Empirical and Theoretical Lower Bounds on Energy Consumption for Networks on Chip

George Bezerra
Massachusetts Institute of Technology
CSAIL, Cambridge, MA 02139, USA.
gbezerra@csail.mit.edu

Dorian Arnold
University of New Mexico
Department of Computer Science
Albuquerque, NM 87131, USA
darnold@cs.unm.edu

Stephanie Forrest
University of New Mexico
Department of Computer Science
Albuquerque, NM 87131, USA
forrest@cs.unm.edu

ABSTRACT

This paper focuses on the network on chip of multi-core systems and proposes empirical and theoretical lower bounds on the energy consumption of applications. The empirical method consists of an linear programming model that simultaneously reduces communication distances and network traffic. When applied to standard benchmarks, our method shows that locality exploitation can lead to 50% energy reduction on average compared to no optimization. The theoretical lower bound is based on the Rent's rule model from VLSI design, and is obtained analytically from the communication graph structure of applications. The theoretical results show excellent agreement with the empirical lower bound.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Packet-switching networks*

General Terms

Design, Performance, Theory

Keywords

Network on chip, energy, multi-core, traffic locality, lower-bound, Rent's rule

1. INTRODUCTION

In Chip Multi-Processors (CMP) with a large number of cores, communication between cores over a Network on Chip (NoC) accounts for a significant fraction of the power budget of the system. For example, the NoC consumes 36% of power in the MIT Raw microprocessor [9], 10% in the Intel 48-core SCC [7], and 28% in the Intel Terascale 80-core chip [4]. As the number of cores on a die grows exponentially and network traffic increases, the need for efficient on-chip communication also increases. In this scenario, intelligent data placement strategies can be used to increase communication locality and significantly reduce NoC energy consumption.

In this paper, we analyze parallel applications running on homogeneous many-core architectures and determine empirical and theoretical lower bounds on the energy consumption for a 2D-mesh NoC. The empirical approach consists of an optimal data placement model that minimizes communication distances while increasing cache capacity utilization, thus simultaneously reducing the energy per packet and the network traffic. Based on a few assumptions, this method provides an accurate estimate of the minimum energy consumption achievable through locality optimization. Our simulation results show an average energy reduction of 50%, and up to 84%, on a system with 64 cores. The theoretical approach is based on the Rent's rule model [1] from VLSI circuit analysis. We adapt this model for CMPs and show that a theoretical lower bound can be obtained analytically from the topology of the communication graph of applications. The theoretical results agree closely with the empirical lower bound.

2. EMPIRICAL ANALYSIS

NoC energy consumption is a function of the communication locality and the total network traffic of an application. Because these two factors often conflict with each other, data placement optimization for energy minimization is a complex problem. In this section, we describe an linear programming model which, with a few simplifying assumptions, achieves the minimum NoC energy consumption.

2.1 Optimal data placement

We guarantee minimum energy consumption under the following assumptions: (1) The energy dissipated to transmit a packet is nearly proportional to distance (measured in number of hops). (2) The application workset fits in the aggregate cache capacity of the system. The first assumption is known to be approximately true in practice [5], and is used to define our cost function. The second simplifies the problem by guaranteeing the existence of a configuration with minimal traffic.

We define the cost of assigning a block i to core p as

$$C_{ip} = \sum_{p=1}^{N} \pi_{ip} \sum_{j=1}^{N} w_{ij} \cdot d_{pj}, \tag{1}$$

where d_{pj} is the distance between core p and the core running the thread j, w_{ij} corresponds to the total communication between the block and the thread, and N is the total number of

NoCArc '13, December 08 2013, Davis, CA, USA
Copyright 2013 ACM 978-1-4503-2370-3/13/12 ...$15.00.

