Area and Energy-Efficient Buffer Designs for NoC based on Domain-Wall Memory

Jinzhí Lái¹², Jueping Cai¹, Ruilian Xie³, Jiao Guan¹

Abstract Networks-on-chip (NoC) is a major contributor to the power consumption in modern many-core processors, especially the router comprising large number of virtual channel (VC) first-in–first-out (FIFO) buffers. In this paper, we propose three buffer designs that leverage the unique serial access mechanism, non-volatility and high density of Domain-Wall Memory (DWM) to replace conventional SRAM based buffers in NoC router. Experiments demonstrates that the proposed DWM designs can achieve considerable improvement in area and power efficiency. The best performing proposed approach shows 36.1% (24.2%) area and 55.1% (24.5%) power saving over conventional SRAM (STT-MRAM) based designs respectively without performance degradation.

key words: Networks-on-chip (NoC), Router, Domain-Wall Memory, Buffer
Classification: Integrated circuits (memory)

1. Introduction

Network-on-Chip (NoC) has brought extendable and high bandwidth communication architecture for many-core design recently. Buffers in NoC can improve the network bandwidth utilization effectively [1]. However conventional SRAM based buffers dominate a significant portion of area and power consumption of NoC [2, 3]. With the continuous improvement of scaled technologies, the leakage power has rapidly become a domain component of the power dissipation in memories [4]. Many efforts have been made to reduce the power dissipation of buffer, such as bufferless NoC [5] and elastic-buffered NoC [6], which eliminate nearly all the buffer to save the power and area, but suffer from performance penalties.

Several non-volatile memory (NVM) technologies have been proposed as a potential replacement of conventional SRAM based memory because of their properties of high-density, low leakage power and non-volatility [7]. DWM is one of the most promising alternatives to conventional SRAM. Various applications based on DWM have been demonstrated [8, 9, 10, 11, 12]. However, most of the previous studies focused on the shifting schemes exploiting and storage redundancy to conquer shift delay and data loss. In this paper, we analyze both the characteristics of DWM device and the data flow of NoC router. We focus on the circuit-level approaches based on the access timing of DWM device and propose three area and energy-efficient DWM based NoC router buffer implementations. Then we perform extensive evaluations on area, power and performance to compare the proposed DWM based buffers with SRAM and STT-MRAM implementations.

2. Related work

2.1 Basic of Domain-Wall Memory

In essence, DWM consists of write port, read port and magnetic nanowire (NW) [13, 14, 15]. NW comprises multiple domains and each domain can store one single bit in the form of magnetization direction. Read and write ports are attached on the designated domain. As shown in Fig. 1(a), the read and write operations are applied similar to the magnetic tunnel-junctions (MTJs) [16] naturally formed between the NW and the fixed magnetic layer. The data stored in NW can be shifted in either direction by applying short current pulse on each end of NW. DWM performs accesses in a serial manner naturally analogous to a tape [17] (Fig. 1(b)). Multiple bits in DWM share read/write ports and the target bit needs to be shifted in a serial fashion through the
ferromagnetic nanowire to the appropriate position before read/write operation. Although DWM has high density by sharing read/write ports in multiple bits, it also leads to latency penalty because of the varying shift latency [18]. The read and write MTJs of DWM are functionally similar to STT-MRAM, which is another type of spintronic non-volatile memory and used to replace pure SRAM based NoC buffers [19], but suffers from high power MTJ write [20]. However, write operation of DWM can be performed by shifting the magnetization from the fixed domain to the free domain on the NW in a shift-based write scheme [21] to reduce the latency and power consumption at the cost of area overhead. The parameters (speed, and power) of DWM and STT-MRAM are given in Table I.

