = \frac{p_s c^{(n-1)/n} b^{1/n}}{4n}. \quad (5)$$

Here, 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. This leads to:

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

B. 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 [8], did not impose any restrictions on the values that the channel width can take while satisfying other constraints. As discussed in Section V-C supporting an arbitrarily large channel width is difficult. Further the data lines are expected to be integral multiples of bytes for easier interfacing with processors and memories. Representative values of supportable channel width (W') as discussed earlier in Section V-C are 12, 24, and 40.

In Equations 4 and 5, we showed that offered channel width W is a function of system parameters n, b, c, and p_p or p_s. Given a channel width technology W' to be supported, 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 (W'). Based on Equations 4 and 5 we analyzed the relationships among these parameters and the impact of varying one parameter on another was studied while maintaining a fixed $W(= W')$. Table III summarizes some of the important interplays. An entry in the table: *fixed*, ↑, or ↓ corresponds to the respective parameter being kept fixed, increased, or decreased, respectively. For example, the first row of the table indicates that for a given fixed board size and pinout density, increasing (↑) cluster size requires the inter-cluster dimensionality to be also increased (↑). Similarly, second and third rows denote the impact of increasing board pinout density on inter-cluster dimensionality and cluster size, respectively. With increase (↑) in board pinout density the supported pinout from a given board is higher. The increased pinout can be used to support more inter-cluster channels from a board while maintaining a given channel width. The extra channels can support a) higher (↑) dimensional inter-cluster networks requiring more channels or b) smaller (↓) clusters. With smaller clusters, building a system of given size requires more clusters. Interconnecting a larger number of clusters, while maintaining the inter-cluster dimensionality and channel width, requires more channels in the system. Increased pinout from boards make it possible to support the needed extra channels. It may be noted that relationships obtained by reversing the sense of all arrows in Table III also hold true.

TABLE III
SUMMARY OF INTERPLAY BETWEEN BOARD SIZE (b), PINOUT DENSITY (p_p OR p_s), CLUSTER SIZE (c), AND INTER-CLUSTER DIMENSIONALITY (n) TO MAINTAIN A FIXED CHANNEL WIDTH (W').

Board Size (b)	Pinout Density (p_p or p_s)	Cluster Size (c)	Inter-Cluster Dimensionality (n)
fixed	fixed	↑	↑
fixed	↑	fixed	↑
fixed	↑	↓	fixed
↑	fixed (periphery)	↑	fixed
↑	fixed (surface)	↓	fixed

C. Designing with Under Utilization of Resources

The expression in Eqn. 3 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. For example, a fraction of the board pinout capacity may remain unutilized if a larger board size is required to fit a desired topology but the resulting larger pinout is not required. To make our design framework more realistic, we allow such under-utilization of board and pinout capacities. However, in order to minimize system cost we also add a design objective to minimize such under-utilization of resources. Let us denote parameters u_p and u_b as the percentage utilization of board pinout and board area, respectively. Now consider supporting a channel width of $W = W'$ with marginal under-utilization of board area and pinout resources being allowed. Based on Eqn. 3 we

can derive,

$$W' = \frac{P_b u_p}{4nu_b(b/c)^{(n-1)/n}} = W\frac{u_p}{u_b} \quad (6)$$

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:

$$W = W'(u_b/u_p) \Rightarrow W \leq 1.1W' \quad (7)$$

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

$$W = W'(u_b/u_p) \Rightarrow W \geq 0.9W' \quad (8)$$

The derived solution configurations, in terms of (n, b, c), comply with supported channel width (W') technology while utilizing board resources maximally. In Section VIII we demonstrate that a subset of these configurations, while complying with other packaging constraints like maximum router pinout and maximum board size, form the set of *design-feasible* or package-able configurations.

D. Offered Inter-Cluster Bisection

Now let us consider the impact of packaging on offered inter-cluster bisection size. As discussed earlier in Section III-A the bisection size of the inter-cluster network B, or more specifically bisection size per processor B/N, is indicative of the supportable average throughput per processor in the system [8]. Observations derived in this subsection for the bisection size per processor reflect similar trends on system throughput. These observations are expected to be broad guidelines to aid the design decision process. A more accurate modeling of the average throughput in presence of contention is presented in the next section.

Given a (k_1 x k_2 x... x k_n) mesh/torus inter-cluster network with a given channel width, W, 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, one orthogonal to y dimension along xz plane, and so on. We are interested in the size of the smallest bisection in the system 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 on either side of such a bisection is given by N'/k_{max}, where N' as defined in Sec. VI-A denotes the total number of nodes (clusters) in the system. Let each node on one side of the bisection be connected to exactly another across the bisection using an unidirectional channel of width W. This leads to the offered bisection size, B, being $\frac{N'W}{k_{max}}$ wires. Thus, the expression for B/N, the bisection size per processor, is derived as $\frac{N'W}{Nk_{max}}$ wires. The presence of wrap-around channels in torus doubles this number[3] to $\frac{2N'W}{Nk_{max}}$. The expression can be simplified by assuming the inter-cluster network to be regular i.e. ($k_{max} = (N/c)^{1/n}$) and $N'/N = 1/c$ to:

$$B/N = \frac{2N'W}{Nk_{max}} = \frac{2W}{c(N/c)^{1/n}} = \frac{2W}{N^{1/n}c^{(n-1)/n}} \quad (9)$$

Based on the above equation the following observations can be made:

RESULT 3: For a given channel width ($W = W'$) and inter-cluster dimensionality ($n > 1$), the offered bisection size per processor (B/N) falls with increase in the system size (N) while keeping cluster size (c) fixed. Similarly, the bisection size per processor (B/N) also falls as the system size (N) remains fixed and cluster size (c) is increased.

