A k-ary n-cube direct network can be modeled in a similar fashion [2] by replacing the expression for T in Equation 1 with,

$$T = \left[1 + \frac{\rho B \frac{1}{k_d} \left(1 - \frac{1}{k_d}\right) \left(1 + \frac{1}{n}\right)}{(1 - \rho)} \right] n k_d + B + M - 1 \qquad (3)$$

where k is the network radix, n is the dimension, and k_d is the average distance a message travels in a dimension. For a torus with channels in both directions, $k_d = k/4$.

B Description of Tmul-T Trace File Format

The trace file is a sequence of 6-byte packets. The first byte is the processor number and the second byte is an opcode to be described below. The next four bytes are generally the longword address but this depends on the particular opcode. In writing programs to filter the trace note that packets follow the VAX byte order (they are little-endian). The first seven opcodes described in the table are always in the first seven packets. Opcode descriptions:

32	The address is the beginning of the static read-only area. (.text)
33	The address is the end of the static read-only area.
34	The address is the beginning of the static writable area. (.data)
35	The address is the end of the static writable area.
36	The address is the beginning of the heap.
37	The address is the end of the heap. Note that the heap is equally divided among the processors from 1 to n.
38	The address is the number of processors, n.
0	The address was read.
1	The address was written.
2	The address was read and written, e.g. add to memory.
3	The address was test and set. (synchronization)
8	The address was read from the stack.
9	The address was written to the stack.
10	The address was read and written to the stack.
16	The task blocked on a future.
17	A task that blocked was restarted.
18	A new future started running.
19	The processor became idle (looking for work).
20	A task started determining its future.
21	A task finished determining its future.
22	A future started to be created.
23	The creation of the future has completed.

Any address outside the ranges indicated by codes 32 through 37 belongs to an object created when the program was loaded or initialized. Codes 16 through 23 indicate actions by the scheduler regarding task scheduling. The address field of the packet is ignored for these codes.

[20] Richard L. Sites and Anant Agarwal. Multiprocessor Cache Analysis using ATUM. In *Proceedings of the 15th International Symposium on Computer Architecture*, pages 186–195, IEEE, New York, June 1988.

[21] K. So, F. Darema-Rogers, D. A. George, V. A. Norton, and G. F. Pfister. *PSIMUL - A System for Parallel Simulation of Parallel Systems.* Technical Report RC 11674 (58502), IBM T. J. Watson Research Center, Yorktown Heights, November 1987.

[22] Craig B. Stunkel and W. Kent Fuchs. TRAPEDS: Producing Traces for Multicomputers Via Execution Driven Simulation. In *Proceedings of ACM SIGMETRICS 1989*, May 1989.

A Performance Models for Interconnection Networks

Hybrid, decoupled trace-driven simulation schemes can use the following network models to estimate network latency. We first present a simple packet-switched, buffered, multistage interconnection network model. The network switches (of size $k \times k$) are pipelined so a message header can leave a switch even while the rest of the message is still being serviced. A network request travels through n switch stages to the destination node and takes M cycles for the memory access. The response to the request (or the acknowledgment in some cases) is inserted into the network by the destination node. Let us further assume that the processor idles from the time that its network request is serviced until it receives a response.

Computation of the processor utilization with multistage networks is based on the network model proposed by Kruskal and Snir [15]. Using the usual assumptions of independent requests uniformly distributed over all the memory modules, the network model yields the average latency T of a memory request through the network. In the equation below, ρ, the channel utilization, is the product of the effective network request rate m and the average message size B.

$$T = \left[1 + \frac{\rho B \left(1 - \frac{1}{k}\right)}{2(1-\rho)}\right] n + B + M - 1 \tag{1}$$

The latency is n times the delay through a switch (note that acknowledgments are included in the request rate), plus the memory delay and the message pipeline delay. The switch delay is one plus contention delay. We can now compute processor utilization U using the following set of equations:

$$\begin{aligned} U &= \frac{1}{1+mT} \\ \rho &= UmB \\ T &= \left(1 + \frac{\rho B \left(1 - \frac{1}{k}\right)}{2(1-\rho)}\right) n + B + M - 1 \end{aligned}$$

The above equations can be solved using standard numerical methods. A closed form solution for U can also be derived as:

$$U = \frac{1}{1 + \frac{m}{2}\left[n + M + 2B - 1 - \frac{1}{m} + \sqrt{(n + M - 1 + \frac{1}{m})^2 + 2nB^2\left(1 - \frac{1}{k}\right)}\right]} \tag{2}$$

[6] David Chaiken, John Kubiatowicz, and Anant Agarwal. LimitLESS Directories: A Scalable Cache Coherence Scheme. August 1990. Laboratory for Computer Science, M.I.T. MIT VLSI Memo. Submitted for publication.

[7] Mathews Cherian. *A Study of Backoff Barrier Synchronization in Shared-Memory Multiprocessors.* Technical Report, S.M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1989.

[8] Helen Davis and John Hennessy. Characterizing the synchronization behavior of parallel programs. In *Proceedings of the ACM/SIGPLAN PPEALS Parallel Programming: Experience with Applications, Languages and Systems*, ACM, July 1988. Published as Vol. 23, No. 9, of SIGPLAN Notices.

