MeSAP: A fast analytic power model for DRAM memories

Sandeep Poddar*, Rik Jongerius*, Leandro Fiorin*, Giovanni Mariani*,
Gero Dittmann†, Andreea Anghel†, Henk Corporaal‡
*IBM Research, The Netherlands †IBM Research – Zurich, Switzerland
‡Eindhoven University of Technology, The Netherlands

***Abstract*—The design of an energy-efficient memory subsystem is one of the key issues that system architects face today. To achieve this goal, architects usually rely on system simulators and trace-based DRAM power models. However, their long execution time makes the approach infeasible for the design-space exploration of next-generation exascale computing systems. Analytic models, in contrast, are orders of magnitude faster.**

In this paper, we propose a new analytic memory-scheduler-agnostic power model for DRAM, henceforth referred to as MeSAP. Similarly to state-of-the-art trace-based approaches, our analytic model achieves an average error of 20%, while being an order of magnitude faster. Furthermore, we integrate MeSAP into an analytic performance model of general-purpose processors and show its applicability to the design of a computing system targeting scientific image processing applications.

I. Introduction

Today, system architects face a huge design space in their pursuit of energy-efficient exascale computing. To explore the design space, architects often rely on micro-architectural and trace-based simulators. However, their limited simulation speed makes this approach infeasible when dealing with the design of exascale systems [1]. Higher analysis speeds can only be achieved through abstraction, often at the cost of a reduced accuracy. Recently, the use of analytic processor performance models was proposed as a solution [2], [3]: analytic models can analyze different design points orders of magnitude faster than simulators, while providing better accuracy than back-of-the-envelope calculations.

The design of an energy-efficient memory subsystem is one of the issues that system architects face. An analytic memory-power model is a crucial component of an analytic system model in order to analyze power-performance trade-offs.

In this work, we present a new analytic power model for DRAM memories. The model is able to analyze design points fast and can be integrated with analytic performance models of targeted processors. The proposed model decouples the problem of power estimation from the performance analysis and the prediction of parameters of future memory technologies, as they are separate problems. This separation can allow a better understanding of the source of modeling errors. To achieve a fast analysis speed, it takes as input the average memory-access statistics of applications instead of their memory traces. The model is therefore agnostic of the memory scheduler.

This work makes the following contributions:

- An analytic memory-scheduler-agnostic power model, referred to as MeSAP, for the fast prediction of the power consumption of a DRAM memory;
- The validation of the model with respect to measurements of two memory systems, and its comparison with two existing memory power models;
- The integration of the model with a state-of-the-art analytic processor performance model and a tool to predict the parameters of memory technology. By using this tool chain, we perform an example design-space exploration targeting scientific image processing applications.

The remainder of this paper is organized as follows. Section II discusses related work on power modeling of memory systems. Section III presents the proposed analytic DRAM model. The model is validated in Section IV, with respect to measurements on a DDR3 and DDR4 memory and compared to two existing memory power models. Section V presents a case study in which we find an energy-efficient memory design for large 2D FFTs. Section VI concludes the paper.

II. Related work

Several tools and simulators for predicting memory power can be found in the literature. DRAMPower is a cycle-accurate DRAM power model for DDR memory architectures [4]. It is integrated with the gem5 processor simulator [5] and takes as input a memory trace. Similarly, Joo et al. [6] introduced a power model based on trace analysis for older memory types. As the runtime of trace-based methods increases linearly with the size of the memory trace, they are not suitable for analyzing very large systems and long-running workloads. Furthermore, analytic performance models usually do not provide traces but report average performance numbers.

CACTI is a widely-used tool to predict power consumption of memory systems [7]. It incorporates a detailed model at the level of the memory cells. However, we show in this paper that its accuracy is insufficient for our purposes. Vogelsang [8] proposes a detailed model at the circuit level to address the shortcomings of CACTI. However, it requires detailed circuit parameters, which are not always available from memory vendors [4], and operates at a level of detail which is too high for our goal of system-level analysis.

Various analytic power models similar to our approach are available as well in the literature. Joshi et al. [9] presented a model for older generations of SDRAM. In contrast, we model modern DDR3 and DDR4 memories. The spreadsheets from Micron [10] are a widely-used memory model, which offer

This work was conducted in the context of the joint ASTRON and IBM DOME project and was funded by the Dutch Ministry of Economische Zaken, and the Province of Drenthe.