Figure 7 shows the bisection size per processor for a system size of $N = 1024$ processors, a given channel width of $W' = 24$, and different inter-cluster dimensionality ($n = 1 - 4$) as cluster size is varied ($c = 1 - 16$). It can be observed that the fall in bisection size per processor is more appreciable at smaller cluster sizes and higher inter-cluster dimensionalities.

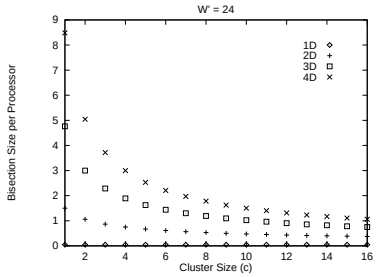


Fig. 7. Offered inter-cluster bisection per processor (B/N) as a function of cluster size and inter-cluster dimensionality while maintaining a channel width $W' = 24$ for a 1024 processor system. Note that for a given pinout technology, the board sizes are not fixed and are changed with cluster size to maintain $W' = 24$.

It is also interesting to observe the impact of board size on offered bisection size per processor under two different pinout technologies. Figure 8 shows the plots of B/N for a $N = 1024$ system. The trends are shown with respect to periphery and surface pinout technologies as board size is increased. To analyze these trends we first simplify Eqn. 9 by using Eqn. 3 to replace W, leading to:

$$B/N = \frac{2}{N^{1/n}c^{(n-1)/n}}\frac{P_b}{2nb'^{(n-1)/n}} = \frac{P_b}{N^{1/n}b^{(n-1)/n}n} \quad (10)$$

[3]For bidirectional duplex channels this value needs to be multiplied by another factor of 2.

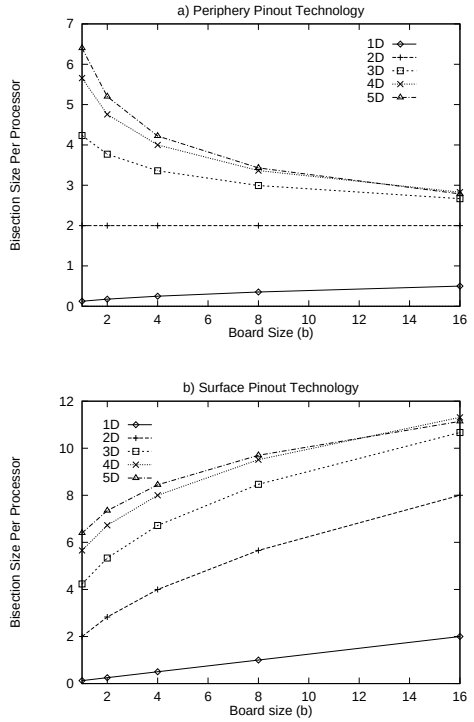


Fig. 8. Offered inter-cluster bisection per processor (B/N) in a system of 1024 processors as a function of board size and different inter-cluster dimensionalities under two different pinout technologies: a) periphery pinout with $p_p = 128$ and b) surface pinout with $p_s = 128$.

For a periphery pinout technology with $P_b = p_p\sqrt{b}$, this can be further simplified to $B/N = \frac{p_p}{N^{1/n}b^{(n-2)/(2n)}n}$. Thus, for a given system size (N) and inter-cluster network dimensionality (n), the bisection size per processor (B/N) falls with increase in board size (b) for $n > 2$. This can be explained by the fact that under periphery pinout the total board pin-count in the system reduces as board sizes are increased. For $n = 2$, the value of B/N remains fixed and for $n = 1$ it rises slowly. The key point is that under periphery pinout technology, it is ideal to work with smaller boards as they offer higher bisection sizes and hence performance. Similarly, under surface pinout technology with $P_b = p_s b$, we can simplify Eqn. 10 as $B = \frac{p_s b^{1/n}}{N^{1/n}n}$. Here, it can be observed that for a given system size (N) and inter-cluster network dimensionality (n), the size of the bisection increases with n. These lead to:

RESULT 4: 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 per processor. However, under surface pinout technology, as board size (b) is increased, bisection size per processor increases for all inter-cluster dimensionalities.

Bisection size per processor has direct impact on the maximum traffic that can be supported in a system. Thus, trends similar to those presented in this section with respect to bisection size per processor are expected in the maximum value of the offered average system throughput. In the following section we present a simple analytical model for (k^n, c) systems to estimate their performance in presence of contention. We use such a model in Sec. VIII to determine a more accurate trend about the average system throughput and offered average message latency while deriving configurations supporting a demanded performance.

VII. LATENCY-THROUGHPUT PERFORMANCE MODEL FOR k-ARY n-CUBE CLUSTER-c SYSTEMS

A. The Model

We develop a simple analytical model to predict performance in (k^n, c) systems with small cluster sizes. This model considers network contention and determines latency and throughput parameters for a given traffic load. Our model is an extension of the model proposed by Agrawal [2] to predict performance in flat k-ary n-cube networks with dimension-order [8] virtual cut-through routing. As shown in [2] the average message latency in the presence of contention through a k-ary n-cube network can be derived as:

$$T_c = \left[1 + \frac{mF^2}{(1 - mFd)}\frac{(d-1)}{d}\left(1 + \frac{1}{n}\right)\right]nd + F, \qquad (11)$$

where T_c denotes the average message latency expressed in network *cycles*. Network cycle, as defined earlier in Sec. III-A, is the time to send a flit across one hop in the system. The parameter d represents the average number of hops taken by a message in a dimension. For a network with unidirectional channels and wrap-around connections, $d = (k-1)/2$. For bidirectional channels this value is $(k-1)/4$. The message size expressed in flits is $F = L/W$, where L is the message length in bits and W the channel width. The parameter m is the message injection rate by a processor in terms of messages/cycle. From the above equation, it can be observed that $nd + F$ cycles is the minimum latency suffered by a message. This happens at a very small value of message injection rate.