### Table I Characteristics of DWM and STT-MRAM at typical 1.2V 25°C 65-nm process [9].

|          | Speed  | Power  |
|----------|--------|--------|
|          | DWM    | STT-MRAM | DWM    | STT-MRAM |
| Read     | 2.81ns | 2.81ns  | 23.08μs | 23.08μs  |
| Write    | 3.9ns  | 3ns     | 55μs    | 120μs    |
| Shift    | 1ms-2ns| -       | 10μs    | -        |
| Shift_based Write | 1ns-2ns | -       | 28.4μs  | -        |

2.2 Basic of NoC router architecture

To investigate the NoC router buffer design, we employ a typical 4-stage wormhole switching router with virtual channel (VC) mechanism [22]. Each VC adopts a FIFO scheme to store and forward data (Fig. 7 (a)). A fixed amount of buffer hardware resource, a total of 64 flits (1K Bytes) for each input port buffer, is set to implement different partitions of VC buffer. BookSim2.0 [23], a cycle-accurate NoC simulator, is adopted for the evaluation using dimension-order routing algorithm, uniform traffic, and 8 × 8 mesh network under wormhole switching mechanism. Fig. 2 shows that 8-VCs and 8-flits buffer scheme leads to optimal performance of network. The simulations in Section 4 will be carried out based on the investigation in this section.

### 3. Designing buffers with DWM

A DWM bank data array organization DWM Block Clusters (DBC) [8] is used for the buffer designing. According to Table I, the read, shift-based write, and shift of DWM at 65-nm technology take 2.81 ns, 2 ns and 2 ns respectively. Shift operation can be merged into write operation under the constraint of clock cycle (200 MHz clock frequency) for the sake of being consistent with regular FIFO operation, as shown in Fig. 3.

#### Fig. 3 Timing diagram of DWM FIFO (@200 MHz clock frequency).

3.1 Single-Read-Port DWM Buffer (SRPB)

To coordinate with the 4-stage pipeline procedure with iso-capacity memory, a Single-Read-Port DWM Buffer (SRPB) with 16 VCs for each input port, 4 flits for each VC FIFO and 128 bits in width using DBCs, is proposed as shown in Fig. 7 (b). Each DBCs in SRPB is implemented by a group of 128 multibit-DWM cells. Each multibit-DWM cell holds 4 domains, one shift-based write port aligning to head domain, and a single read port aligning to tail domain. Access to the 128 DWM cells will be carried out simultaneously in parallel. Data is written into bank head and read out from bank tail, proceeding in serial. The address management logic used in conventional SRAM based FIFO is eliminated in DWM FIFO consequently. Shift operation is implemented after write operation if the bank is not full to make the leftmost domain valid for the subsequent flit. A novel credit update mechanism is used to maintain the credit counter in upstream router in order to avoid bank overflow. Read operation is activated when the current VC receives a grant. Then a flit will be read out from tail of the bank. DWM FIFO proceeds shift operation when there is flit data remaining and the bank is not full concurrently to prepare for reading. Shifting is inhibited when read/write is active or the rightmost domain is occupied to prevent data loss.

#### Fig. 2 Latency of different input port buffer partitioning.

3.2 Single-Read-Port DWM Buffer with Dual-bank (SRPBD)

According to the discussions in Section 2, we apply 8 VCs and 8-flit buffer to SRPB to improve its performance. In order to preserve the 4-stage pipeline procedure and minimize the shift distance, Single-Read-Port DWM Buffer with Dual-bank (SRPBD) based on SRPB is proposed as demonstrated in Fig. 7 (c). Each
bank of VC in SRPBD is half the length of buffer.

Domain status queue (DSQ), a single bit register queue equals to the length of VC, is used to maintain the occupation status of each bank in SRPBD. Each ‘1’ in DSQ indicates the corresponding domain is occupied by valid flit, and ‘0’ indicates the domain is free conversely. The queue keeps the same shifting pace with the data. The occupation of each bank can be concluded from the queue. The timing diagram of SRPBD is illustrated in Fig. 4. Like SRPB, shift is moved to write cycle in SRPBD. Appropriate bank for the incoming flits is determined by the write operation finite state machine (FSM) (Fig. 5 (a)). When a flit data reaches the rightmost position of the bank and there is no read applying on the current bank concurrently, there will trigger a full state assertion. The whole VC is full when the dual banks are both in full state. Write operation switches to the other available bank only when the current bank is full, which keeps the flits in order. Guaranteed by a well-designed credit mechanism, there will not be flits coming up when the VC is full. Read operation is controlled by a reading FSM (Fig. 5 (b)). Read operation switching to the other bank arises only when the current bank is already empty concluded by DSQ. All the flits come out in the succession as they were written into banks previously, and the sequence of flits retains consequently. Shift operation will stop when the bank is empty or full controlled by a shift FSM (Fig. 5 (c)) in each VC.