[9] Susan J. Eggers, David R. Keppel, Eric K. Koldinger, and Henry M. Levy. Techniques for Efficient Inline Tracing on a Shared-Memory Multiprocessor. In *Proceedings of ACM SIGMETRICS 1990*, May 1990.

[10] Robert H. Halstead. Multilisp: A Language for Parallel Symbolic Computation. *ACM Transactions on Programming Languages and Systems*, 7(4):501–539, October 1985.

[11] Robert R. Henry. Tracer - Address and Instruction Tracing for the VAX Architecture. University of California, Berkeley, November, 1984.

[12] Mark Holliday. Trace-Driven Simulation of Distributed Shared Memory Environments. June 1990. The 17th Annual International Symposium on Computer Architecture, Workshop III: Scalable Shared-Memory Architectures.

[13] D. Kranz, R. Halstead, and E. Mohr. Mul-T: A High-Performance Parallel Lisp. In *Proceedings of SIGPLAN '89, Symposium on Programming Languages Design and Implemenation*, June 1989.

[14] D. Kranz et al. ORBIT: An Optimizing Compiler for Scheme. In *Proceedings of SIGPLAN '86, Symposium on Compiler Construction*, June 1986.

[15] Clyde P. Kruskal and Marc Snir. The Performance of Multistage Interconnection Networks for Multiprocessors. *IEEE Transactions on Computers*, C-32(12):1091–1098, December 1983.

[16] Manoj Kumar and Kimming So. Trace Driven Simulation for Studying MIMD Parallel Computers. In *International Conference on Parallel Computing*, pages I–68 – I–72, 1989.

[17] Kiyoshi Kurihara. *Performance Evaluation of Large-Scale Multiprocessors.* Technical Report, S.M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 1990.

[18] J. Rees, N. Adams, and J. Meehan. *The T Manual, Fourth Edition.* Technical Report, Yale University, Computer Science Department, January 1984.

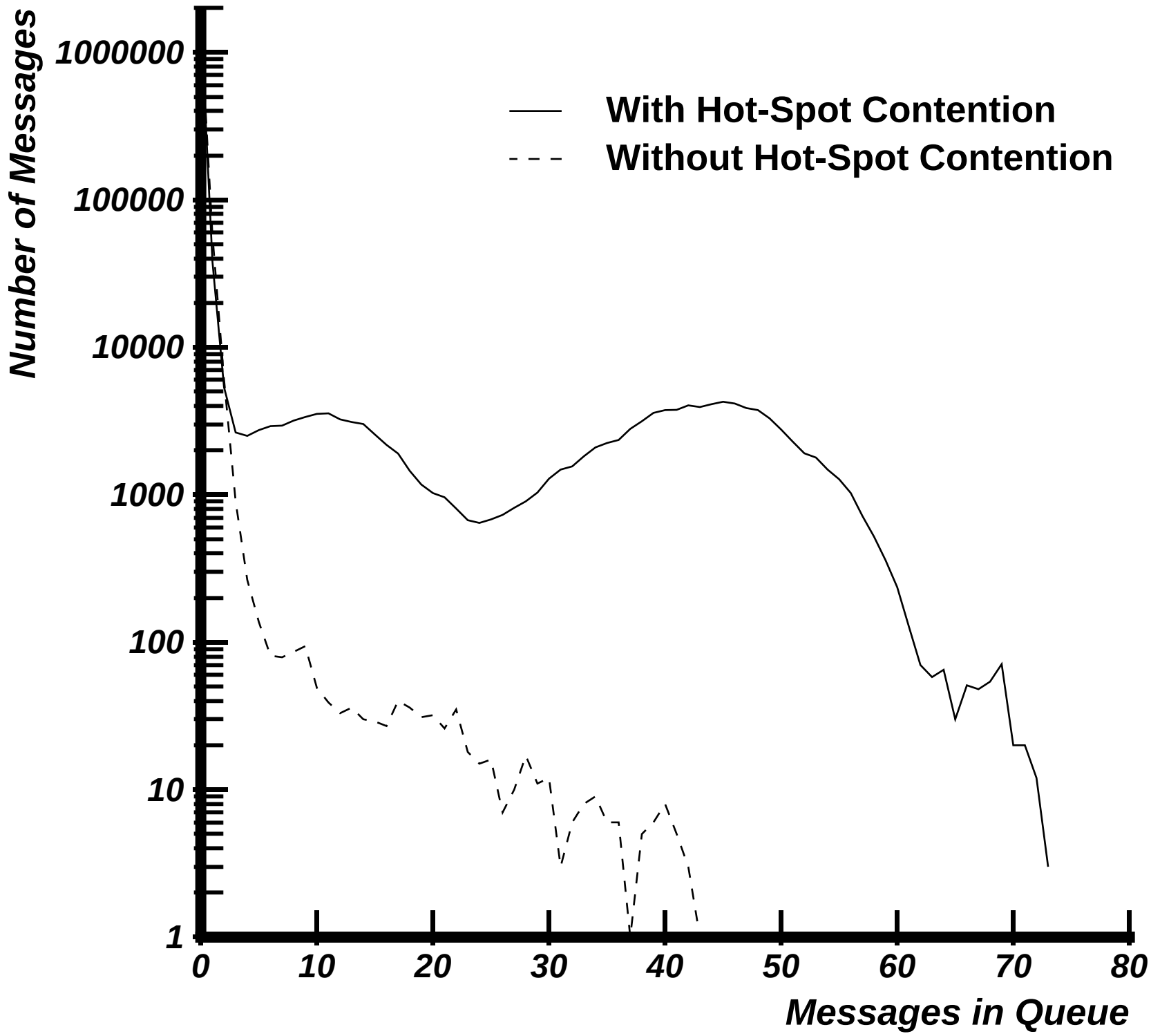
[19] Arturo Salz. VTRACE. 1984. Computer Systems Laboratory, Stanford University.