Intel and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Other product or service names may be trademarks or service marks of IBM or other companies.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

a fast analysis speed similar to the one of MeSAP. However, they need application memory-access statistics specified at a level of abstraction that are not generated by current analytic processor performance tools. Examples are the page hit rate or the percentage of time all banks are precharged, statistics which, in contrast to our approach, requires modeling of the memory scheduler. The parameters required by our model are provided by state-of-the-art processor models, such as those presented by Van den Steen et al. [3] and Jongerius et al. [2].

MeSAP decouples the prediction of power consumption from the prediction of memory technology parameters. Shih et al. [11] presented DArT, an area, timing, and power model for DRAM. The power numbers presented in their paper are actually currents consumed while the memory chips operate in certain DRAM states, independent of the application behavior. Such current values form the input for MeSAP. Recently, Naji et al. [12] introduced DRAMSpec, a model similar to DArT. In contrast to DArT, DRAMSpec predicts current values for more DRAM states. In this paper, we employ DRAMSpec to generate memory technology parameters.

III. Proposed DRAM power model

This section introduces MeSAP. Before presenting it, we briefly discuss the background of DRAM technology.

A. DRAM architecture basics

In a commercial server system, a DRAM memory storage consists of DIMMs with one or more ranks. Multiple DRAM memory chips are combined to form a rank. A single memory chip comprises a set of banks and each bank contains memory elements arranged in rows and columns [13], as shown in Figure 1. The interface of a DRAM consists of command, address and data buses. To access a bank, the content of the required row is copied into the row buffer by issuing an *ACT* command. This opens the required row. The contents of the row can then be read using *RD* commands or data can be written into the row using *WR* commands. After a read or a write operation is completed, the content of the row buffer is copied back to the open row by issuing a *PRE* command. In addition, DRAM must be periodically refreshed to preserve the stored data, using a *REF* command.

A memory controller manages access to DRAM rows. There are three different row-buffer management policies [13]: close-page, open-page, and hybrid. A close-page policy favors random accesses and patterns with low degrees of access locality. An open-page policy favors access patterns with high degrees of temporal and spatial locality. A hybrid policy is suitable for workloads whose memory request rate and access patterns change during run-time.

B. Power model

The input of our model is a set of architectural parameters and application statistics (Table I), and specifications of a memory chip based on JEDEC standards (Table II). The basic block of our model is a *rank*, composed of n_c chips. We model 1) the active energy per rank, associated with individual memory commands, 2) the background energy per rank, consumed in each power state, 3) the I/O power and the termination power, and 4) the scaling of power per rank to the total system memory. However, we do not take into account the memory controller and the scheduling. This choice allows the decoupling of the DRAM power model from the performance analysis, significantly increasing the speed of the overall analysis, at the cost of reduced accuracy.

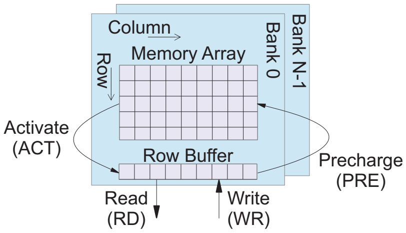

Fig. 1. Basic view of DRAM architecture with commands.

TABLE I
Architectural parameters of a memory subsystem and application statistics.

Symbol	Description	Unit
f_c	CPU clock frequency	Hz
n_d	DIMMs in a system	-
n_{dr}	Ranks in a DIMM (single or dual)	-
s_r	Size of a rank	GB
n_c	Number of memory chips in a rank	-
t_c	Application execution cycles w.r.t. CPU clock	cycles
n_r	Bytes read from DRAM	bytes
n_w	Bytes written to DRAM	bytes

TABLE II
Specifications of a single memory chip.

Symbol	Description	Unit
m_t	Memory type, e.g. DDR3, DDR4, HMC	-
s_c	Size of a memory chip	MB
b_c	Data bus width of a memory chip	bits
d_r	Single or dual data rate of DRAM	1/cycle
b_l	Maximum supported burst length	-
f_m	Memory bus clock frequency	Hz
V_{DD}	Nominal operating voltage	V
t_{REFI}	Refresh interval	cycles
t_{RFC}	Refresh cycle	cycles
t_{RAS}	Row access strobe	cycles
t_{RP}	Row precharge	cycles
idd_0	Operating one-bank active-precharge current	mA
idd_{2n}	Precharge standby current	mA
idd_{3n}	Active standby current	mA
idd_{4r}	Burst read current	mA
idd_{4w}	Burst write current	mA
idd_{5b}	Refresh current	mA

The total number of ranks is given by $n_{dev} = \sum_{i=1}^{n_d} n_{dr}^i$, where n_d is the number of DIMMs, and n_{dr}^i is the number of ranks in DIMM i. The cycles elapsed during the execution of an application with reference to the memory bus clock frequency, f_m, is $t_m = t_c f_m / f_c$, where t_c is the number of elapsed CPU cycles, and f_c is the CPU clock frequency.

