Optimizing the performance of a data vortex interconnection network

Assaf Shacham and Keren Bergman

Department of Electrical Engineering, Columbia University, 500 West 120th Street, New York, New York 10027, USA

The definition of the data vortex architecture leaves broad room for decisions regarding the exact design point required for achieving a desired performance level. A detailed simulation-based study of various parameters that affect a data vortex interconnection network's performance is reported. Three implementations are compared by acceptance rate, latency, and cost. © 2007 Optical Society of America

OCIS codes: 060.0060, 060.2360, 060.4250.

1. Introduction

The data vortex is an optical-packet-switching architecture targeted to address the performance requirements of next-generation high-performance computing systems (HPCS). High bandwidth, ultralow latency, and packet-level granularity are realized by transmission of wavelength-striped packets through end-to-end photonic paths. Based on a distributed structure of 2×2 switching nodes, data vortex interconnection networks route the packets according to optically encoded addresses while resolving contentions through deflection routing [1–4].

Since its initial presentation by Yang *et al.* [1], the data vortex architecture has been studied from various aspects; namely, architecture [5], performance [2,6], implementation [3,7], and physical layer [8], and a complete 12×12 prototype interconnection network has been assembled in a laboratory setup [3]. The performance study of the data vortex is of great interest because of the topological complexity and the large number of parameters that can be tuned to attain a desired performance level. Some of these parameters are the network dimensions, packet injection and extraction policies, and the employment of supplemental photonic injection control modules (ICMs) [9].

In this paper these parameters are studied using computer simulations in order to examine their effect on the latency and the acceptance rate. Several system configurations are considered, and their performance is weighed against a cost metric. The paper is organized as follows: A brief overview of the data vortex architecture is given in Section 2. In Section 3, the performance metrics are defined and the simulated problem is formulated. Results are given and explained in Section 4. Section 5 provides a concluding discussion.

2. Architecture and Performance Metrics

The data vortex topology is a multistage interconnection network, composed of 2×2 switching nodes, where contentions are resolved by buffering packets in the same stage until they can be routed to the subsequent one. To facilitate optical buffering of packets in the stages, additional switching nodes are added at every stage and packets are deflected to switching nodes in the same stage when their progression path is occupied. The additional switching nodes are organized in circles guaranteeing the availability of deflection paths, so the stages can be viewed as 3D cylinders. A data vortex interconnection network is, therefore, defined by three structural parameters: H, the height of the network; $C (=\log_2 H + 1)$, the number of stages; and A, the number of switching nodes along the circumference of each stage. Figure 1 illustrates an exemplary data vortex network.

The native packet format in the network is wavelength striped, where the data are encoded simultaneously on several wavelengths, to achieve large transmission band-
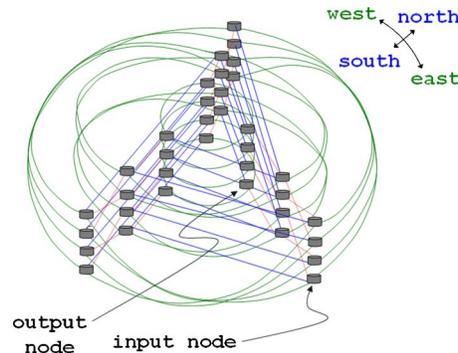
Fig. 1.   Example data vortex topology with the dimensions $C=3$, $H=4$, $A=3$. This network has $N=36$ switching nodes and can have 4 or 12 I/O ports, according to the operation mode chosen. Progression paths (blue), deflection paths (green), and control cables (red) are shown.

width [10]. The packets are injected into the network in a time-slotted manner through the switching nodes in the input stage (outer cylinder). They are then routed inward, one node-hop per time slot, in a binary-tree fashion to a height that corresponds to their optically encoded address. Destination tag routing, where a single bit is required for the routing in each stage [11], is employed to simplify the address decoding and the switching-node design [7]. The information about the availability of each stage is transmitted upstream by means of a minimal electronic signaling mechanism. For additional information on the data vortex interconnection network architecture, the reader is referred to [3].

The data vortex architecture guarantees that packets are not lost inside the system and congestion is manifested as backpressure, transmitted on control cables interconnecting the switching nodes. When the network becomes congested, the backpressure propagates to the input ports and packets may be delayed in the input stage (i.e., the outer cylinder), blocking the injection of new packets. The acceptance rate is, therefore, defined as the ratio of the successfully injected packets (packets not blocked by internal traffic) to the number of total injection attempts.