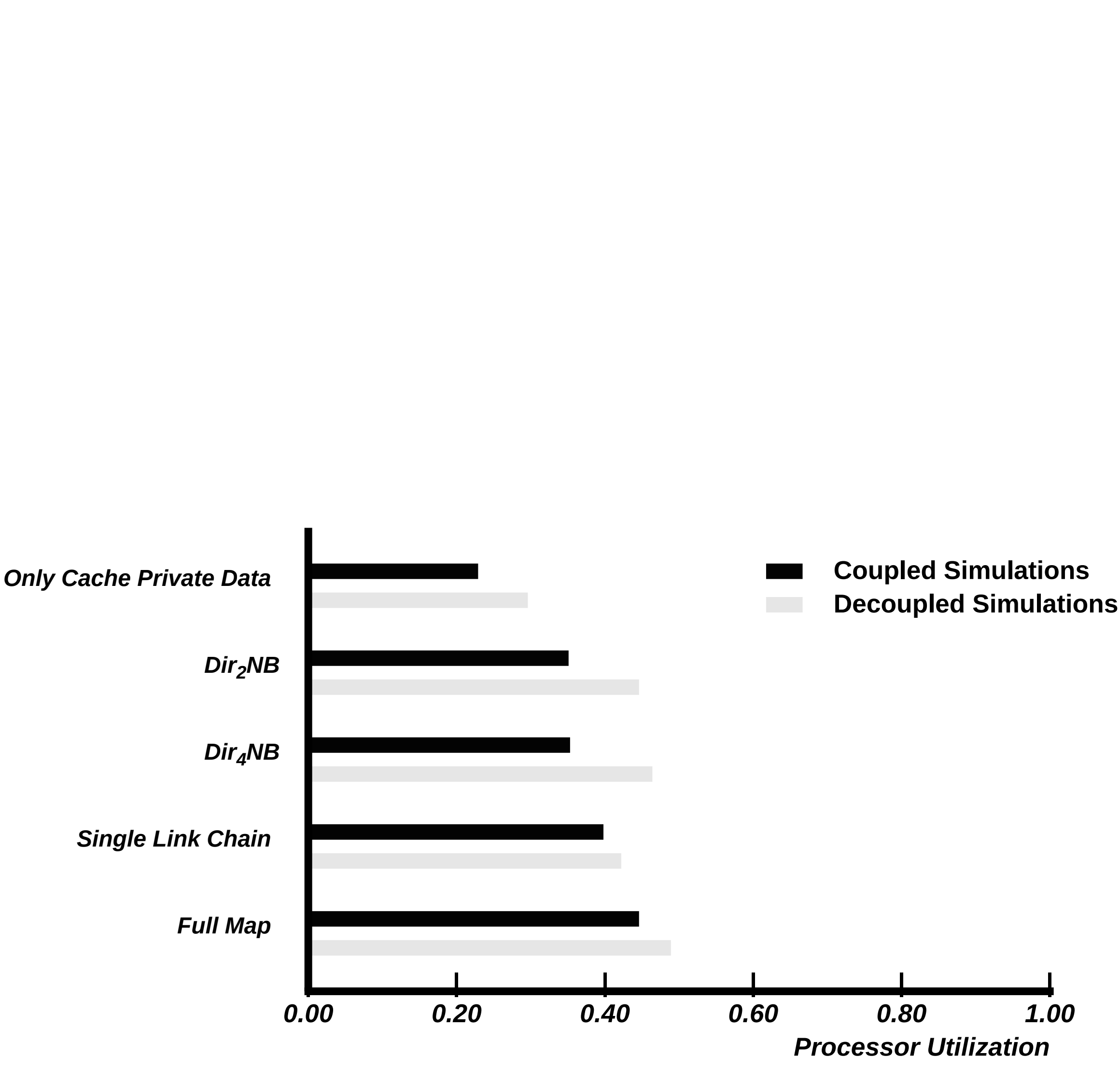
cience MIT MIT I Memo umitted for pulication