We assume that each transaction happens in bursts and that the size of a transaction, b_s, in bytes, is given by

$$b_s = n_c \cdot b_l \cdot b_c / 8, \qquad (1)$$

where n_c is the number of chips within a rank, b_c (in bits) is the data bus width of a memory chip, and b_l is the maximum supported burst length. Both b_l and b_c are a power of 2.

1) Active energy per rank: To estimate active energy consumption per rank, we calculate the required number of read, write, activate, precharge, and refresh commands per rank. We assume that the number of bytes read and written are distributed evenly across n_{dev} ranks. Therefore, for a memory transaction in maximum burst length mode, read and write commands per rank, namely n_{RD} and n_{WR}, are given by

$$n_{RD} = n_r/(b_s \cdot n_{dev}), \text{ and} \tag{2}$$

$$n_{WR} = n_w/(b_s \cdot n_{dev}). \tag{3}$$

We assume a close-page policy and a burst refresh every refresh interval. Hence, on a rank level, the number of activate (n_A) and precharge (n_P) commands is

$$n_A = n_P = n_{RD} + n_{WR}, \tag{4}$$

while the required number of refresh (n_{REF}) commands is

$$n_{REF} = t_m/t_{REFI}. \tag{5}$$

The active energy E_{ar} is calculated, as described in [4], as

$$E_{ar} = \frac{V_{DD} \cdot n_c}{f_m} \sum \begin{cases} n_{RD} \cdot t_b \cdot (idd_{4r} - idd_{3n}) \\ n_{WR} \cdot t_b \cdot (idd_{4w} - idd_{3n}) \\ n_A \cdot t_{RAS} \cdot (idd_0 - idd_{3n}) \\ n_P \cdot t_{RP} \cdot (idd_0 - idd_{2n}) \\ n_{REF} \cdot t_{RFC} \cdot (idd_{5b} - idd_{3n}) \end{cases}, \tag{6}$$

where $t_b = b_l/d_r$ is the time, in cycles, taken by a burst read or write from an active bank.

2) Background energy per rank: The background (or static) energy consumption in a DRAM is determined by the amount of cycles spent in different power states. The model is based on the DRAM state diagram shown in Figure 2, i.e., a simplified version of the diagram presented by Goossens et al. [14] with the power-down states removed. The state diagram consists of 3 states: PRECHARGE, ACTIVE$_1$, and ACTIVE$_2$. The memory remains in the PRECHARGE state all the time, except when data is being read or written, or when the memory is refreshed. During a read/write transaction, the memory goes through the steps of cycle 1 in Figure 2. An *ACT* command activates a row of a memory bank and changes the memory state from the PRECHARGE to the ACTIVE$_1$ state. Data can be read or written in the ACTIVE$_1$ state using a *RD* or *WR* command, respectively. After the reading or writing of data is completed, the row is closed using a *PRE* command (or *PREA* command for more than one bank). A single read or write transaction is assumed to complete in $t_{RC} = t_{RAS} + t_{RP}$ cycles.

The cycle time, t_{RC}, is a minimum, optimistic value. The cycle time can increase in two cases: when more than one bank in a memory chip is active at a given time, or when the data bus is shared between memory chips. Figure 3 shows three possible scenarios for two consecutive read transactions from different banks of a memory chip. In scenario 1, the transactions from both banks are completely sequential. In scenario 2, the transactions overlap but as soon as bank 1 is ready to read, bank 0 has finished. In scenario 3, in contrast, bank 0 has not finished the transaction, but bank 1 is active and ready to read. This means that bank 1 has to wait for a time $t_{penalty}$ until bank 0 frees the data bus. During this time, bank 1 remains active and will dissipate extra background energy. For the calculation of the background energy, we assume that only one bank in a memory chip is active at a given time, and we neglect both scenarios 2 and 3, underestimating therefore the power consumption. With these assumptions, $(n_{RD} + n_{WR}) \cdot t_{RC}$ is the number of cycles required for $n_{RD} + n_{WR}$ transactions.