cores. Finally,

$$\pi_{ip} = \begin{cases} 1 & \text{if block } i \text{ is in position } p \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

determines the assignment of a block to a particular core. We now define the following load-balancing constraints:

$$\sum_{p=1}^{N} \pi_{ip} = 1 \quad (3)$$

$$\sum_{i=1}^{B} \pi_{ip} \leq K. \quad (4)$$

Equation 3 expresses the constraint that a block can only be assigned to a single position. Equation 4 restricts the number of blocks assigned to each node to be smaller than or equal to the capacity constraint K, where K is the cache capacity of a node. Using the equations above, we define an Integer Programming (IP) model for data placement in multi-core systems as

Optimize:

$$\min_{\Pi} C_{total} : \sum_{i=1}^{B} \sum_{p=1}^{N} \pi_{ip} \sum_{j=1}^{N} w_{ij} \cdot d_{pj}$$

Subject to:

$$\sum_{p=1}^{N} \pi_{ip} = 1, \sum_{i=1}^{B} \pi_{ip} \leq K, \text{ and } \pi_{ip} \geq 0$$
$$\forall i \in \{1, 2, \ldots, B\} \text{ and } \forall p \in \{1, 2, \ldots, N\},$$

where the goal is to find the placement matrix Π containing all the variables π. The solution to this model corresponds to the maximum communication locality with the least traffic and, therefore, provides minimum energy consumption.

Unfortunately, because IP is **NP**-hard, the above model may not be computationally tractable even for relatively small instances. However, we were able to prove that the constraint matrix for this particular formulation is totally unimodular, which allows for a relaxed version of the problem, where the variable π takes continuous values, and which can be solved in polynomial time with any linear programming technique, such as the simplex algorithm.

2.2 Simulation setup

Full-system simulations were performed with Graphite [8]. Cores feature in-order, single issue execution. The L1-I and L1-D caches are 4-way set-associative with 32 KB cache-capacity, and 64-byte blocks. The L2-cache is 8-way set-associative with 512 KB capacity, and 64-byte blocks. The directories are full-map with no broadcast and use cache-line granularity. NoC energy consumption was measured with Orion-2 [6], and each hop on the 2D-mesh network takes one cycle. All simulations were performed on a 64-core system, and runtime and energy were measured after the initialization phase of applications. The parallel applications used in the simulations are POSIX Threads implementations of the modified SPLASH-2 benchmark [10].

2.3 Experimental results

We computed and ran the optimal placement for 10 scientific benchmarks using our model to determine the minimum NoC energy consumption. We use bin hoping, which has no locality optimization, as a baseline of comparison, and present normalized results for energy and runtime. We also report results for first-touch, a widely used heuristic for data placement optimization, in which a block of data is dynamically assigned to the first core that tries to access it.

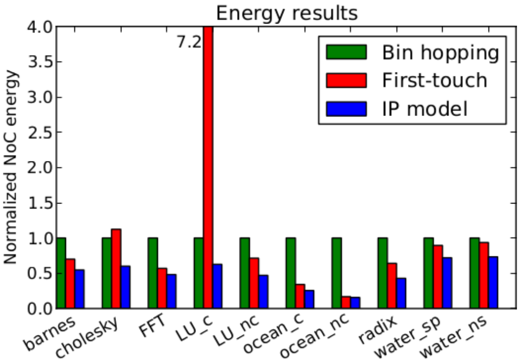

Figure 1: Normalized energy results.

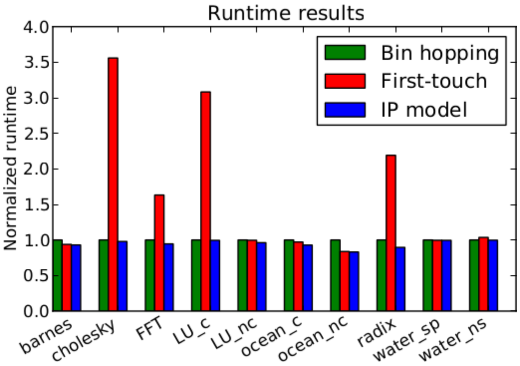

Figure 2: Normalized runtime results.