In a clustered (k^n, c) system depending on whether a message is intra-cluster or inter-cluster it encounters different latencies. For relatively small clusters $(c \leq 8)$ it is reasonable to assume advanced packaging, faster interconnect, and wider buses inside a cluster leading to very high intra-cluster message bandwidth. With such high bandwidth it is reasonable to expect small delays inside clusters. Inter-cluster messages on the other hand need to travel across clusters. The average delay for inter-cluster traversal is relatively much larger than average intra-cluster delay.

Our objective is to derive the average message latency as a function of average message throughput. The average message latency in a clustered system is determined as a weighted sum of the intra-cluster and inter-cluster message latencies. The weights are the relative frequency of these messages being generated by a processor. Under uniform traffic, the probability of an intra-cluster message occuring is given by the expression (c/N). Clearly, for reasonably large systems and relatively small cluster sizes the value of c/N is negligible. For example, in a system with $N = 1024$ processors and a cluster size of $c = 8$ the value

of $c/N = 0.781\%$. The probability of a message being inter-cluster, $(1 - c/N)$, is therefore very close to 1. This indicates that most messages generated in the system are inter-cluster. Under such an assumption, the average message latency in the system is closer to the average latency of an inter-cluster message. In this analysis we therefore focus on deriving the average latency of an inter-cluster message. An uniform traffic model was chosen because it is considered more representative while designing a general purpose machine. No prior knowledge is assumed about the nature of the applications to be run on it.

Let us consider the latency of an inter-cluster message. It has three components T_{intra_1}, T_{inter}, and T_{intra_2}. These terms denote the delays from the source processor to source-CI through the source intranet, source-CI to destination-CI through the k-ary n-cube internet, and destination-CI to the destination processor through the destination intranet, respectively. The exact expressions for T_{intra_1} and T_{intra_2} depend on the cluster topology, cluster size, and rate of traffic. However, from earlier discussion we know that these latency components inside clusters, T_{intra_1} and T_{intra_2}, are much smaller as compared to the inter-cluster component T_{inter}. For star-based clusters which is a popular trend in current multi-processor systems as discussed earlier in Section II, these components are negligible. Assuming a clustered system offering small intra-cluster latencies, the average latency seen by a message, T_c, is dominated by the average inter-cluster latency. To estimate this factor we modify the model in [2] in the following manner.

Given the message injection rate from each processor to be m messages/cycle, the combined traffic generated by all the processors in a cluster is mc messages/cycle. Under uniform traffic assumption, most of this traffic is inter-cluster. We assume this resultant traffic to be injected by the CI into the internet. Assuming the resultant traffic stream to be Poisson, the inter-cluster latency can be predicted from Equation 11 by replacing the parameter m by the expression mc leading to:

$$T_c = \left[1 + \frac{mcF^2}{(1 - mcFd)} \frac{(d-1)}{d} \left(1 + \frac{1}{n}\right)\right] nd + F. \quad (12)$$

Equation 12 can be rewritten to express injection rate (m) as a function of T_c and other parameters as:

$$m = \frac{(T_c - F - nd)d}{cFd((n+1)(d-1)F + (T_c - F - nd)d)} \quad (13)$$

B. Comparing Performance of the Model with Simulations

We validated the above model through simulations against a wide range of clustered systems with varying inter-cluster topology and cluster size with variety of workload parameters. It was found that the model closely predicted the simulated performance. For example, a comparison of simulation results with analytical prediction from the above model are depicted in Fig. 9. In Fig. 9(a) as message length F is varied from 2-12 flits in a 4-ary 3-cube cluster-4 $(4^3, 4)$ system, the model closely matches the simulation results. Similar matches were obtained with varying cluster sizes from 1 through 8, as depicted in Fig. 9(b). Our clustered simulation testbed models star-clusters, routes messages through the inter-cluster network in dimension-order, and generates statistics such as average message latency and observed message rates. Each simulation was run until the 95% confidence interval of a data point was within 5% of its mean. In the next section we demonstrate the use of the above model in deriving good configurations.

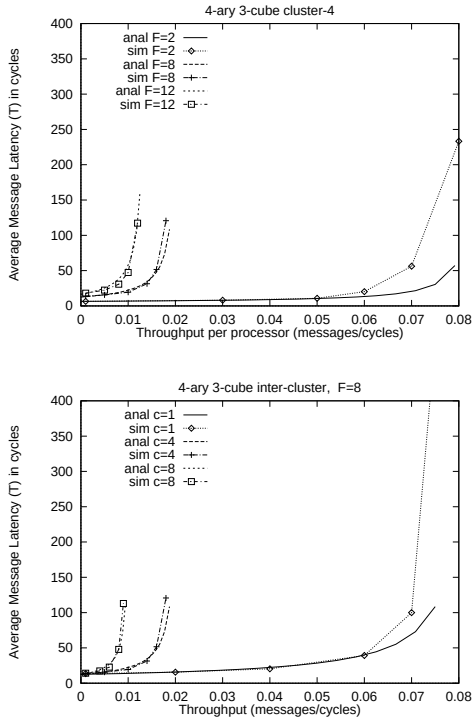


Fig. 9. Comparing the analytical latency-throughput model with simulations for: a) 4-ary 3-cube cluster-4 system with message length being varied from $F = 2$ to 12 flits and b) 4-ary 3-cube inter-cluster network with cluster size being varied from $c = 1$ to 8 while message length is kept fixed at $F = 8$ flits. Lines shown without points correspond to model predictions.