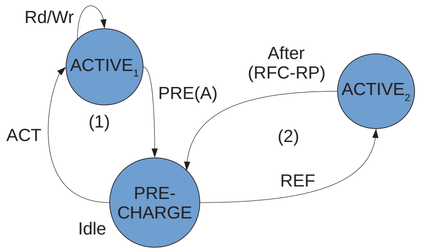

Fig. 2. Simplified state diagram for DRAM.

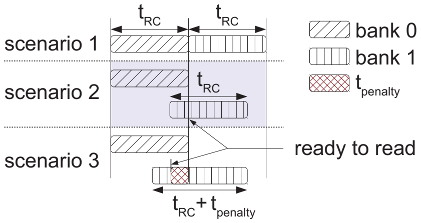

Fig. 3. Timing of consecutive transactions for three scenarios.

The number of cycles spend on read and write transactions, n_{ac}, is derived as

$$n_{ac} = \min \begin{cases} (n_{RD} + n_{WR}) \cdot t_{RC} \\ t_m - (n_{REF} \cdot (t_{RP} + t_{RFC})) \end{cases}. \tag{7}$$

The first line represents the number of cycles required for strictly sequential transactions, while the second line represents the time to schedule all transactions after subtracting the time required for mandatory refresh. We assume that transactions are schedulable and, as the total elapsed time can not exceed the total number of memory cycles, t_m, we calculate n_{ac} as the minimum of the two. Out of the n_{ac} cycles, a fraction $\frac{t_{RAS}}{t_{RC}}$ of background cycles is spend in the ACTIVE$_1$ state consuming active background current idd_{3n} and a fraction $\frac{t_{RP}}{t_{RC}}$ of background cycles consuming precharge background current idd_{2n}.

During a refresh, the memory goes through cycle 2 in Figure 2. A *REF* command changes the memory state from the PRECHARGE to the ACTIVE$_2$ state. A refresh completes in t_{RFC} cycles, out of which t_{RP} cycles consume idd_{2n} and the remaining cycles consume idd_{3n} [4]. Furthermore, we take into account that the memory has to be in the PRECHARGE state for t_{RP} cycles before it can refresh. As we assume a close-page policy, we do not have to issue the *PRE* or *PREA* commands before a refresh.

The number of background cycles, n_{ab}, for which memory is dissipating active background current, idd_{3n}, is

$$n_{ab} = n_{ac} \cdot \frac{t_{RAS}}{t_{RC}} + n_{REF} \cdot (t_{RFC} - t_{RP}). \qquad (8)$$

The number of background cycles, n_{pb}, for which memory is dissipating precharge background current, idd_{2n}, is equal to

$$n_{pb} = n_{ac} \cdot \frac{t_{RP}}{t_{RC}} + n_{REF} \cdot (t_{RP} + t_{RP}) + n_{pc}, \qquad (9)$$

where n_{pc} are the remaining idle cycles in the PRECHARGE state, unrelated to transactions or refresh, derived as

$$n_{pc} = t_m - (n_{ac} + n_{REF} \cdot (t_{RP} + t_{RFC})). \qquad (10)$$

The background energy E_{br} is, as described in [4],

$$E_{br} = \frac{V_{DD} \cdot n_c}{f_m} \sum \begin{cases} n_{ab} \cdot idd_{3n} \\ n_{pb} \cdot idd_{2n} \end{cases}. \qquad (11)$$

3) I/O and termination power: The I/O and termination power is calculated as given by Chandrasekar et al. [4]. The I/O power associated with a read operation is the product of the I/O power per bit, the number of bits read, and the number of data strobes. The termination power associated with a write operation is the product of the termination power per bit, the number of bits written, and the number of data strobes. The data strobes depend on the width of the DRAM, and the I/O and termination power per bit is determined by the circuit that drives input and output bits, respectively. These values can be obtained from, for example, Micron technology reports [10].

4) Scaling of the power to the total system memory: The total power consumed by the DRAM is given by the sum of active power for all ranks ($n_{dev} \cdot E_{ar} \cdot \frac{f_m}{t_m}$), the background power for all ranks ($n_{dev} \cdot E_{br} \cdot \frac{f_m}{t_m}$), the I/O power, and the termination power.