The energy results are shown in Figure 1, where an energy reduction of approximately 50% on average was obtained by our model relative to bin hopping, and up to 84%, for the `ocean_non_contiguous` application. Notice that for most applications first-touch also obtains significant energy improvements, approaching in some cases the lower bound defined by our algorithm, although never surpassing it. In other cases, however, it performs poorly. The extreme case is `LU_contiguous` in which energy is increased by 7×!

We also report a small performance improvement with our algorithm, as shown in Figure 2. Runtime is reduced because packets travel a smaller number of hops, which decreases latency. However, because many factors that impact performance are not accounted for in our formulation, there is no true guarantee this is also a performance lower bound. First-touch also obtains performance improvements in some cases, but for four applications it slows down execution significantly.

An analysis of the number of cache misses reveals why first-touch performs poorly in some cases. Similar to bin hopping, the IP model takes full advantage of the aggregate cache capacity of the system and, therefore, no capacity misses or evictions are observed in the entire execution of the applications. However, first-touch does not constrain the number of blocks assigned to each core, which may result in unbalanced placements. The poor runtime results for first-touch in cholesky, FFT, LU_contiguous, and radix are all correlated with a large number of evictions (ranging from the order of 10^5 evictions for FFT, up to 10^6 for LU_contiguous).

Our data placement formulation allows an optimal, polynomial-time solution to a complex problem, providing an empirical lower bound on NoC energy consumption for multi- and many-core systems. This creates a baseline for assessing the quality of data placement algorithms as well as revealing the potential for communication locality exploitation in parallel applications.

3. THEORETICAL ANALYSIS

We now analyze NoC energy consumption theoretically. This analysis requires no optimization heuristic, and is based only on the topology of the communication graphs.

3.1 The bandwidth version of Rent's rule

Rent's rule is a fractal pattern observed in the interconnection structure of digital circuits. The applicability of Rent's rule to multi-core chips was first discussed in [2], where the bandwidth version of Rent's rule was proposed. In [3], the authors applied the hierarchical partitioning algorithm to the communication graph of several parallel applications, showing that, similar to digital circuits, they also follow a fractal pattern. The bandwidth version of Rent's rule is given by

$$B = bN^p, \tag{5}$$

where B is the communication bandwidth sent or received by a cluster of N network nodes, b is the average communication per node, and $0 \leq p \leq 1$ is the Rent's exponent. A communication graph follows Rent's rule if its behavior when plotted on a log-log plot of N vs. B can be approximated by a straight line, where the slope of this line is the Rent's exponent. To measure p and b, we applied the hierarchical partitioning algorithm to the communication graph of the benchmark applications and obtained a wide variation in the Rent's exponent from 0.36 up to 0.94.

3.2 Modeling Communication Locality

Rent's rule can be used to provide a theoretical upper-bound on communication locality. We use the Wire Length Distribution (WLD) model of [1], which was initially developed for VLSI circuits, and compute the average distance traveled by a message in different applications. Equation 6 defines the probability of having a wire connecting two logic gates with Manhattan distance d. We use this equation to represent the probability of communication between cores, where N is the number of cores on a square mesh network.

Region I: $1 \leq d < \sqrt{N}$

$$P(d) = \frac{\Gamma}{2N\left(1 - N^{p-1}\right)} \left(\frac{d^3}{3} - 2\sqrt{N}d^2 + 2\sqrt{N}d\right) d^{2p-4}$$

Region 2: $\sqrt{N} \leq d < 2\sqrt{N} - 2$

$$P(d) = \frac{\Gamma}{6N\left(1 - N^{p-1}\right)} \left(2\sqrt{N} - d\right)^3 d^{2p-4} \tag{6}$$

where Γ is a normalization constant. From the above formula, the average communication distance is computed as the weighted sum of the probabilities with their respective distances as

$$\overline{d} = \sum_{d=1}^{2\sqrt{N}-2} d \cdot P(d). \tag{7}$$

Figure 3(a) shows the results of the model using the Rent's exponents, p, computed in the previous section. The figure compares the empirical (after optimization with our model) and theoretical lower bounds on the communication distance of applications. The dashed line represents values where the predicted values equal the measured ones. It is evident that all points fall below the dashed line, suggesting that the Rent's rule model provides an overly-optimistic lower bound, underestimating the empirical distance values with an average error of 27.52%.