VIII. DESIGN FRAMEWORK METHODOLOGY

Let us put together all the components of our supply-demand framework, as discussed earlier. For a given system size, our goal is to derive the best configuration. This configuration should satisfy packaging and demand constraints in the most cost-effective manner. We formulate this as a search problem of selecting the *best* configuration from amongst the various configurations possible in the design space. The problem is solved in three phases.

Phase I: Deriving Design-Feasible Configurations
The set of configurations satisfying all packaging constraints are derived in this phase. These packaging constraints, as discussed earlier, are:

- All board sizes are less than a maximum board size $(b \leq b_{max})$.
- Pinout from a board is bounded by board size and pinout density ($P < P_b$, where $P_b = p_p\sqrt{b}$, under pe-

riphery pinout technology and $P_b = p_s b$, under surface pinout technology).
- Intended channel width is supportable by channel width technology ($W = W'$, where $W' = 12, 24$, and 40).
- Pinout from a router is less than maximum supportable pinout R.

For a given set of packaging parameters: a, b_{max}, p_p, p_s, W', R, and desired system size N, the design-feasible solutions can be easily derived through a computer-aided search. We refer to this set of design feasible configurations as the *supply* side.

Phase II: Deriving Good Configurations

The set of good configurations is a subset of the design-feasible configurations satisfying the *demanded* performance requirements. The demanded performance requirements refer to the upper bound on average network latency (T_{max} cycles) and a minimum average throughput (λ bits/cycle) as discussed earlier in Sec. III. For ease of discussion we represent the demanded minimum average throughput in terms of λ/L messages/cycle, where L denotes the length of a message in bits. For each design-feasible configuration, using Eqn. 13 we derive the value of maximum average throughput that can be sustained with average message latency being less than T_{max}. Such a maximum value of average throughput is denoted by m_{max} messages/cycle. A design-feasible system configuration offering $m_{max} \geq \lambda/L$ meets both latency and throughput demands on performance and is therefore marked as a good configuration. Any member from the set of good configurations can be used to build a machine under packaging constraints while offering the desired performance.

Phase III: Deriving the Best Configuration

The final phase of the problem is to select the *best* configuration from the set of good configurations. Cost-effectiveness, offered maximum throughput, and *scope for future scalability* are three important considerations for deciding the best configuration. For each good configuration our framework provides information such as exact board size used, number of inter-board connectors required, and total volume of wires. Such information can be used by a system designer to estimate the cost of a given configuration. However, a cost comparison is dependent on the exact cost model used. Thus, the search for the configuration with lowest cost may not be a very clear-cut process. An alternative approach is to select the good configuration offering the maximum value of m_{max} as the best configuration. Another factor considered is the potential of a configuration for future scalability to larger sizes to meet higher computing needs. Let us analyze this *scalability* aspect in choosing the best configuration.

As an example, let us consider the design-problem of building a system with $1K$ processors. Once a configuration (say n =4D, $c = 4$, $b = 8$, and $W = 24$) is chosen to fabricate a machine, these system parameters are fixed. To scale a given system to a larger size more processors need to be connected. However, to have a homogeneous scaled system the values of the parameters n, b, c, and W must be maintained. 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 $(n, b, c, W)_{2K}$ denote such a set for system size of $N = 2K$ processors. The intersection set $(n, b, c, W)_{1K} \bigcap (n, b, c, W)_{2K}$ denotes good configurations under both $N = 1K$ and $2K$ processors. Such configurations can be used to build a system with $1K$ processors which can be scaled up to $2K$ processors while still meeting demanded performance. We emphasize on such scalability in our framework to derive best configurations.

IX. APPLYING THE FRAMEWORK

In this section we illustrate the above three phase process by deriving the best configuration to design a system with $N \approx 1024$ processors for a representative set of packaging and technological parameters. The packaging parameters used are $b_{max} = 8$, $p_p = 128$, and $R = 250$. A demanded performance of $T_{max} = 200$ cycles, $\lambda = 3.0$ bits/cycle, and a message length of $L = 192$ bits are assumed.

A. Deriving Design-Feasible Configurations

Table IV summarizes various design-feasible configurations obtained through the framework for three different values of $W' = 12, 24$ and 40. The results are organized in columns with respect to number of clusters per board (b'). For example, column 3 in Table IV summarizes these solutions with one cluster per board ($b' = 1$). Similarly, columns 4, 5, and 6 in Table IV present solutions obtained with allowable number of clusters per board of b' =2, 4, and 8, respectively. Each entry is either a single cluster size or a range of cluster sizes. Such a cluster size coupled with the value of dimension (n), indicated on the row of the entry, represents a feasible configuration. For example, the entry of $c = 2$ in column 3 and the row corresponding to $W' = 24$ and $n = 4$ represents a feasible configuration 4D c-2. Blank entries in the table indicate that there was no valid configuration for that inter-cluster dimensionality, pinout technology, channel width, and board area. For a given value of W', the inter-cluster dimensionality was varied up to a maximum dictated by the router pinout constraint of $R = 250$, as discussed in Sec. V-D.