3.3 Multi-Read-Port DWM cell (MRPB)

In principle, both SRPB and SRPBD cannot make full use of the bank’s capacity because of the flit gaps induced by successive shift operations in the case of flit stall, which brings up a decline in performance. To address this inefficiency, we leverage multiple read-only ports multibit-DWM cell [8] to improve shift fashion and read efficiency. Fig. 7 (d) depicts the block diagram of the proposed Multi-Read-Port DWM Buffer (MRPB). According to analysis in Section 2, MRPB prolongs the VC’s depth to 8 and reduces the VC number to 8 based on SRPB. A shift based write port is placed on the leftmost domain and 8 read-only ports are distributed corresponding with all the domains in the multibit-DWM. Data can be read out from the current position without being shifted to the tail of bank. Write and shift operation are merged into a single cycle in MRPB like SRPB and SRPBD. Different from SRPB and SRPBD, shifting flit to the rightmost domain for reading is not necessary in MRPB, and flits stay continuously without gap consequently. The elimination of invalid domain occupation leads to full utilization of the buffer. When a FIFO read comes, the appropriate read port will be selected by the read port controller according to the DSQ. The read port corresponding to the headmost of valid data instead of the rightmost of the bank is selected. Excessive shifting is avoided consequently. Shift operation stops when the bank is empty or full.
6. With enough timing margin for read and write performing like an ordinary FIFO, MRPB helps cutting the extra latency introduced by shift operation, thereby reduces the average packet latency dramatically over SRPB and SRPBD. A read-only port can be realized by only one minimum sized transistor, and adding a read-only port to multi-bit DWM cell brings only 13% increase in area [8]. Thus, it is reasonable to utilize additional read ports to achieve a better tradeoff between performance and area overhead. Moreover, the credit-based flow control can be inherited from conventional wormhole router, making MRPB an ideal candidate for replacing the SRAM based buffer directly.

Table II. System configuration

| core count | 64 | topology | 8×8 2D mesh |
|------------|----|----------|-------------|
| L1 I & D cache | private, 32KB | router pipeline | 4-stage |
| L2 cache | shared, 512B/bank | VC count per port | SRPB: 16 VCs SRPBD: 8 VCs MRPB: 8 VCs |
| cacheline size | 64B | buffer depth per VC | SRPB: 4 flits SRPBD: 8 flits MRPB: 8 flits |
| frequency | 1GHz | packet length | 20 flits |

4. Experimental setup and results

4.1 Experimental setup

In order to evaluate area, power and performance of the proposed DWM buffers, we perform extensive evaluations to compare the proposed designs with SRAM and STT-MRAM implementations. NVSim [24] was used for area and power examinations of SRAM and STT-MRAM, and then a modified NVSim was used for the DWM based buffers using data from [21] [25]. A parametrized RTL FIFO control implementation [26] and modified versions for SRPB, SRPBD, and MRPB were synthesized using Synopsys Design Compiler (DC). STT-MRAM adopts the same FIFO control logic with SRAM [27], and they have identical performance accordingly. We employed BookSim2.0 to run different synthetic traffic patterns for latency simulation of SRAM and STT-MRAM based designs. And then a modified BookSim2.0 was used for DWM based designs. The application traces were collected from Netrace [28] by running PARSEC [29] benchmark suite in a full-system simulator M5 [30] with the system configurations as shown in Table II. The traces were then applied to a modified BookSim2.0 to estimate the performance of all the proposed designs and counter parts.

4.2 Experimental results

1) Area

Considering the embedded memory only, SRPB, SRPBD and MRPB achieve 76.1%, 76.1% and 50.8% area saving over SRAM based design respectively as shown in Fig. 8 (a). Compared to STT-MRAM, DWM based designs also gain 63.3%, 63.6% and 40.6% area reduction respectively. Different from SRAM and STT-MRAM, DWM can store multiple bits in a single cell (4-bit for SRPB and SRPBD, 8-bit for MRPB) and the peripheral circuit of DWM based memory decreases accordingly. Multiple read ports implementation in MRPB leads to a more area consumption among the DWM based designs.