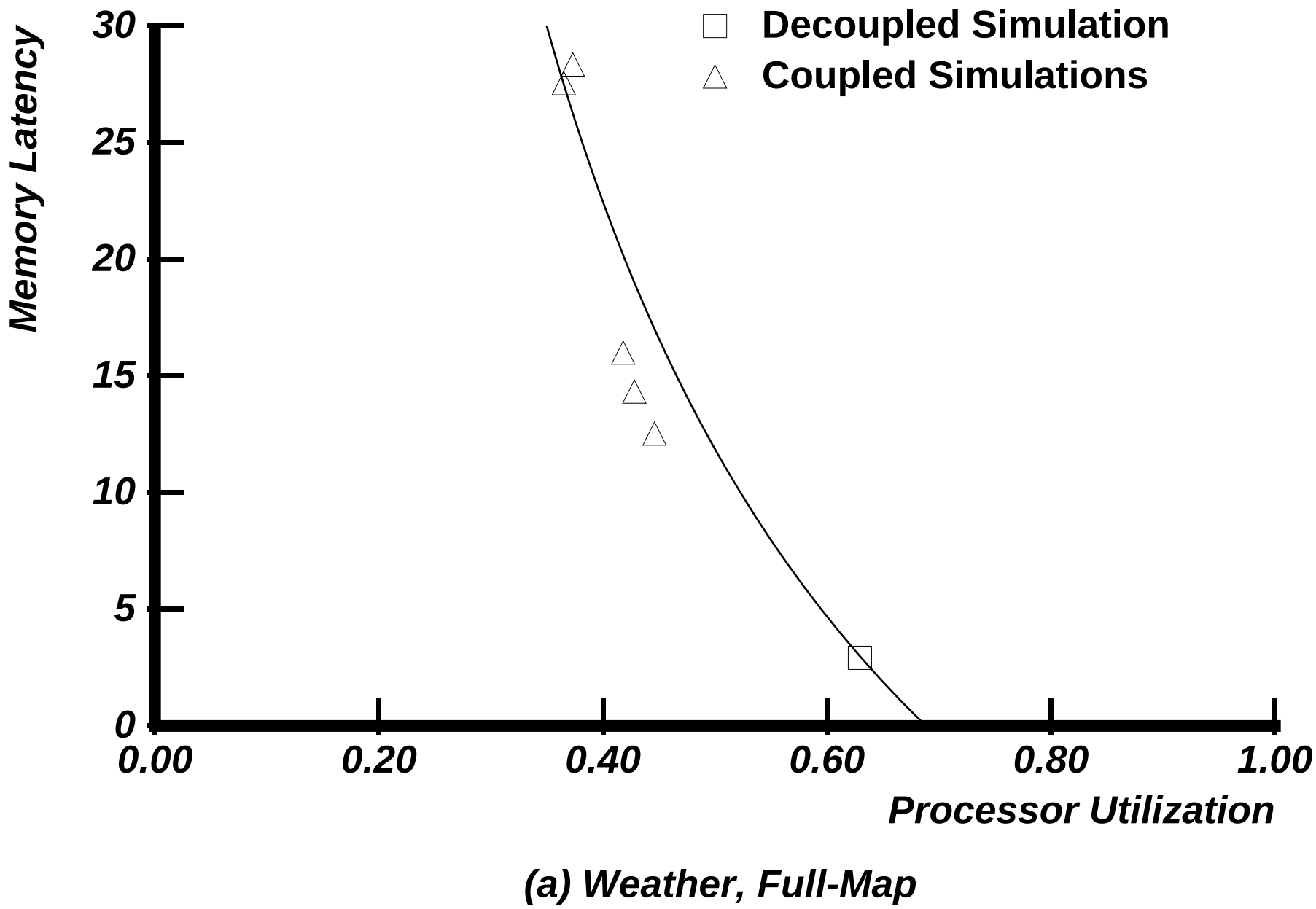
Anant Aarwal icard ites and Mark orowit ATM A New Tecniue for apturin Address Traces sin Microcode In *Proceedings of the 1th Annual Symposium on Computer Architecture*, pages 119–127, IEEE, New York, June 1986.

[4] Laura Bagnall. *Paris: A Program isualization Tool for Multilisp.* Technical Report, S.M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, February 1989.