From Table IV we observe that for thinner supportable channel width ($W' = 12$), configurations with lower dimensionality are not design-feasible (hence not depicted in the table). This limits wastage of board resources as discussed in Sec. VI-C. Similarly, for wider channel width ($W' = 40$), configurations with higher dimensionality are not design-feasible in order to remain within the router pinout limit. From Table IV it can also be observed that for wider channel width technologies ($W' = 24$ and 40) clustered organizations ($c > 1$) are feasible while flat organizations are not. Note that the above representation of design-feasible solutions is independent of the total processors in the system. Given a system size of N processors, the above representation of feasible configurations can be expanded as explained in the next paragraph. In the remaining discussion we con-

TABLE IV

VALID RANGES OF CLUSTER SIZE (c) FOR PACKAGING CONSTRAINTS OF $b_{max} = 8$, $p_p = 128$, $R = 250$, AND THREE DIFFERENT SUPPORTABLE CHANNEL WIDTHS OF $W' = 12$, 24, AND 40. NUMBER OF CLUSTERS/BOARD (b') IS VARIED FROM 1 - 8.

	n	cluster size (c)			
		$b'=1$	$b'=2$	$b'=4$	$b'=8$
W'=12	4		1		
	5	1		2	
	6		2		
	7	2	3		
	8	2	4		
$W'=24$	3		2	2	
	4	2	3 - 4		
	5	3 - 4			
$W'=40$	2		2		
	3	3 - 4	4		

tinue the illustration of the framework for only one channel width of $W' = 24$. The framework can be similarly applied to other values of W'.

For each of the design-feasible configuration in Table IV we first derive the exact inter-cluster topology to realize a system with $N \approx 1024$ processors. For example, consider the entry in Table IV corresponding to $n = 4$, $b' = 1$, $c = 2$, and $W' = 24$. The desired number of clusters in this system with 1024 processors is $1024/2 = 512$. Let us consider selecting a 4D ($n = 4$) inter-cluster topology to interconnect 512 clusters. The closest configuration offering 4D inter-cluster is $5^3 \times 4$. The resultant system configuration has 600 clusters or 1200 processors which is more than the desired number of 1024. Without loss of generality we allow such deviations in system size up to reasonable limits. Similarly, other configurations corresponding to $W' = 24$ are also expanded. All design-feasible configurations corresponding to $W' = 24$ in Table IV are presented in the top half (corresponding to $N = 1024$) of Table V. It can be observed that configurations with identical values for n, c, W' but different b' lead to same inter-cluster topology. For example, consider the two entries in Table IV corresponding to $n = 3$, $c = 2$, $W' = 24$ but different $b' = 2$ and 4, respectively. Both lead to the same 8x8x8 inter-cluster topology. However, the sizes of the boards being used in the two configurations are different, $b = b'c = 4$ and 8, respectively. In a cost-model where board size is a factor in system cost, a careful distinction may be necessary among these configurations. However, to avoid making our study sensitive to a specific cost-model, in the remaining part of the paper we do not make such a distinction.

B. Deriving Good Configurations

From among the design-feasible configurations presented in the left half of Table V we can derive a set of good configurations. Let us assume a demanded performance of $T_{max} = 200$ cycles, $\lambda = 3.0$ bits/cycle, and a message length of $L = 192$ bits. This leads to $\lambda/L = 0.015$ messages/cycle. For each configuration in Table V, the value of maximum throughput sustainable (m_{max}) while maintaining average message latency less than $T_{max} \leq 200$ cycles is shown next to it. These values are derived using Eqn. 13. Some of the values for very low radix systems were derived through simulation experiments. This is because Agrawal's model [2], on which our model is based, does not hold for very low radix configurations. The performance plots depicting the average message latency versus average throughput for all feasible configurations to build a $N \approx 1024$ processor system are shown in Fig. 10. These were obtained through actual simulation experiments. These plots demonstrate similar comparative trends between configurations as derived by the analytical model. The configurations in Table V offering $m_{max} \geq \lambda/L$ are good configurations and depicted in boldface. For example, the configurations 4D c-2, 4D c-3, 5D c-3, and 5D c-4 were derived to be good.

TABLE V

DESIGN FEASIBLE CONFIGURATIONS TO BUILD SYSTEMS WITH $N \approx 1024$ AND 4096 PROCESSORS UNDER THE PACKAGING PARAMETERS $p_p = 128$, $W' = 24$, $R = 250$, AND $b_{max} = 8$. GOOD CONFIGURATIONS ARE SHOWN IN BOLDFACE AND THESE OFFER $m_{max} \geq \lambda/L = 0.015$ WITH $T_{max} = 200$ CYCLES.

Internet topology	cluster size (c)	Maximum Average Throughput (m_{max})
$N = 1024$		
3D: 8x8x8	2	0.0100
4D: 5x5x5x4	2	**0.019**
4D: 5x4x4x4	3	**0.015**
4D: 4x4x4x4	4	0.013
5D: 4x3x3x3x3	3	**0.024**
5D: 3x3x3x3x3	4	**0.018**
$N = 4096$		
3D: 13x13x12	2	0.006
4D: 7x7x7x6	2	0.012
4D: 6x6x6x6	3	0.009
4D: 6x6x6x5	4	0.007
5D: 5x4x4x4x4	3	**0.015**
5D: 4x4x4x4x4	4	0.012

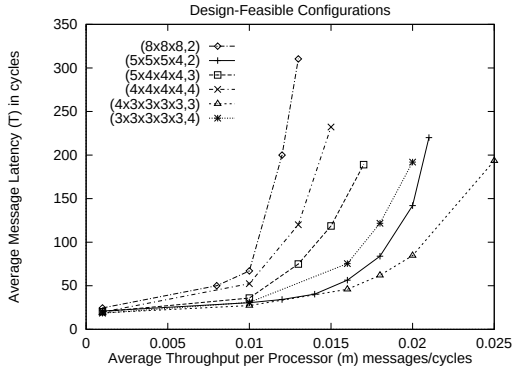


Fig. 10. Comparing the performance, obtained through simulation, of the design-feasible configurations shown in Table V to build a system with $N \approx 1024$ processors.