C. Integration with performance and technology models

To provide a complete framework for the design-space exploration of energy-efficient memory subsystems, we integrated MeSAP with the analytic processor performance model introduced by Jongerius et al. [2], and with DRAMSpec [12], a memory technology model. The performance model provides the application statistics needed as input for MeSAP, while DRAMSpec provides an infrastructure to generate the JEDEC standard currents and timing parameters for the different memory technologies used by our model.

In Section V, we use this infrastructure for exploring the design space of a memory subsystem for a targeted application.

IV. VALIDATION

We validated our model by comparing the accuracy of MeSAP and existing memory power models against measurements performed on two types of memory, i.e., a 48 GB DDR3-1600 and a 32 GB DDR4-2133 memory on an Intel® Xeon® E5-4650 and Intel® Xeon® E5-2697 v3 platform, respectively. While executing the workload on the Intel® Xeon® processors, we measure DRAM power using the likwid-powermeter tool [15].

We executed 21 applications from the PolyBench/C v3.2 benchmark set [16] on the gem5 simulator [5] to generate the input statistics fed to the power models. We compared our results (MeSAP-gem5) against DRAMPower v4.0 (trace-based, integrated with the gem5 simulator) and CACTI v5.3 (the latest version specifically targeting commodity DRAM).

TABLE III
AVERAGE ERROR AND CORRELATION OF DIFFERENT MODELS WITH RESPECT TO MEASUREMENTS.

	DDR3		DDR4	
	Error	**Corr.**	**Error**	**Corr.**
MeSAP-gem5	19.7%	0.82	16.7%	0.92
MeSAP-ICCD	18.6%	0.66	17.9%	0.69
CACTI	17.9%	0.82	49.7%	0.92
DRAMPower	19.2%	0.80	22.6%	0.92

We configured CACTI for a memory chip of 512 MB, as it is available in the measured platforms, and for a burst length of 8 bytes. CACTI outputs the static power per bank, p_s, and the dynamic read and write energy per burst access (E_r, and E_w, respectively). We calculated the power as

$$(p_s \cdot n_b \cdot s_d / s_m) + (N_r \cdot E_r + N_w \cdot E_w), \qquad (12)$$

where n_b is the number of banks, s_d is the total system memory, s_m is the size of a memory chip, and N_r and N_w are the number of read and write burst accesses per second.

Figure 4 and 5 show the results for the DDR3 and the DDR4 memory module, respectively. Table III summarizes the average error and correlation. With respect to the measurements, the MeSAP-gem5 results show a good correlation of 0.815 for DDR3 and 0.921 for DDR4. For DDR3, the average error is 19.7%, while the worst-case error is 35.5%, obtained for the *gemver* application. For DDR4, the average error is 16.7%, and the worst-case error is 24.1%, obtained for the *adi* application.

MeSAP has a similar correlation as CACTI and DRAMPower for both memory types. For DDR3, the average error of CACTI is 17.9%, while it increases to 49.7% for DDR4. MeSAP and DRAMPower achieve similar accuracy. However, the abstraction level of our model is higher. DRAMPower is a cycle-accurate trace-based model, while MeSAP employs a fast analytic approach and uses average memory bandwidths. Note that both MeSAP and DRAMPower use JEDEC current parameters from device datasheets as input. These parameters reflect worst-case measures [4], leading to an overestimation of power consumption.

To evaluate the influence of using MeSAP with an analytic processor performance model instead of a simulator, we also generated input statistics with the model introduced by Jongerius et al. [2] (MeSAP-ICCD), and compared the results with those obtained with the MeSAP-gem5 configuration. As shows in Figure 4 and 5, when using MeSAP-ICCD, we achieve an average error, with respect to power measurements, of 18.6% for DDR3 and 17.9% for DDR4. The correlation decreases to 0.66 for DDR3 and 0.69 for DDR4. This reduction is not related to our model, but is due to the lower correlation of 0.7 of the DRAM bandwidth usage obtained with the analytic model, in contrast to a 0.9 correlation obtained with MeSAP-gem5. Improving the analytic performance model will increase the accuracy of the power predictions.