Part of the error is explained by the fact that the empirical locality is constrained by the limited capacity of nodes, which is not accounted for by Rent's rule. In fact, we found in our experiments that a system with no capacity constraints would lead to communication distances which are 8% smaller on average. Another contributing factor is the small scale of the analyzed system. By measuring Rent's rule on a 64-node system, the weight of Region II of Rent's rule is greater in the measurements, thus biasing the Rent's exponents towards lower values. In VLSI, Rent's rule is generally applied to systems with tens of thousands of nodes.

However, the high correlation coefficient of 96% shows that, even at this small scale, Rent's rule can explain most of the variation in communication distance and is, therefore, a good predictor of communication locality in CMP applications. To improve the accuracy of the model and account for the deviations described above, we performed a linear correction by adding a multiplicative coefficient to the model. Using least squares regression, a coefficient value 1.38 was extracted which maximizes the fit of the model to the data. The new curve, which has average error of 4.58% and maximum error of 8.64%, is shown in Figure 3(b). Once adjusted in this manner, the results generalize for new parallel applications running on the same system.

3.3 Energy predictions

The energy consumption of the interconnect is easily computed from the estimated average communication distances. The energy used by a message of length l (in bytes) when traversing one hop on a 2D-mesh NoC is given by

$$E_{hop}(l) = E_{router}(l) + E_{link}(l), \tag{8}$$

where E_{router} and E_{link} are the average energy used by the message when traversing a router and a link, respectively. The average energy used per byte can be obtained by

$$E_{hop}(1) = \frac{\sum_l E_{hop}(l) \cdot N_l}{\sum_l l \cdot N_l}, \tag{9}$$

where N_l is the number of messages of size l. Using the estimated average distance traveled by a message (Equation 7), the energy used by a byte when traversing a hop (Equation 9), and the average number of bytes per node (the parameter b of Rent's rule) the total energy of the application can be calculated:

$$E_{total} = \overline{d} \times E_{hop}(1) \times N \times b, \tag{10}$$

where $N \times b$ is the total number of bytes sent and received over the network.