C. Deriving the Best Configuration

Let us first consider the best configuration derived based on selecting the good configuration offering the maximum value of m_{max}. From Table V this best configuration is 5D c-3 $(4 \times 3^4, 3)$ with a $m_{max} = 0.024$ messages/cycle. Next let us analyze the best configuration with potential to scale up to a system size of $N \approx 4096$ processors.

The design-feasible configurations and the good configurations were also derived for a larger system size with $N \approx 4096$ processors for a similar set of packaging and performance parameters. These are shown in the bottom half of Table V. For a set of packaging parameters the set of design-feasible configurations (in terms of n and c) are similar irrespective of size of the system being designed. However, the size of the inter-cluster needs to be larger to accommodate more clusters. For example, consider the 3D c-2 $(8 \times 8 \times 8, 2)$ feasible configuration, shown in first row of Table V to build a $N \approx 1024$ system. This configuration needs to be scaled to a larger 3D c-2 $(13 \times 13 \times 12, 2)$ feasible configuration to build a system with $N \approx 4096$ processors. However, from Result 3 we know that the inter-cluster bisection size per processor does not scale linearly with system size. This is also reflected in a fall in the value of supportable throughput m_{max}, derived using Eqn. 13, from 0.010 to 0.006 messages/cycle as shown in Table V. Similarly, for other configurations a fall in the offered value for m_{max} was observed with increasing system size. It can be observed that only one configuration, 5D c-3, is capable of meeting the demanded performance of $\lambda/L = 0.015$ messages/cycle to build a $N \approx 4096$ processor system. This leads us to conclude that 5D c-3 $(5 \times 4^4, 3)$ is the only good configuration to build a system with 4K processors. The intersection of this set with the set of good configurations for a system with 1K processors again leads us to the 5D c-3 as the best configuration. This configuration has potential to scale up to 4K processors while still offering the minimal demanded performance.

Similar to the above illustration our framework can be applied to derive the best configuration under different sets of packaging and demand parameters. Considering the above example as a base case we also studied the impact of varying different packaging and demand parameters on design. Further results on best configurations are derived by choosing the good configuration offering maximum value of m_{max}.

X. IMPACT OF VARYING PACKAGING AND DEMAND PARAMETERS ON THE DESIGN PROCESS

In this section, we illustrate the impact of changing various packaging and demand parameters on the set of design-feasible and good configurations. In all following tables the derived good configurations are shown in boldface and the best configuration shown preceded by a + sign.

A. Processor and Interconnect Technology

For a given set of packaging technologies the set of design-feasible solutions is fixed. However, the set of good configurations is dependent on the demanded values of T_{max} and λ. In Section III-A it was indicated that the value of λ depends on the relative advancements in processor and interconnect technologies. We analyzed the impact of varying λ on the set of good configurations. Table VI shows the same design-feasible configurations as shown in Table V to build a system with $N \approx 1024$ processors. The top half of Table VI shows the good configurations (depicted in boldface) for a lower value of $\lambda = 2.0$, corresponding to $\lambda/L = 0.010$ messages/cycle. In this example it was observed that all design-feasible configurations become good. The bottom half of Table VI similarly shows the good configurations obtained with a higher value of $\lambda = 4.0$, corresponding to $\lambda/L = 0.020$ messages/cycle. In this case only the 5D c-3 and 5D c-4 configurations were derived to be good. Under both values of λ the best configuration, based on maximum value of offered m_{max}, is 5D c-3. This best configuration is depicted in Table VI preceded by a + sign.

TABLE VI
IMPACT OF VARYING DEMANDED THROUGHPUT (λ) ON THE DESIGN OF A SYSTEM WITH $N \approx 1024$ PROCESSORS. PARAMETERS $p_p = 128$, $W' = 24$, $R = 250$, AND $b_{max} = 8$ ARE ASSUMED.

Internet topology	cluster size (c)	Maximum Average Throughput (m_{max})
$\lambda = 2.0$		
3D: 8x8x8	2	**0.010**
4D: 5x5x5x4	2	**0.019**
4D: 5x4x4x4	3	**0.015**
4D: 4x4x4x4	4	**0.013**
+ 5D: 4x3x3x3x3	3	**0.024**
5D: 3x3x3x3x3	4	**0.018**
$\lambda = 4.0$		
3D: 8x8x8	2	0.010
4D: 5x5x5x4	2	0.019
4D: 5x4x4x4	3	0.015
4D: 4x4x4x4	4	0.013
+ 5D: 4x3x3x3x3	3	**0.024**
5D: 3x3x3x3x3	4	**0.018**

B. Increasing Supported Channel Width

We studied the impact of wider channel width of $W' = 40$ while designing a system with $N \approx 1024$ processors. The other parameters were maintained at $p_p = 128$, $R = 250$, and $b_{max} = 8$. The resultant feasible configurations are shown in Table VII. The performance plots depicting the average message latency versus average throughput for these configurations obtained through actual simulation are shown in Figure 11. The simulation results again conform to the latency-throughput trends indicated by the analytical model. Let us compare these configurations with those obtained in Table V for a similar design problem with $W' = 24$. It can be observed that higher dimensional

systems ($n = 4, 5$) feasible earlier under $W' = 24$ are no longer feasible under $W' = 40$. Similarly, for a given inter-cluster dimension under both channel widths, the cluster size is larger with $W' = 40$. For example, the configuration 3D c-2 is feasible with $W' = 24$ while configurations with larger cluster sizes, 3D c-3 and 3D c-4, become feasible with $W' = 40$. These results confirm the observations in Result 1. The best configuration in this case was derived as $(7 \times 7 \times 7, 3)$.

TABLE VII

IMPACT OF INCREASED CHANNEL WIDTH TECHNOLOGY ($W' = 40$) ON THE DESIGN OF A SYSTEM WITH $N \approx 1024$ PROCESSORS. PARAMETERS $p_p = 128$, $R = 250$, AND $b_{max} = 8$ ARE ASSUMED.