The packet latency in data vortex interconnection networks, measured in node hops, is nondeterministic and relies to a large extent on the rate of contentions in the network and the level of congestion. The more frequent the contentions, the longer it takes for packets to find available progression paths, so additional hops are taken in each stage. As the load increases, the mean packet latency grows larger, as is the width of the latency distribution [2].

In a HPCS interconnection network, it is desirable to keep the latency as low as possible, for a minimal memory access time. Limiting the latency variance is considered necessary to allow for efficient programming and predictable performance [12]. The latency and latency variance should, therefore, be minimized when designing a data vortex interconnection network. Since every denied packet has to be reconstructed and therefore consume expensive transmitter time and power, a near 100% acceptance rate is also required.

Various design parameters can be tuned to control performance. Populating only a subset of the ports, limiting the load in a statistical or timing-based manner are a few possibilities. The effects of these port-populating and injection strategies are studied in the following sections.

3. Simulation Methodology

To investigate the effect of port population and injection strategies on the data vortex performance, a C++ cycle-accurate simulator is constructed. A set of simulation runs are conducted, with 50,000 to 500,000 packets injected in each run. The traffic model is uniform Bernoulli traffic, which provides a good estimate of the upper boundary of the network performance [11]. Using the simulator, three 64-port systems are modeled and compared. The 64-port system is chosen as the subject of the study because it can support, using clustering, a large number of processors while remaining manageable in terms of packaging. Larger networks can be constructed by scaling the data vortex structural parameters or using a hierarchial design [13].

In the first system, the data vortex dimensions are $(C,H,A)=(7,64,3)$, yielding a node count of $N=7\times64\times3=1344$. Injection is only performed into a single node at each of the 64 levels. At the output stage, each of the levels is used as a single output port, and packets are extracted once they reach any node in the output stage, so that no extra hops are required. This system will be referred to as the *single-angle* system.

The second and the third systems are of the dimensions $(C,H,A)=(5,16,4)$, yielding a node count of $N=5\times16\times4=320$. All the switching nodes in the input stage are used as input ports, and all the switching nodes in the output stage are used as output ports. In these systems, the packets are required to travel along the circumference of the output stage to reach the desired output port, located at a specific angle. The two systems are differentiated by their injection strategies: in the *simple-all-angle* system, any input port may inject at any time, whereas in the *token-all-angle* system an input port may only inject if its switching node holds a token, which may be received by the node periodically. In other words, each input port may inject only once every N_T slots, where N_T is defined as the token period, and the input ports are divided to N_T groups to balance the load on the system. Naturally, in the case of the token-all-angle system the maximum offered load is limited to $1/N_T$.

The three systems are compared by three parameters: (1) acceptance rate, (2) median latency, and (3) 99.9% percentile latency. For the token-all-angle system, we also study the effect of varying N_T between the values 3, 4, and 5. The systems are simulated with and without ICMs [9], whose task is to reattempt injection of blocked packets to improve the acceptance rate. In this simulation, as experimentally demonstrated in [9], the ICMs may buffer only a single packet and packets are dropped after three failed injection attempts, or when the ICM is already full.

The relative cost of the systems is, obviously, of interest when comparing their performance. The wavelength-parallel structure of the packets facilitates the use of very simple switching nodes, each based on two semiconductor optical amplifiers (SOAs) controlled by inexpensive electronics. Conversely, the generation of these packets requires multiple high-speed expensive optoelectronic elements in the terminals for the encoding and decoding of multiple wavelengths concurrently.

To estimate the relative costs of the subsystems used in the network (switching nodes, ICMs, and O/E interfaces), we use a simple model that counts the number of optoelectronic elements: switches, modulators, and receivers. Since these elements represent the lion's share of the cost in the construction of these subsystems, this model provides a simple, yet reasonable, estimate.

Denoting the cost of a switching node (with two SOAs) as α, we estimate the cost of an I/O terminal (for 16-wavelength packets [7] it uses 16 high-speed modulators, 6–8 low-speed modulators for header wavelengths, and 16 high-speed optical receivers) at 25α. The cost of an ICM, using four SOAs, can be estimated at 2α. The total cost of the systems with (without) ICMs is therefore 3200α (2944α) for the single-angle system and 2176α (1920α) for the all-angle systems.