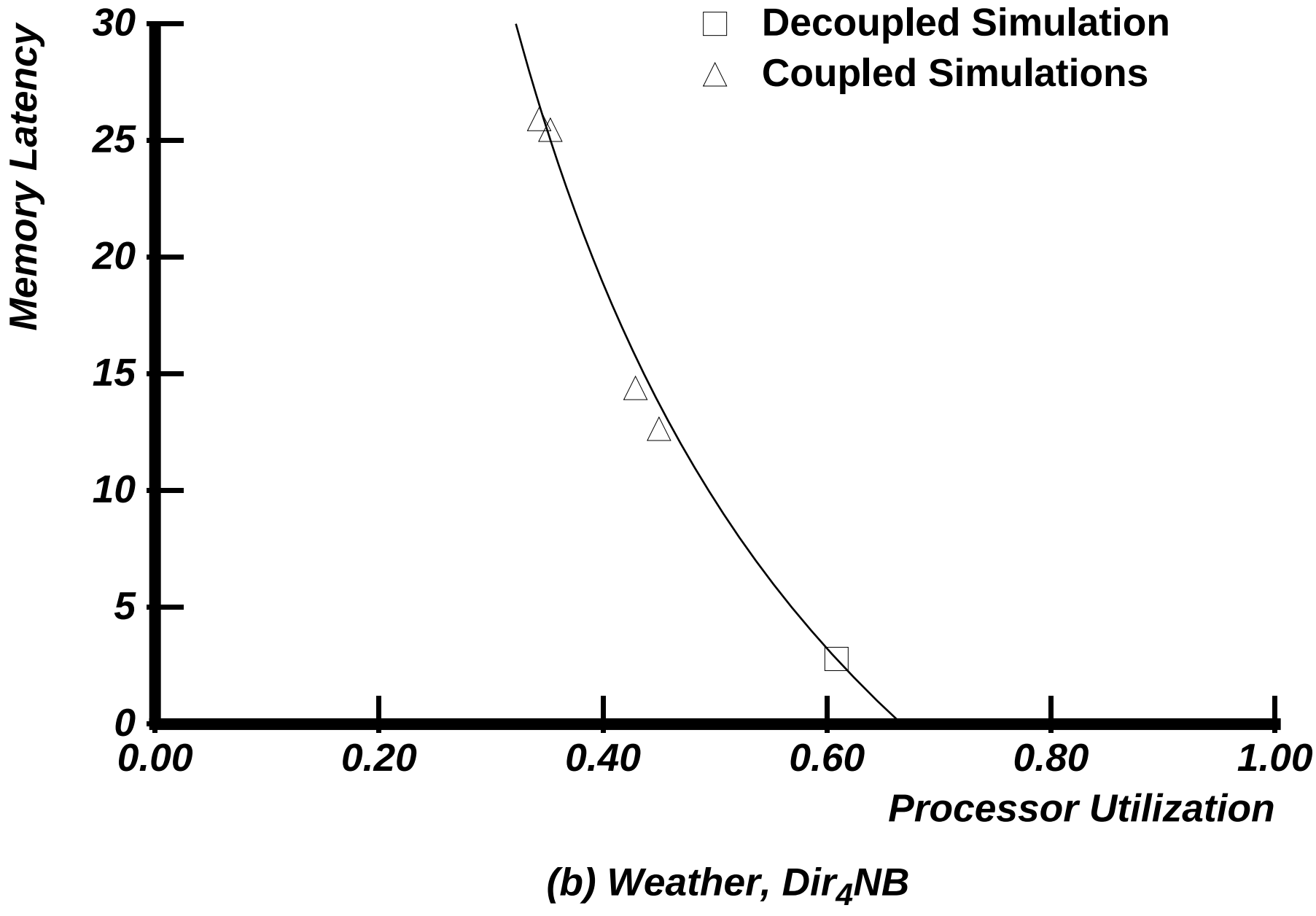
[5] David Chaiken, Craig Fields, Kiyoshi Kurihara, and Anant Agarwal. Directory-Based Cache-Coherence in Large-Scale Multiprocessors. *IEEE Computer*, June 1990.


Number of Messages
1000000
100000
10000
1000
100
10
1
0
10
20
30
40
50
60
70
80
Messages in Queue
With Hot-Spot Contention
Without Hot-Spot Contention