As shown in Fig. 8 (b), SRPB and MRPB achieve 26.3% and 36.1% reduction in total area over SRAM. There are also 12.7% and 24.2% area saving compared to STT-MRAM. Relying on the shift characteristic, DWM based FIFO works in the absence of address counters and achieves a smaller area overhead in FIFO control logic. With the same memory architecture, the difference between SRPB and SRPBD lies in the FIFO control logic and a modified NVSim was used for the same FIFO control logic with SRAM.

The numerical result of power dissipation is shown in Table III. Static power comparison of different designs normalized to SRAM is demonstrated in Fig. 8 (c). Owing to the significant leakage power reduction, SRPB, SRPBD and MRPB show 76.1%, 80.5% and 55.1% static power saving over SRAM on average respectively. They also provide 60.8%, 68.6% and 26.5% static power reduction over STT-MRAM respectively. The dynamic energy parameters are reported in Table IV. Utilizing the shift-based write, DWM designs achieve a lower power write operation. MRPB performs the same read operation as STT-MRAM. Owing to less read ports and simpler memory peripheral circuit, both SRPB and SRPBD consume less energy than STT-MRAM and MRPB in read.

Table III. Buffer power for different technologies from synthesized control logic and a modified NVSim

| Memory Technology (1KB) | Static Power(uW) | Dynamic Energy(pJ/bit) |
|------------------------|------------------|------------------------|
| (Memory array & control) | Read | Write | Shift |
| SRAM (8 VCs) | 2.862 | 1.188 | 0.745 | — |
| STT-MRAM (8 VCs) | 1.747 | 0.441 | 1.058 | — |
| SRPB (16 VCs) | 0.684 | 0.337 | 0.209 | 0.209 |
| SRPBD (8 VCs) | 0.558 | 0.337 | 0.209 | 0.209 |
| MRPB (8 VCs) | 1.285 | 0.441 | 0.209 | 0.418 |

3) Performance

a) Synthetic traffic patterns

As seen in Fig. 9, SRPB and SRPBD perform similarly to SRAM at low injection rate, resulting from that flits can always gain grants in the absence of fierce competition in switch allocation (SA) stage and the flow of flits is consistent in the shift procedure of DWM memory. At medium and high injection rate, competitions intensify in SA stage, and flits cannot get the grants to pass through in time. When a body/tail flit fails in the bid of SA stage in upstream router while the
head flit has been written into DWM FIFO of downstream router, invalid domain occupations in DWM bank are introduced in continuous shifting, and a gap within a packet occurs consequently. As a result, the utilization of VC buffer drops, therefore, the latency rises sharply and the network steps into the saturation stage rapidly. Especially for SRPB with more VCs, the arbitration decreases the chance of getting grants for each VC, which worsens the latency further. Fig. 10 (a) shows the saturation throughput normalized to SRAM. Compared to SRAM, SRPB decreases the saturation throughput by 48.2%, 38.4%, 21.1% and 38.1% for uniform, transpose, shuffle and tornado traffic respectively, and 37.8% on average. On the other hand, SRPBD performs better than SRPB because of the optimization of control structure. However, it still encounters degradations of 37.9%, 30.7%, 10.5% and 33.3% for the corresponding traffic patterns, and 29.2% on average. In contrast, MRPB realizes bit-wise access similar to SRAM and gains identical performance with SRAM in synthetic traffic simulations.

b) Application traffic patterns

Fig. 10 (b) summarizes the average packet latency of the proposed designs normalized to SRAM in application traffic. MRPB achieves the same performance as SRAM as expected. In low contented traffic, such as blaskholes and x264, SRPB and SRPBD have a slight degeneration compared to SRAM. However, when encountering high contented traffic, such as fluidanimate and canneal, the performance of SRPB and SRPBD degrade significantly, because of the decrease in bank utilization as previously mentioned. In general, SRPB and SRPBD perform poorly in application traffic simulations, introducing throughput decrease by 42.7% and 29.6% compared to SRAM on average respectively.