4. Simulation Results and Discussion

4.A. Acceptance Rate

The acceptance rate curves of the simulated systems, without ICMs, are shown in Fig. 2. The token-all-angle system is simulated three times with different token periods (N_T). Some conclusions can be drawn from these simulations. First, the perfor-
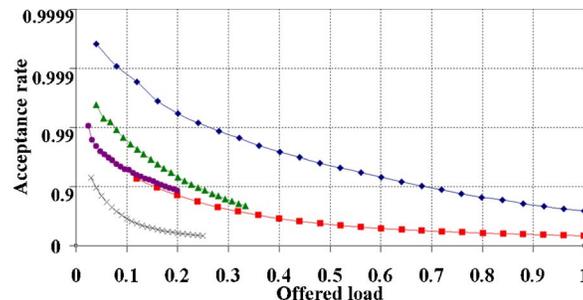

Fig. 2. Acceptance rate curves of the modeled systems without ICMs: single angle (♦), simple all angle (■), and token all angle [N_T=3 (▲), 4 (×), 5 (•)].
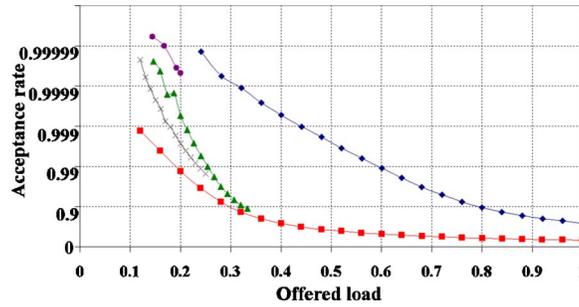
Fig. 3.   Acceptance rate curves of the three modeled systems with ICMs: single angle (♦), simple all angle (■), and token all angle [N_T=3 (▲), 4 (×), 5 (•)].

mance of the token all angle strongly depends on the choice of the token period N_T. The systems with N_T=4 and N_T=2 (not shown) exhibit poor performance, and the one with N_T=3 shows performance exceeding that of the simple-all-angle system. This can be explained by the angular progression of packets in the data vortex and the need to decouple the injection process from the internal packet movement. If the rotating injection process moves with the packets as they traverse around the cylinders in the system (i.e., coupled to the packet movement), repeated deflections occur at every angle and the acceptance rate falls. If, conversely, the injection and movement processes are decoupled, the chances of contentions between packets in the system and packets that attempt injection are reduced.

Choosing a value for N_T that is relatively prime to A, the data vortex angle parameter, produces the best performance. It is also observed that the single-angle system drastically outperforms the all-angle systems. This can be explained by two reasons: (1) larger internal buffering capacity in the additional switching nodes and (2) simplified extraction strategy that does not require extra node hops at the output stage, thus avoiding the increased backpressure.

Adding the ICMs (Fig. 3) notably improves the acceptance rate, as predicted in [9]. At acceptance rates approaching 1.0, the plots are limited by the simulation length, but the sensitivity of the different systems to the offered load is clear from the trends. The single-angle system offers the best performance and the lowest sensitivity owing to the factors described in Section 3. The token-all-angle systems also exhibit better than the simple-all-angle system, since every packet enjoys at least three undisturbed injection attempts, and no packets are dropped due to the ICM single-packet capacity limits. The main conclusions are that the operating load should be kept below 0.3 for any system and that the choice of system should be based on finding an operating point, based on a desired acceptance rate, on a simulation-based curve similar to the curves in Fig. 3.

4.B. Latency
As explained in Section 2, the latency and the width of its distribution should be minimized to ensure a low memory access time and predictable HPCS performance. Because of the irregular distribution of the packet latency in the data vortex, the mean latency figure does not provide all the required information. In Figs. 4 and 5 the median latency and the 99.9 percentile latency are plotted as estimates of the expected latency and the width of the latency distribution, respectively.
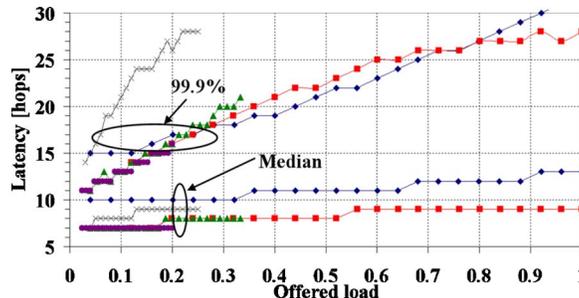

Fig. 4.   Latency curves of the modeled systems without ICMs: single angle (♦), simple all angle (■), and token all angle [N_T=3 (▲), 4 (×), 5 (•)].
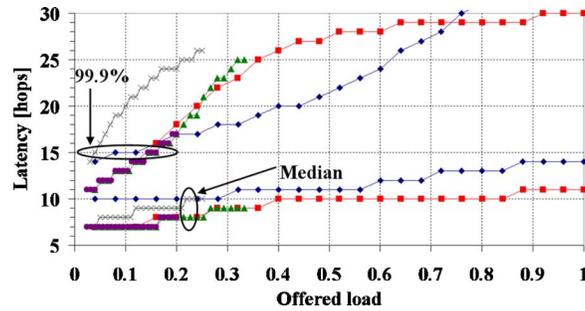
Fig. 5.   Latency curves of the modeled systems with ICMs: single angle ($\blacklozenge$), simple all angle ($\blacksquare$), and token all angle [N_T=3 ($\blacktriangle$), 4 ($\times$), 5 ($\bullet$)].