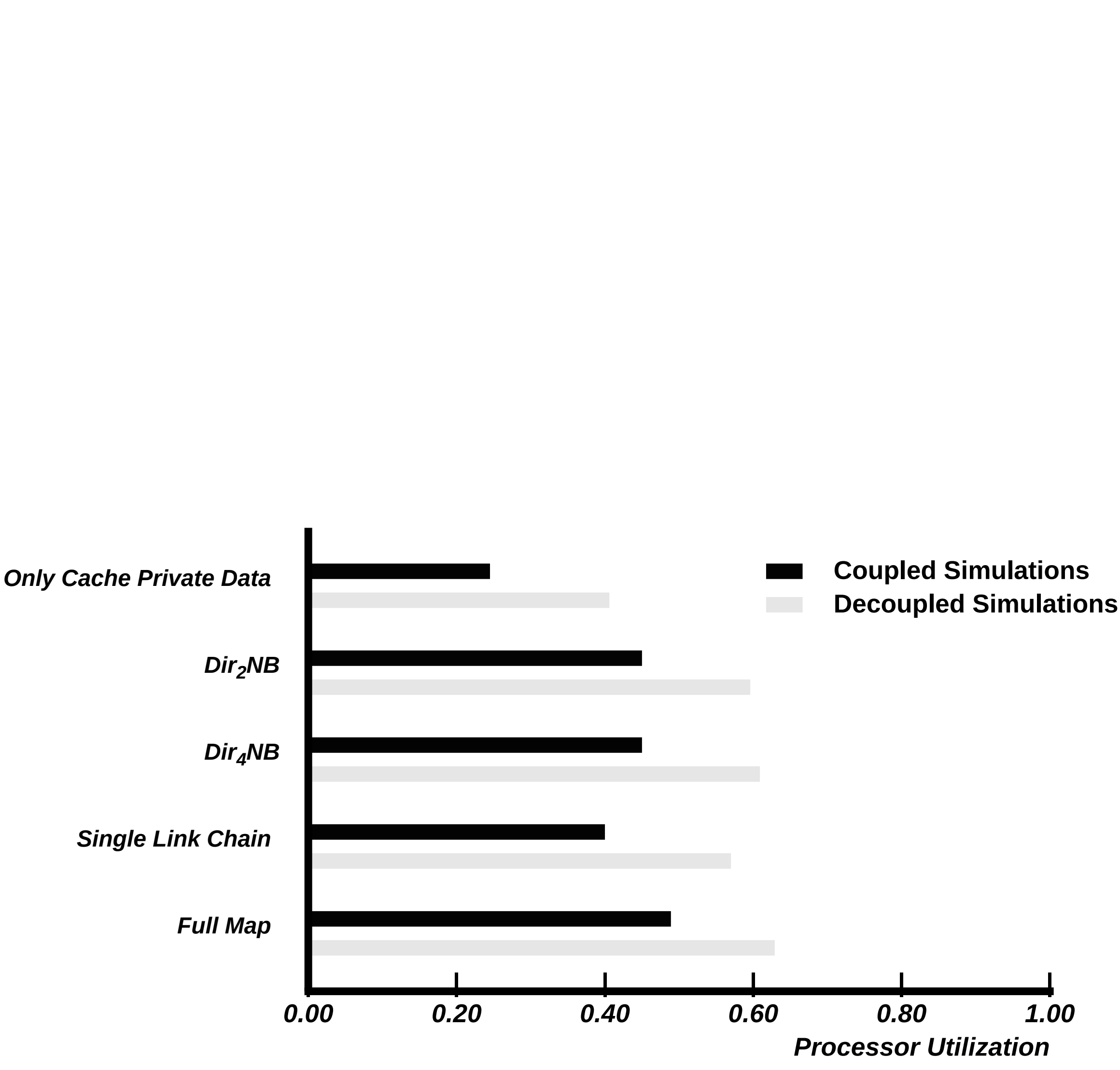

Only Cache Private Data
Dir2NB
Dir4NB
Single Link Chain
Full Map
Coupled Simulations
Decoupled Simulations
0.00
0.20
0.40
0.60
0.80
1.00
Processor Utilization