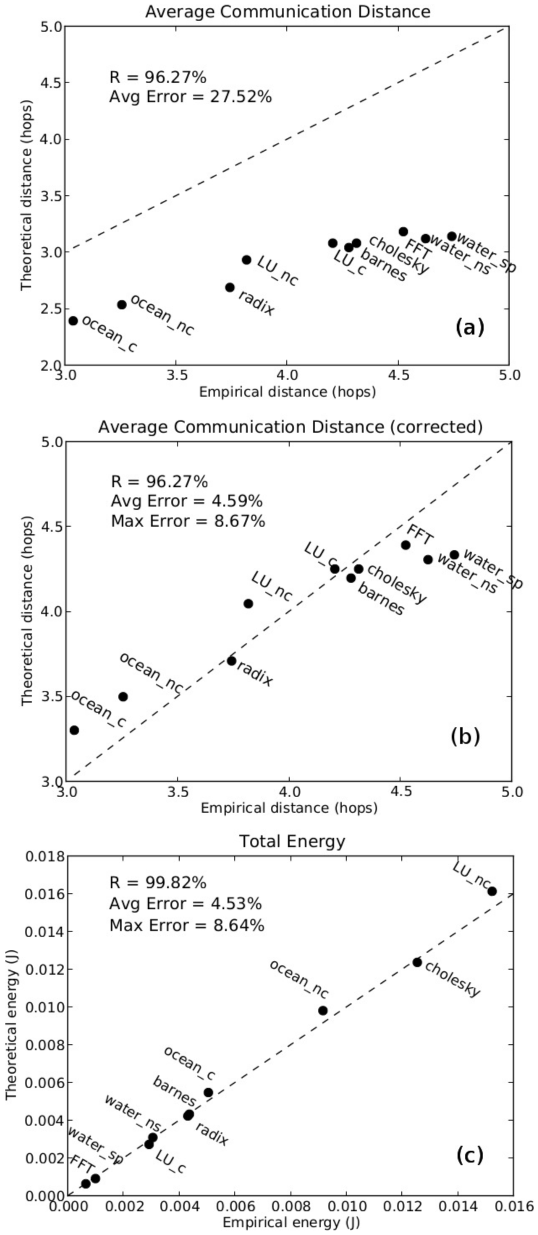

Figure 3: (a) Comparison between empirical and theoretical average communication distances. (b) Empirical and theoretical communication distances after linear least squares fit. (c) Empirical and theoretical NoC energy consumption.

We compare the theoretical and empirical energy lower bounds in Figure 3(c), where the theoretical values were obtained using the corrected Rent's rule model. The results show excellent agreement, with a high correlation coefficient of 99.82%, average error of 4.53%, and maximum error of 8.64%. Notice that the total energy requires the two Rent's rule parameters, p and b, while the average distance only requires the Rent's exponent. Because there is little uncertainty regarding the measurement of b, the correlation coefficient for total energy is actually higher in this case.

4. CONCLUSION

This paper presented an empirical and a theoretical methods for determining the maximum energy savings from communication locality exploitation in an NoC. The empirical method uses a polynomial time algorithm that computes the optimal allocation of data to cores such that energy is minimized. The theoretical one is based on a model from VLSI design called Rent's rule and can be used for fast, first order energy estimates that do not require simulation or optimization. We showed that the empirical and theoretical results are highly consistent with each other. Our methods can be used as a baseline for designing better data placement heuristics as well as more energy efficient applications.

5. REFERENCES

[1] J. A. Davis, V. K. De, and J. D. Meindl. A stochastic wire-length distribution for gigascale integration (GSI) - Part I: Derivation and validation. *IEEE Transactions on Electron Devices*, VOL 45(3):580–589, 1998.

[2] D. Greenfield, A. Banerjee, J.-G. Lee, and S. Moore. Implications of Rent's rule for NoC design and its fault-tolerance. In *Proceedings of the First International Symposium on Networks-on-Chip (NOCS'07)*, 2007.

[3] W. Heirman, J. Dambre, D. Stroobandt, and J. Campenhout. Rent's rule and parallel programs: Characterizing network traffic behavior. In *Proceedings of the 2008 International Workshop on System Level Interconnect Prediction, SLIP'08*, 2008.

[4] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-GHz mesh interconnect for a teraflops processor. *IEEE MICRO*, 27(5):51–61, 2007.

[5] J. Hu and R. Marculescu. Energy-aware mapping for tile-based NOC architectures under performance constraints. In *Proceedings of ASP-Design Automation Conference*, pages 233–239, 2003.

[6] A. Kahng, B. Li, L. Peh, and K. Samadi. Orion 2.0: A fast and accurate NOC power and area model for early-stage design space exploration. In *Design, Automation, and Test in Europe*, pages 423–428, 2009.

[7] T. Mattson, M. Riepen, T. Lehnig, P. Brett, W. Haas, P. Kennedy, J. Howard, S. Vangal, N. Borkar, G. Ruhl, et al. The 48-core scc processor: the programmer's view. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–11. IEEE Computer Society, 2010.

[8] J. Miller, H. Kasture, G. Kurian, C. Gruenwald, N. Beckmann, C. Celio, J. Eastep, and A. Agarwal. Graphite: A distributed parallel simulator for multicores. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, pages 1–12, 2010.

[9] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, J.-W. Lee, and W. Lee. The Raw microprocessor: A computational fabric for software circuits and general purpose programs. *IEEE MICRO*, 22(PART 2):25–35, 2002.

[10] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta. The splash-2 programs: Characterization and methodological considerations. In *Proceedings of the 22nd annual international symposium on Computer architecture*, pages 24–36, 1995.