Comparing Figs. 4 and 5, although it may seem that adding the ICM increases the latency for a given load, this is not the case because due to the improved acceptance rate offered by the ICMs, the systems at Fig. 5 are actually working under a higher load. It can also be observed that the additional nodes and paths and the reduced backpressure resulting from the quicker packet extraction policy in the single-angle system are proven to be beneficial in reducing the latency.

5. Conclusions

Injection policies and input port-population strategies are studied as means of controlling the acceptance rate and the latency of a 64-port data vortex interconnection network. The main conclusions to be drawn are that many parameters affect the performance and a detailed simulation should be run to ascertain the exact performance of any configuration.

Operating loads at approximately 0.25, which are required to attain sufficient performance, are viable due to the enormous peak bandwidths offered by wavelength-striping and the nature of memory access patterns in HPCS [14]. Under this load, the *single-angle* system provides an acceptance rate better than 0.9999, a median latency of 10 hops, and a 99.9% latency of 17 hops. Its cost, ~50% higher than that of the other systems, must be weighed against these advantages.

Additionally, the performance gained by adding injection control modules [9] is verified. This work was performed for systems of a specific size, but the methodology can be applied to investigate other design parameters and systems of different sizes.

Acknowledgment

References

1.   Q. Yang, K. Bergman, G. D. Hughes, and F. G. Johnson, "WDM packet routing for high-capacity data networks," J. Lightwave Technol. **19**, 1420–1426 (2001).
2.   Q. Yang and K. Bergman, "Performances of the data vortex switch architecture under nonuniform and bursty traffic," J. Lightwave Technol. **20**, 1242–1247 (2002).
3.   A. Shacham, B. A. Small, O. Liboiron-Ladouceur, and K. Bergman, "A fully implemented 12×12 data vortex optical packet switching interconnection network," J. Lightwave Technol. **23**, 3066–3075 (2005).
4.   G. I. Papadimitriou, C. Papazoglou, and A. S. Pomportsis, "Optical switching: switch fabrics, techniques, and architectures," J. Lightwave Technol. **21**, 384–405 (2003).
5.   C. Hawkins, B. A. Small, D. S. Wills, and K. Bergman, "The Data Vortex, an all optical path multicomputer interconnection network," IEEE Trans. Parallel Distrib. Syst. **18**, 409–420 (2007).
6.   C. Hawkins and D. S. Wills, "Impact of number of angles on the performance of the data vortex optical interconnection network," J. Lightwave Technol. **24**, 3288–3294 (2006).
7.   B. A. Small, A. Shacham, and K. Bergman, "Ultra-low latency optical packet switching node," IEEE Photon. Technol. Lett. **17**, 1564–1566 (2005).
8.   O. Liboiron-Ladouceur, B. A. Small, and K. Bergman, "Physical layer scalability of WDM optical packet interconnection networks," J. Lightwave Technol. **24**, 262–270 (2006).
9.   A. Shacham, B. A. Small, and K. Bergman, "A wideband photonic packet injection control module for optical packet switching routers," IEEE Photon. Technol. Lett. **17**, 2778–2780 (2005).

10. T. Lin, K. A. Williams, R. V. Penty, I. H. White, M. Glick, and D. McAuley, "Performance and scalability of a single-stage SOA switch for 10×10 Gb/s wavelength striped packet routing," IEEE Photon. Technol. Lett. **18**, 691–693 (2006).

11. W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks* (Morgan Kaufmann, 2004).

12. D. Dai and D. K. Panda, "How much does network contention affect distributed shared memory performance?" in *Proceedings of the International Conference on Parallel Processing, Bloomingdale, Ill., August 11–15, 1997*, pp. 454–461.

13. C. Minkenberg, F. Abel, P. Müller, R. Krishnamurthy, M. Gusat, and B. Roe Hemenway, "Control path implementation for a low-latency optical HPC switch," in *Proceedings of the 13th Annual IEEE Symposium on High Performance Interconnects* (IEEE, 2005), pp. 29–35.

14. S. Petit, J. Sahuquillo, and A. Pont, "Characterizing parallel workloads to reduce multiple writer overhead in shared virtual memory systems," in *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing* (IEEE, 2002), pp. 261–268.