Internet topology	cluster size (c)	Maximum Average Throughput (m_{max})
$W' = 40$		
2D: 32x32	1	0.010
2D: 23x22	2	0.007
+ 3D: 7x7x7	3	**0.017**
3D: 7x6x6	4	**0.015**

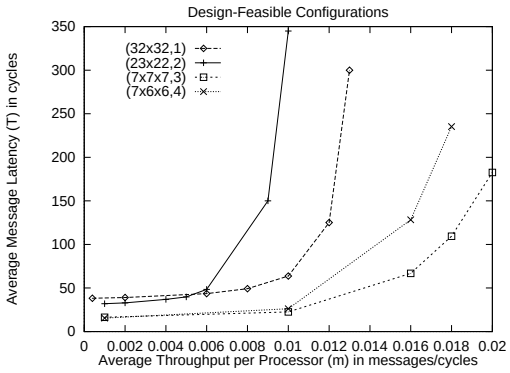


Fig. 11. Comparing the performance, obtained through simulation, of the design-feasible configurations shown in Table VII to build a system with $N \approx 1024$ processors.

C. Increasing Board Pinout Density

As observed in Table III in Section VI-B, an increase in board pinout density makes higher dimensional inter-cluster networks feasible. Similarly, smaller cluster sizes also become feasible with fixed inter-cluster dimensionality. These impacts of increasing pinout density to $p_p = 192$ and 256 were analyzed while designing a system with $N \approx 1024$ processors. The other parameters were maintained at values used in Sec. IX: $R = 250$, $W' = 24$, and $b_{max} = 8$. The resultant feasible configurations obtained are shown in Table VIII. We observed 3D c-2 to be a feasible configuration under $p_p = 128$ in Table V. With increasing pinout density, $p_p = 192$, the feasibility set shifted to support a smaller cluster size, a 3D c-1 configuration, as shown in the first row of Table VIII. With even higher pinout density, $p_p = 256$, a 3D topology cannot utilize the increased pinout effectively because the cluster size can not be smaller than one. Thus, a 3D configuration is no longer feasible and hence not depicted in Table VIII. Although higher dimensional ($n > 5$) systems become supportable with increased pinout density, the maximum router pinout ($R = 250$) constrains systems with $n > 5$ from being feasible. The best configurations derived with $p_p = 192$ and 256 were $(6 \times 6 \times 6 \times 5, 1)$ and $(4 \times 4 \times 4 \times 4 \times 4, 1)$, respectively.

TABLE VIII

IMPACT OF INCREASING PINOUT (p_p) ON THE DESIGN OF A SYSTEM WITH $N \approx 1024$ PROCESSORS. PARAMETERS $W' = 24$, $R = 250$, AND $b_{max} = 8$ ARE ASSUMED.

Internet topology	cluster size (c)	Maximum Average Throughput (m_{max})
$p_p = 192$		
3D: 10x10x10	1	**0.015**
+ 4D: 6x6x6x5	1	**0.029**
4D: 5x5x5x4	2	**0.019**
5D: 4x3x3x3x3	3	**0.024**
$p_p = 256$		
4D: 6x6x6x5	1	**0.029**
+ 5D: 4x4x4x4x4	1	**0.049**
5D: 4x4x4x3x3	2	**0.029**

D. Increasing the Maximum Router Pinout

The impact of increasing maximum router pinout to $R = 500$ was observed in designing a system with $N \approx 1024$ processors. The other parameters were maintained at $p_p = 128$, $W' = 40$, and $b_{max} = 8$. The resultant feasible configurations are shown in Table IX. The feasible configurations derived for a similar design problem with $R = 250$ were presented in Table VII. By comparing Tables VII and IX, it can be observed that the maximum dimensionality of feasible configurations increased from $n = 3$ to 5 as R was raised from 250 to 500. Larger dimensional systems were also associated with larger cluster sizes: 4D configurations with $c = 6$ and 7 and 5D configuration with $c = 8$. The best configuration derived with $R = 500$ was $(3 \times 3 \times 3 \times 2 \times 2, 8)$.

TABLE IX

IMPACT OF INCREASED ROUTER PINOUT ($R = 500$) ON THE DESIGN OF A SYSTEM WITH $N \approx 1024$ PROCESSORS. PARAMETERS $p_p = 128$, $W' = 40$, AND $b_{max} = 8$ ARE ASSUMED.

Internet topology	cluster size (c)	Maximum Average Throughput (m_{max})
$R = 500$		
2D: 32x32	1	0.010
2D: 23x22	2	0.007
3D: 7x7x7	3	**0.017**
3D: 7x6x6	4	**0.015**
4D: 4x4x4x3	6	**0.019**
4D: 4x4x3x3	7	**0.018**
+ 5D: 3x3x3x2x2	8	**0.027**

E. Summary of results

The impact of varying other packaging constraints like maximum board size and surface board pinout technology was also considered. These results are available in [6]. Based on representative current and expected future technologies our analysis indicated that flat configurations may not be design-feasible under all packaging technologies. On the other hand, clustered configurations demonstrate higher potential in offering design-feasible configurations. For a wide range of technological parameters, we observed that best configurations are achieved with up to 8 processors per cluster and 3D-5D inter-cluster interconnection.

XI. 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 framework explores the design space of flat k-ary n-cube topologies and their clustered variations (k-ary n-cube cluster-c) to derive design-feasible/best configurations. The elegance of this framework lies in its parameterized representation of different technologies and then deriving best 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 this framework lies in its generality which has never been taken into consideration by earlier researchers. All proposed earlier works have centered around fixed technology and packaging constraints. This generalized framework can be applied to a wide variety of technological parameters and constraints for years to come.

