Copyright Notice

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Foreword

Further investigations into this topic showed, that the formula given to calculate the tail-SRAM size with PRR-MMA holds not for all traffic arrival patterns. According to our current knowledge the correct formula is $Q \cdot (k+1)/2$. This amount of shared tail-SRAM guaranties zero packet loss for any traffic arrival pattern. With this, a 100 Gbps system with $k=21$ can still decrease the tail-SRAM size by 47 %.

We found further, that this formula holds also for the plain round robin MMA. Concluding we can say, that the shared tail-SRAM accounts for the complete decrease in SRAM size.

27.10.2009
Arthur Mutter
arthur.mutter@ikr.uni-stuttgart.de

A Novel Hybrid SRAM/DRAM Memory Architecture for Fast Packet Buffers

Arthur Mutter
University of Stuttgart, IKR
Pfaffenwaldring 47, Stuttgart, Germany
arthur.mutter@ikr.uni-stuttgart.de

ABSTRACT

This paper addresses the design of fast packet buffers for high speed Internet routers and switches. These buffers usually use a memory hierarchy that consist of expensive but fast SRAM and cheap but slow DRAM to meet both speed and capacity requirements. One challenge building these packet buffers is to provide worst-case bandwidth guarantees and fixed latencies, not to stall pipelines or to reduce throughput. My colleagues and I propose a novel packet buffer architecture along with a new memory management algorithm which reduces the amount of required SRAM compared to other architectures, e. g. by 73 % for a 100 Gbps system using DDR3-DRAM. Furthermore, our architecture scales well with line rate.

Categories and Subject Descriptors

C.2.6 [**Internetworking**]: Routers

General Terms

Algorithms, Design, Performance

1. INTRODUCTION

Internet routers and switches need buffers to store packets in case of congestion. Buffering is typically done on each individual line card, which maintains a separate FIFO queue for each service class and egress port (virtual output queuing, VOQ). The number of VOQs to be maintained is in the order of hundreds to thousands.

Until recently, vendors built packet buffers from cheap, high capacity DRAM (dynamic RAM). The drawback of DRAMs is their rather long access time of around $T = 50$ ns. T determines the minimum time between two consecutive accesses to the DRAM in the worst-case and limits the number of possible read/write operations. For example, it takes 5.12 ns to receive a 64 byte packet at 100 Gbps. A stream of 64 byte packets requires an access time of $T = 2.56$ ns as packets have to be written to and read from memory.

ANCS'09, October 19-20, 2009, Princeton, New Jersey, USA.

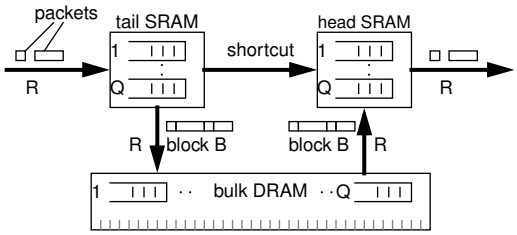

Figure 1: Basic hybrid SRAM/DRAM architecture

Researchers propose a hybrid SRAM/DRAM (HSD) architecture [1, 2, 3] to overcome this gap in access time and provide worst-case bandwidth guarantees. SRAM meets access time requirement and DRAM capacity and throughput requirements. The drawback of the proposed architectures is that they require large amounts of expensive SRAM.

In this paper, we propose a novel hybrid SRAM/DRAM packet buffer architecture. Our new Memory Management Algorithm (MMA) reduces the required amount of SRAM significantly, while it still provides worst-case bandwidth guarantees. Furthermore, our architecture scales well with respect to the line rate.

2. RELATED WORK

Several HSD packet buffer architectures have been proposed with worst-case bandwidth guarantees. [1] initially introduced a HSD packet buffer (cf. Fig. 1). It maintains Q FIFO queues (one per flow) and stores the heads and tails of all queues in the corresponding SRAM. The remaining part is stored in bulk DRAM. As soon as for a flow enough data has arrived in the tail SRAM, an MMA initiates the transfer of a large fixed size data block B to the corresponding queue in DRAM. Similarly, in preparation for a packet departure, an MMA initiates a data transfer of same block size B from the corresponding queue in DRAM to the head SRAM. The authors prove, that in worst-case the required SRAM size, both at head and tail, is QB.

[2] shows that exploiting bank interleaving reduces the SRAM size. The price they pay is fragmentation of the DRAM for special traffic patterns and additional effort for DRAM bank management. In contrast to today's DRAMs, they assume DRAMs with several hundred banks, which do not exist now and in near future. According to our calculations, the SRAM savings for a system with $R = 100$ Gbps, $Q = 5000$ flows and DDR3-DRAM (8 banks) will only be around 26 %.

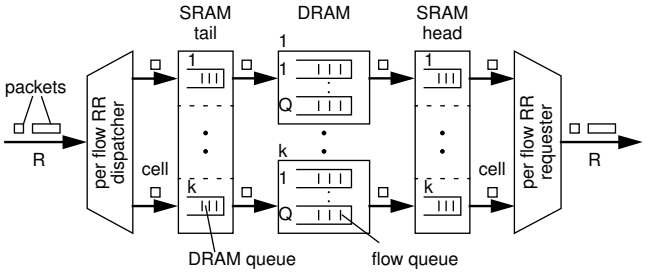

Figure 2: General PHSD architecture

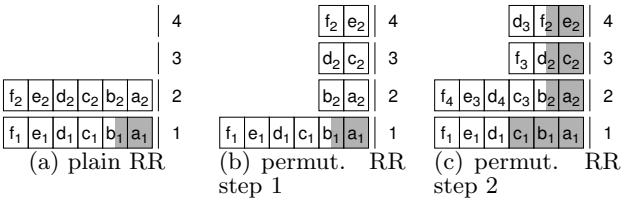

Figure 3: Status of the 4 DRAM queues in SRAM; grey cells were already transmitted to DRAM

[3] presents a PHSD (parallel hybrid SRAM/DRAM) architecture built from many parallel DRAMs operated individually. Finally, for worst-case bandwidth guarantees, it requires the same amount of SRAM as the first proposal [1].