c) Energy computation

The energy consumption is calculated by the count of access behavior, including read, write and shift operations obtained from a modified Booksim2.0 in each benchmark run and the access energy parameter in Table IV. Fig. 11 shows the energy consumptions of designs involved normalized to SRAM. Compared to SRAM (STT-MRAM), SRPB and SRPBD achieve energy saving by 36.1% (16.3%) and 41.3% (21.5%) on average respectively. Constrained by more shifts, SRPB and SRPBD perform worse than MRPB despite of their static power reduction. MRPB acquires an outstanding achievement with more than 50.1% reduction over SRAM and more than 30% reduction over STT-MRAM on average. We can conclude that MRPB is the best energy efficiency choice of the proposed designs.

4) Timing

In order to test the timing of VC FIFO using MRPB, an 1KB SRAM was generated by Memory Compiler, and a DWM liberty file was built using data from [25] based on the liberty file (.lib) of SRAM and then synthesized in DC with 65-nm CMOS technology at 200 MHz clock frequency combined with the FIFO control RTL for MRPB. Test result demonstrated that timing constraint was met without any violation in VC FIFO with MRPB.

5. Conclusion

Domain Wall Memory is one of the emerging spin-based nonvolatile memory technologies that has much higher density and better energy efficiency. In this paper, we propose three DWM based buffer designs, SRPB,
SRPB, and MRPB, to exploit area and energy efficiency buffer for NoC router. Experimental results demonstrate that the proposed designs can achieve significant area and power reduction over conventional SRAM and STT-MRAM based implementations. However, SRPB and SRPBD suffer from considerable network throughput degradation. Which is preferred, MRPB achieves considerable improvements in area and energy while maintaining the same performance as SRAM based implementation.

Acknowledgments

This work was supported by Natural Science Basic Research Plan in Shaanxi Province of China (2021ZDLGY02-01), and the Innovation Wisdom Base for Wide Bandgap Semiconductor and Micro-Nano Electronics of China (B12026).

References

[1] J. Hu, et al.: “System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25 (2006) 2919 (DOI: 10.1109/TCAD.2006.882474).

[2] Paul V. Gratz, et al.: “Implementation and Evaluation of On-Chip Network Architectures,” 2006 International Conference on Computer Design (ICCD) (2006) 477 (DOI:10.1109/ICCD.2006.4380859).

[3] E. A. Shipner, et al.: “On the Capacity of Bufferless Networks-on-Chip,” IEEE Transactions on Parallel and Distributed Systems 26 (2015) 492 (DOI: 10.1109/TPDS.2014.2310226).

[4] E. Morifiu, et al.: “Supply and threshold-Voltage trends for scaled logic and SRAM MOSFETS,” IEEE Transactions on Electron Devices 53 (2006) 1427 (DOI: 10.1109/TED.2006.874752).

[5] Y. Kao and H. J. Chao: “Design of a Bufferless Photonic Clos Network-on-Chip Architecture” IEEE Transactions on Computers 63 (2014) 764 (DOI: 10.1109/TC.2012.250).

[6] I. Seitandis et al.: “ElastiStore: Flexible Elastic Buffering for Virtual-Channel-Based Networks on Chip,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems 23 (2015) 3015 (DOI: 10.1109/TVLSI.2014.2383442).

[7] S. Mittal, et al.: “A Survey of Architectural Approaches for Managing Embedded DRAM and Non-Volatile On-Chip Caches,” IEEE Transactions on Parallel and Distributed Systems 26 (2015) 1524 (DOI:10.1109/TPDS.2014.2324563).

[8] Rangharajan Venkatesan, et al.: “Cache Design with Domain Wall Memory,” IEEE Transactions on Computers 65 (2016) 1010. (DOI:10.1109/TC.2015.2506581).

[9] J. Chung, et al.: “Exploiting Serial Access and Asymmetric Read/Write of Domain Wall Memory for Area and Energy-Efficient Digital Signal Processor Design,” IEEE Transactions on Circuits and Systems I: Regular Papers 63 (2016) 91 (DOI: 10.1109/TCSIS.2015.2497558).

[10] Mengjie Mao, et al.: “An Energy-efficient GPGPU Register File Architecture Using Racetrack Memory,” IEEE Transactions on Computers PP (2017) 1 (DOI:10.1109/TC.2017.2690855).