Using this framework, the following design guidelines have been obtained for building large cluster-based multiprocessor systems with k-ary n-cube cluster-c organizations:

1. A two-level clustered architecture widens the design space, in terms of alternative configurations possible, to build a system with a given number of processors.
2. Flat configurations may not be design-feasible under all packaging technologies. On the other hand, clustered configurations demonstrate higher potential in offering design-feasible configurations. This indicates that clustered systems would be a dominant trend in building future systems.
3. For a wide range of technological parameters, it is shown that best configurations are achieved with up to 8 processors per cluster and 3D-5D inter-cluster interconnection.

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.

Additional Information: A number of related papers and technical reports are available electronically through the home page of *Parallel Architecture and Communication* (PAC) research group. The URL is `http://www.cis.ohio-state.edu/~panda/pac.html`.

Acknowledgments

The authors are grateful to the anonymous referees for their valuable comments which have helped us in improving the quality of this paper. This research is supported in part by the National Science Foundation Grant MIP–9309627, Career Award MIP–9502294, CDA–9413962, and an Ohio State University Presidential Fellowship.

References

[1] S. Abraham and K. Padmanabhan. Constraint based evaluation of multicomputer networks. In *Proc. of the Int. Conf. on Parallel Processing*, pages 521–525, Aug 1990.

[2] A. Agrawal. Limits on Interconnection Network Performance. *IEEE Trans. on Parallel and Distributed Systems*, 2(4):398–412, 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*, pages 323–327, 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 Processor, 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 Clustered Multiprocessor Systems under Packaging and Technological Advancements. Technical Report OSU-CISRC-11/95-TR51, Department of Computer and Information Science, The Ohio State University, 1995.

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

[8] W. J. Dally. Performance Analysis of k-ary n-cube Interconnection Networks. *IEEE Trans. on Computers*, 39(6):775–785, June 1990.

[9] S. Dandamudi and D. Eager. Hierarchical Interconnection Networks for Multicomputer Systems. *IEEE Trans. on Computers*, C-39(6):786–797, June 1990.

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

[11] W. Hsu and P. C. Yew. The Performance of Hierarchical Systems with Wiring Constraints. In *Proc. of the Int. Conf. on Parallel Processing*, pages 9–16, Aug 1991.

[12] W. Hsu and P. C. Yew. The Impact of Wiring Constraints on Hierarchical Network Performance. In *Proc. of the Int. Parallel Processing Symposium*, pages 580–588, Mar 1992.

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

[14] 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.

[15] K. Padmanabhan. Efficient Architectures for Data Access in a Shared Memory Hierarchy. *Jour. of Parallel and Distributed Computing*, 11:314–327, 1991.

[16] 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*, pages 5–10, 1994.

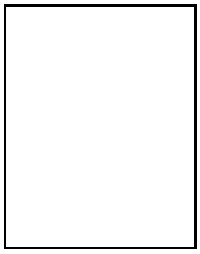
[17] D. A. Patterson. Observations in Massive Parallelism - Trends and Predictions for 1995 to 2000. Technical Report 93-87, DIMACS, Sept 1993.

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

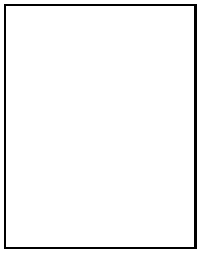
[19] M. T. Raghunath and A. Ranade. Designing interconnection networks for multi-level packaging. In *Proc. of the Supercomputing*, pages 772–781, 1993.

[20] 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.

[21] A. A. Sawchuk, B. K. Jenkins, C. S. Raghavendra, and A. Varma. Optical Crossbar Networks. *IEEE Computer*, pages 50–62, 1987.
[22] S. L. Scott and J. R. Goodman. The impact of pipelined channels on k-ary n-cube networks. *IEEE Trans. on Parallel and Distributed Systems*, pages 2–16, Jan 1994.
[23] Seraphim, Lasky, and Li Eds. *Principles of Electronic Packaging*. McGraw Hill, 1989.
[24] Tummala and Rymaszewski. *Microelectronics Packaging Handbook*. Van Nostrand Reinhold, 1989.



Debashis Basak received the B.Tech degree in Computer Science and Engineering from the Indian Institute of Technology, N. Delhi, India, in 1991 and the M.S. degree in Computer Science from The Ohio State University, Columbus, Ohio, in 1992. He is currently a Ph.D. student at the Department of Computer and Information Science, The Ohio State University, Columbus, Ohio. His research interests include parallel architectures, inter-processor communication and synchronization, interconnection network design, high performance computing, ATM switch design, and network of workstations. He is a member of the IEEE Computer Society and the Association for Computing Machinery.



Dhabaleswar K. Panda (S'86-M'92) received the B.Tech degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1984, the M.E. in electrical and communication engineering from the Indian Institute of Science, Bangalore, India, in 1986, and the Ph.D. in computer engineering from the University of Southern California, USA, in 1991. Since September 1991, he has been an Assistant Professor in the Department of Computer and Information Science, The Ohio State University, Columbus, USA. His research interests include parallel computer architecture, wormhole-routing, interprocessor communication, synchronization, clustered and heterogeneous parallel systems, mapping and scheduling, and high-performance computing.

Dr. Panda has served on Program Committees of the International Parallel Processing Symposium, International Conference on Parallel Processing, International Conference on High Performance Computing, and IEEE International Conference on Distributed Computing Systems. He also serves on the Executive Committee of the IEEE Technical Committee on Parallel Processing. He is a 1995 recipient of the NSF Faculty Early Career Development Award. Dr. Panda is a member of the IEEE Computer Society and the Association for Computing Machinery.