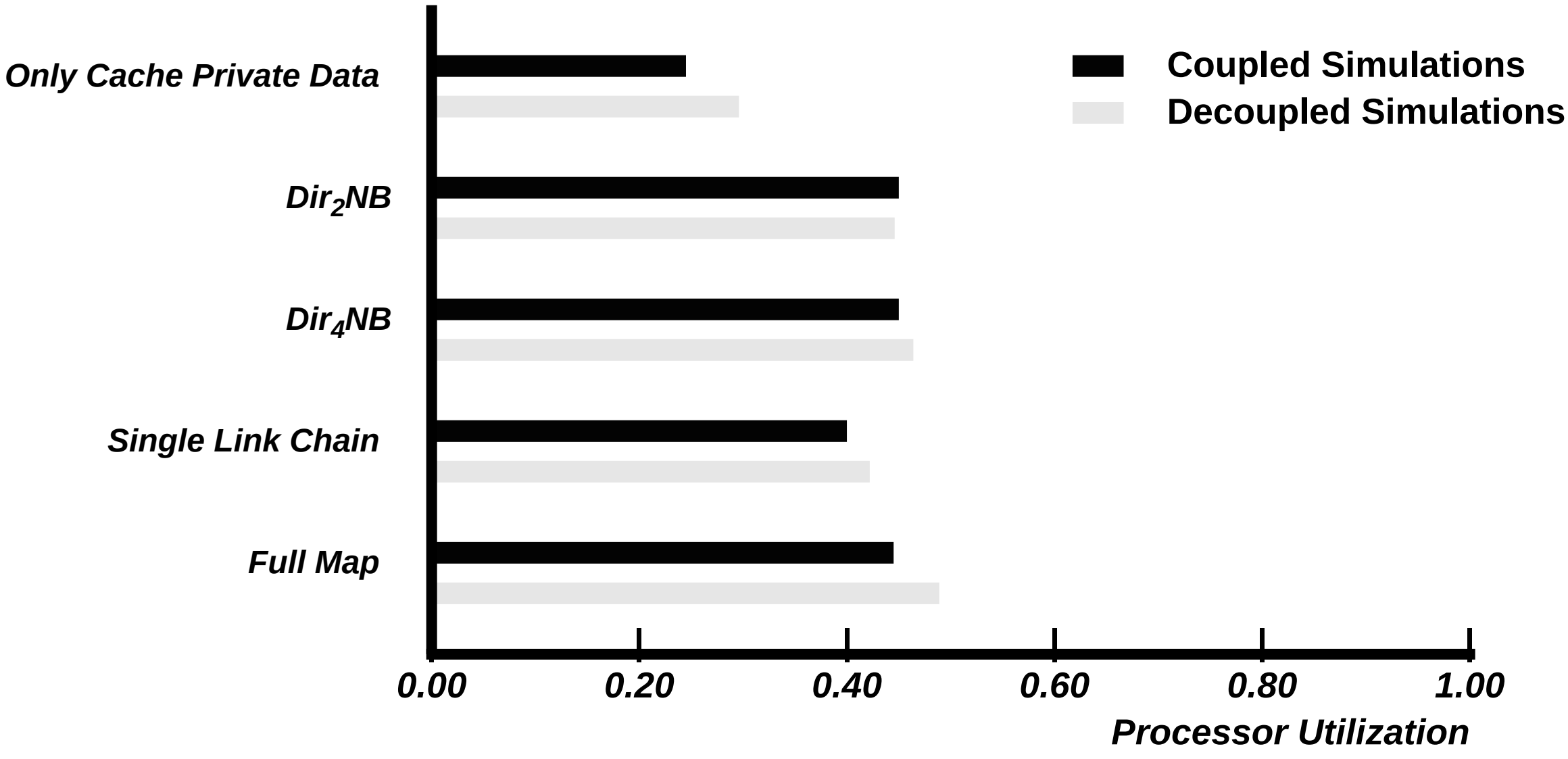


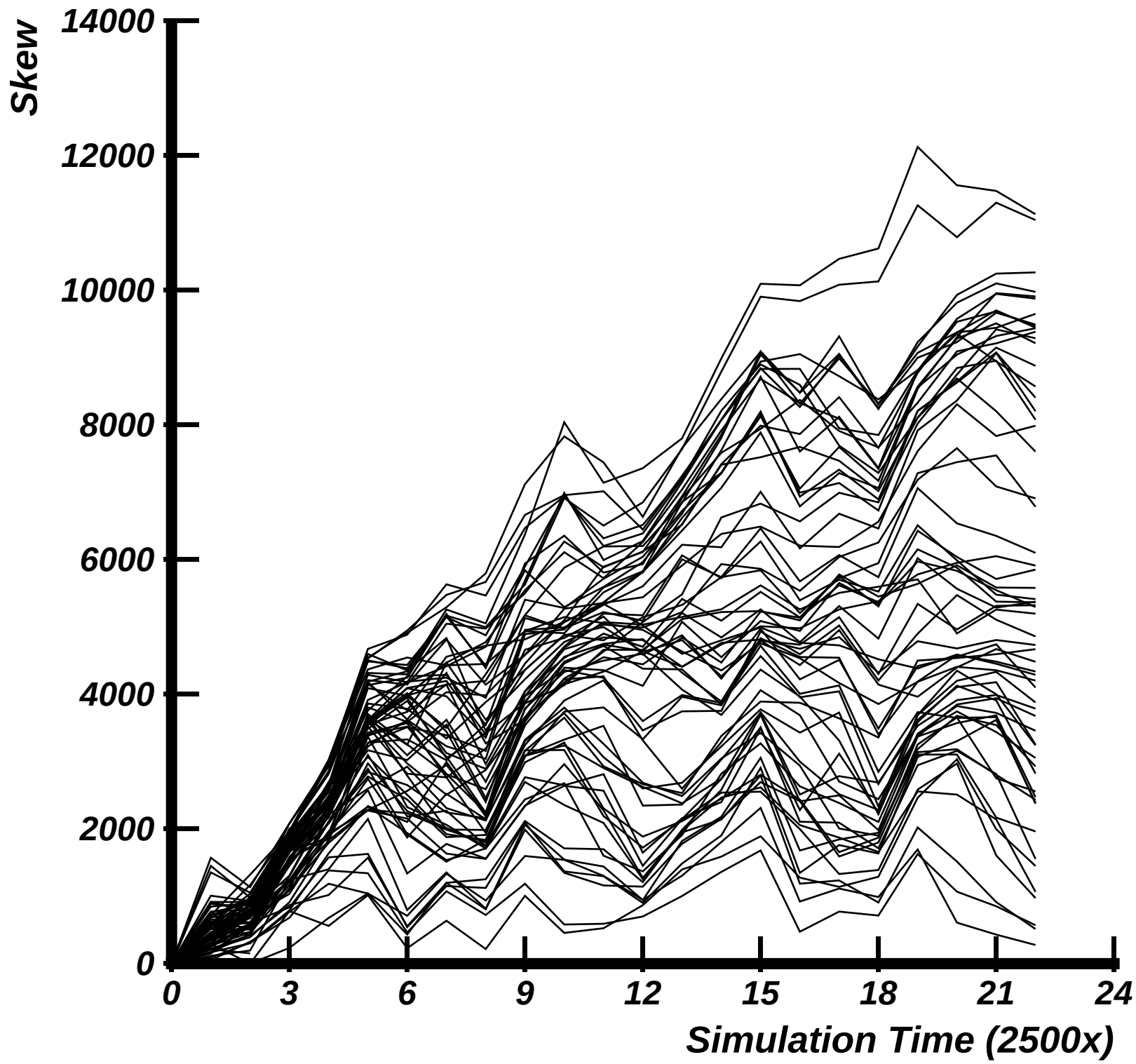
(a) Weather, Full-Map

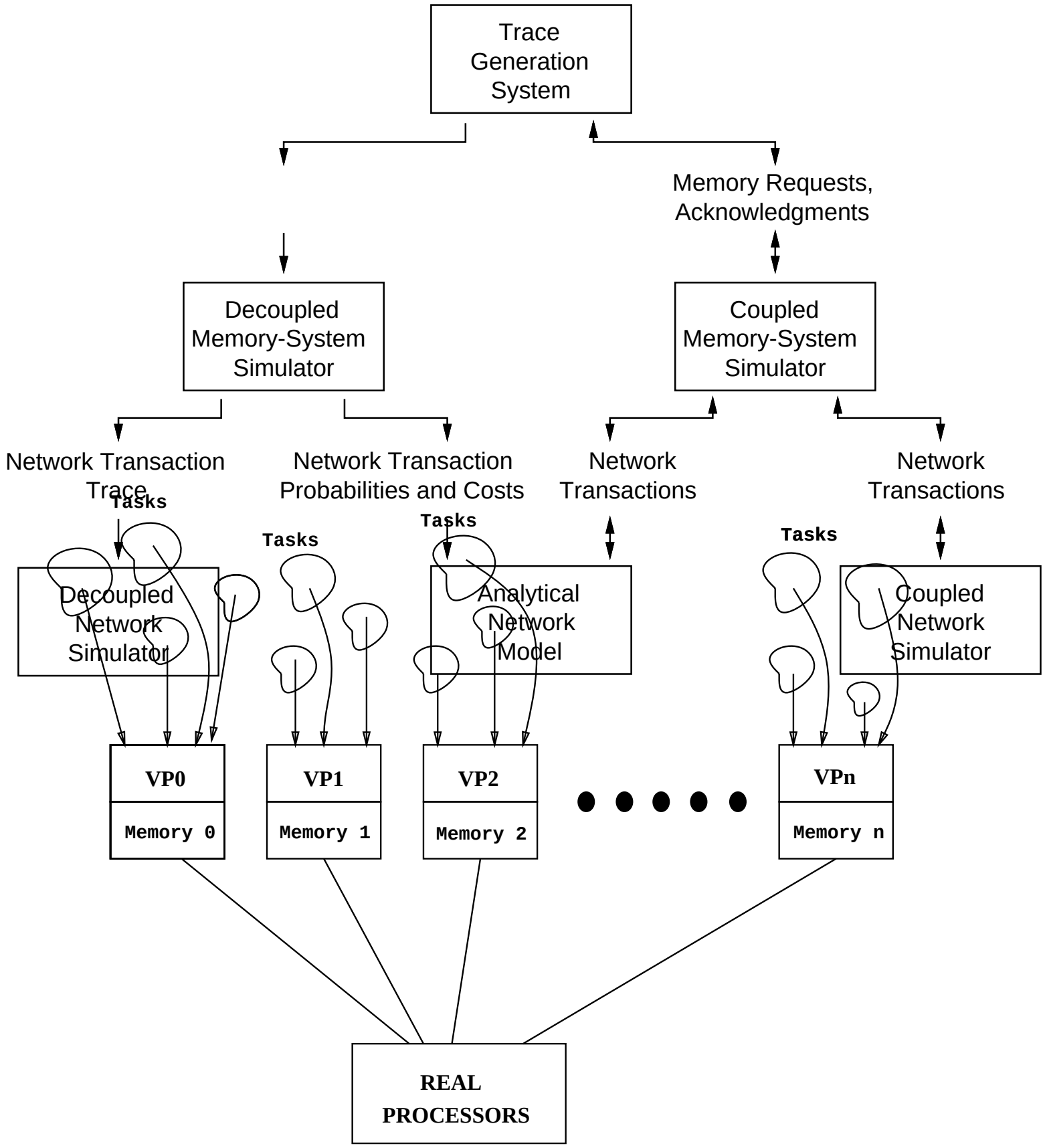


(b) Weather, Dir_4NB


Only Cache Private Data
Dir2NB
Dir4NB
Single Link Chain
Full Map
Coupled Simulations
Decoupled Simulations
0.00
0.20
0.40
0.60
0.80
1.00
Processor Utilization


Only Cache Private Data
Dir2NB
Dir4NB
Single Link Chain
Full Map
Coupled Simulations
Decoupled Simulations
0.00
0.20
0.40
0.60
0.80
1.00
Processor Utilization


Skew
14000
12000
10000
8000
6000
4000
2000
0
0
3
6
9
12
15
18
21
24
Simulation Time (2500x)


Trace Generation System
Memory Requests, Acknowledgments
Decoupled Memory-System Simulator
Coupled Memory-System Simulator
Network Transaction Trace
Network Transaction Probabilities and Costs
Network Transactions
Network Transactions
Tasks
Tasks
Tasks
Tasks
Decoupled Network Simulator
Analytical Network Model
Coupled Network Simulator
VP0
Memory 0
VP1
Memory 1
VP2
Memory 2
VPn
Memory n
REAL PROCESSORS