[11] Jr. Donald Kline, et al.: “Racetrack Queues for Extremely Low-Energy FIFOs,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26 (2018) 1531 (DOI: 10.1109/TVLSI.2018.2819945).

[12] Y. M. Chee et al.: “Coding for Racetrack Memories,” IEEE Transactions on Information Theory 64 (2018) 7094 (DOI: 10.1109/TIT.2018.2807480).

[13] S. Parkin, et al.: “Magnetic Domain-Wall Racetrack Memory,” Science 320 (2008) 190 (DOI: 10.1126/science.1145799).

[14] L. Thomas, et al.: “Racetrack Memory: A high-performance, low-cost, non-volatile memory based on magnetic domain walls,” 2011 International Electron Devices Meeting (2011) 24.2.1 (DOI: 10.1109/IEDM.2011.6131603).

[15] S. Parkin, and S. Yang: “Memory on the racetrack,” Nature nanotechnology 10 3 (2015) 195-8. (DOI: 10.1038/nnano.2015.41).

[16] J. Wang and H. Meng: “Programmable spintronics logic device based on a magnetic tunnel junction element,” Journal of Applied Physics 97(10) (2005). (DOI: 10.1063/1.1857655).

[17] Ranjan, et al.: “DyReCTape: A dynamically reconfigurable cache using domain wall memory tapes,” 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2015) 181 (DOI:10.7873/DATE.2015.0838).

[18] A. Attoofian and A. Saghir: “Shift-aware racetrack memory,” 2015 33rd IEEE International Conference on Computer Design (ICCD) (2015) 427 (DOI:10.1109/ICCD.2015.7357140).

[19] Jia Zhan, et al.: “Hybrid Drowsy SRAM and STT-RAM Buffer Designs for Dark-Silicon-Aware NoC,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24 (2016) 3041 (DOI: 10.1109/TVLSI.2016.2536747).

[20] K. Kuan and T. Adgebjua: “HALLS: An Energy-Efficient Highly Adaptable Last Level STT-RAM Cache for Multicoresystem,” IEEE Transactions on Computers 68 (2019) 1623 (DOI: 10.1109/TC.2019.2918153).

[21] Venkatesan, R., et al.: “DWM-TAPESTRI - An energy efficient all-spin cache using domain wall shift based writes.” 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2013) 1825 (DOI:10.7873/DATE.2013.365).

[22] Daily W and Towles B. Principles and Practices of Interconnection Networks [M]. 2003.

[23] Jiang, Nan et al.: “A detailed and flexible cycle-accurate Network-on-Chip simulator,” 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS) (2013) 86 (DOI:10.1109/ISPASS.2013.6557149).

[24] X. Dong, et al.: “NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 31 (2012) 994 (DOI:10.1109/TCAD.2012.2189390).

[25] Chao Zhang, et al.: “Quantitative modeling of racetrack memory, a tradeoff among area, performance, and power,” The 20th Asia and South Pacific Design Automation Conference (ASP-DAC), (2015) 100 (DOI:10.1109/ASPDAC.2015.7058988).

[26] D. U. Becker: “Efficient microarchitecture for network-on-chip routers”, Ph.D Dissertation, Stanford University, Stanford, Calif, USA (2012).

[27] H. Jang, et al.: “A Hybrid Buffer Design with STT-MRAM for On-Chip Interconnects,” 2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip (2012) 193 (DOI:10.1109/NOC.2012.30).

[28] J. Hestness, et al.: “Netrace: dependency-driven, trace-based Network-on-Chip simulation,” Proceedings of the 3rd International Workshop on Network on Chip Architectures (NoCRec) (2010) 31 (DOI:10.1145/1921249.1921258).

[29] C. Biena, et al.: “The PARSEC benchmark suite: Characterization and architectural implications,” 2008 International Conference on Parallel Architectures and Compilation Techniques (PACT) (2008) 72 (DOI:10.1145/1454115.1454128).

[30] N. L. Binkert, et al.: “The M5 simulator: modeling networked systems,” IEEE Micro 26 (2006) 52 (DOI:10.1109/MM.2006.82).