There are further proposed PHSD packet buffer architectures, which give only statistical guarantees based on the available SRAM size, e. g. [4]. Consequently, these systems occasionally lose packets. As we give a 100 % bandwidth guarantee, we do not compare to statistical systems here.

3. PROPOSED ARCHITECTURE

Our proposal bases on the PHSD architecture in [3]. Fig. 2 shows a general PHSD architecture with k parallel DRAMs. Each DRAM provides $1/k^{th}$ of the required bandwidth and contains Q FIFO flow queues, i. e. each logical flow queue is spread over all k DRAMs. The head and tail SRAMs contain just one FIFO queue per DRAM.

The dispatcher segments incoming variable length packets into fixed length cells and distributes them per flow in a round robin (RR) sequence over all k DRAMs. This is necessary to read/write a single flow from/to memory as only all k DRAMs together provide line rate, i. e. every k^{th} cell of every flow is dispatched to the same DRAM. The minimal access granularity of the DRAMs determines the cell size, e. g. 64 byte for DDR3-DRAM. At a given cell size, the line rate determines k to achieve the required access time.

In the following, we focus on the tail SRAM as the problem is symmetrical. Assume, we have $k = 4$ DRAMs and $Q = 6$ flows named a to f. If now the cell arrivals of the individual flows correlate conflictingly to the RR sequence, up to Q cells will accumulate per DRAM queue. Fig. 3(a) shows the case after the sequential arrival of the cells a_1 to f_1, a_2 to f_2. a_i denotes the i^{th} cell of flow a. Accumulation can happen since the individual DRAMs operate at $1/k^{th}$ of the line rate. Meanwhile the other DRAMs are idle.

If we can reduce this conflicting correlation, all DRAMs will receive and process cells. To achieve this, we propose an architecture with the following improvements over [3]: a) use of a new Permutated Round Robin - Memory Management Algorithm (PRR-MMA) in the dispatcher; b) use of shared tail SRAM memory for all DRAM queues to save SRAM.

Instead of using the same RR sequence for all Q flows our PPR-MMA uses Q different permutated sequences in the best case. With k DRAMs $P = (k-1)!$ different permutated RR sequences are possible. For $k = 4$, the $P = 6$ RR sequences are: [1234], [1243], [1324], [1342], [1423], [1432].

As all P sequences use the same numbers 1 to k a conflicting correlation is still possible, but only in one number in a time period. For example, when the system is empty and the cells a_1 to f_1 arrive they will accumulate at DRAM 1. The PRR-MMA will distribute further arriving cells equally to the remaining DRAMs. Fig. 3(b) shows the state after the sequential arrival of the cells a_1 to f_1, a_2 to f_2.

To accumulate Q cells in DRAM queue 2 beginning from the state in Fig. 3(b), we have to receive another 6 cells. As all DRAMs have cells to processes the total amount of cells in the SRAM does not increase (c. f. Fig. 3(c)).

We found by systematic testing, that for $Q = n \cdot P$ with $n = 1, 2, \ldots$ the maximal amount of necessary tail SRAM is $Q \cdot (2 - 1/k)$ cells. In contrast, [3] requires $Q \cdot (k-1)$ cells. For a given system the number of used DRAMs k can be varied between a maximal value $\hat{k}$ and 1. With $k = \hat{k}$ the system can use minimum size cells. With $k = 1$ the system has to use blocks of size $B = k$ cells what is equal to the system in Fig. 1. For a realistic scenario with $R = 100$ Gbps and DDR3-DRAM the value of $\hat{k}$ is 21. For this system with $Q = 5000$ and $k = 7$ SRAM size is decreased by 73 %.

In our ongoing work, we concentrate on the following two topics. First, for $P > Q$ the Q used sequences have to be selected properly from the available set P. Second, if we reduce the head SRAM size by the same amount as on tail side, cells may be delivered out of order.

4. CONCLUSION

High speed routers and switches utilize hybrid SRAM/ DRAM packet buffers to meet both speed and capacity requirements. We reduce the amount of required tail SRAM significantly by our new PRR-MMA. Therefore, the PRR-MMA spreads the incoming data as equally as possible to the individual DRAMs so only little data can accumulate in the SRAM. Our architecture decreases required SRAM size of a 100 Gbps system with 5000 flows by 73 % while still providing worst-case bandwidth guarantee.

5. ACKNOWLEDGMENTS

This work was funded by Alcatel Lucent Bell Labs Germany, Stuttgart.

6. REFERENCES

[1] S. Iyer, et al., Analysis of a memory architecture for fast packet buffers. In *IEEE HPSR*, 2001.

[2] J. García, et al., Design and implementation of high-performance memory systems for future packet buffers. In *MICRO 36*, Washington, DC, 2003.

[3] F. Wang, et al., Using parallel DRAM to scale router buffers. *IEEE Transactions on Parallel and Distributed Systems*, 20:710–724, 2009.

[4] S. Kumar, et al., Design of randomized multichannel packet storage for high performance routers. In *HOTI 2005*, Washington, DC, USA, 2005.