We performed two additional experiments to evaluate the implications of not using traces and of using an analytic

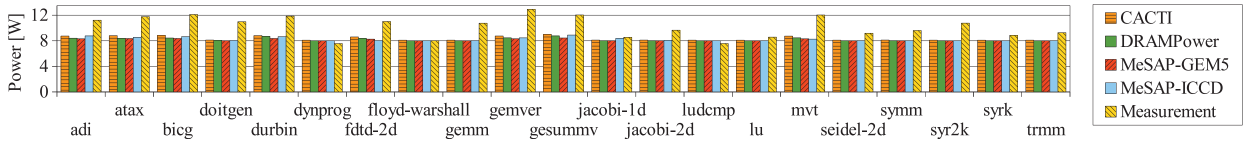

Fig. 4. Comparison of different DRAM power models with measurements of a 48-GB DDR3-1600 memory.

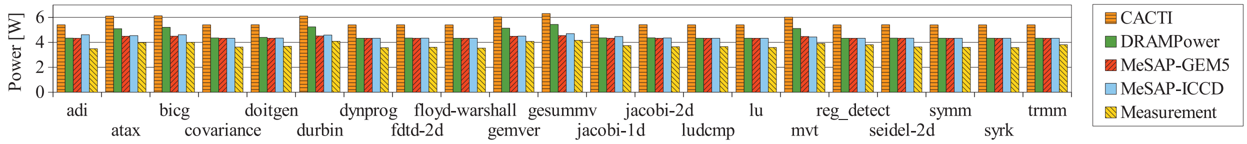

Fig. 5. Comparison of different DRAM power models with measurements of a 32-GB DDR4-2133 memory.

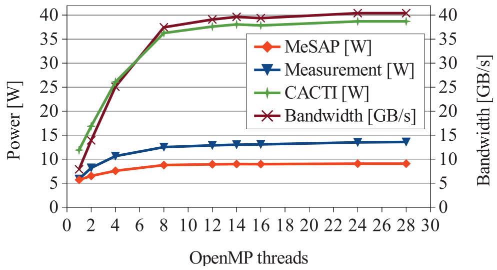

Fig. 6. Power estimation of the STREAM benchmark on DDR4.

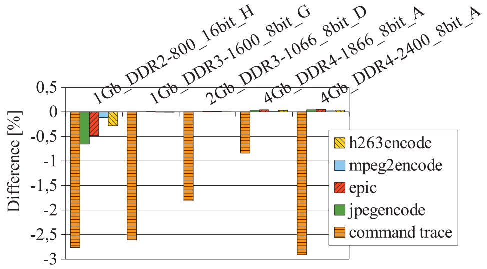

Fig. 7. Comparison of the analytic MeSAP model with the trace-based DRAMPower model.

performance model to generate input statistics. The results in Figure 4 and 5 show low variation in the power consumption. To understand the variation in power estimation with the bandwidth usage, in the first of the two experiments we ran the OpenMP version of the STREAM [17] benchmark on a Intel® Xeon® E5-2697 v3 platform with 32 GB DDR4-2133, for 1 to 28 threads. By using the likwid-perfctr tool [15], we obtained memory-access statistics used as input for MeSAP and CACTI (the statistics cannot be used with DRAMPower as it requires memory traces as input). Figure 6 shows that MeSAP and CACTI predictions follow the power measurements and bandwidth usage, with about a 0.99 correlation. The measurements show a strong correlation between the used bandwidth and the power consumption. This effect is captured using Equation 6. As the bandwidth is the only statistic that changes during the experiment, this leads to a strong correlation between the model and the measured results. MeSAP shows an average error of about 26.8% with respect to power measurements, while CACTI shows a high error of 165.5%.

In the second experiment, we analyzed the difference between an analytic and a trace-based model. DRAMPower can analyze either a transaction trace or a command trace. It is released with four example transaction traces (*h263encode*, *mpeg2encode*, *epic*, and *jpegencode*) and one command trace. A transaction trace comprises a sequence of read and write queries to the memory, while a command trace captures the memory commands issued by a memory controller. The command trace contains 3,040 commands, while the transaction traces contain between 92,905 and 512,856 transactions. We used these traces to compare power predictions of DRAMPower and MeSAP, both used as standalone tools. For MeSAP, we used memory-access statistics generated from the traces, such as the number of read and write transactions, and elapsed DRAM clock cycles. Note that it is easier to obtain memory statistics than traces.

Figure 7 shows the results. We obtain a prediction difference of up to 3% when comparing DRAMPower and MeSAP, due to the higher number of details about the memory scheduler contained in the command trace. For transaction traces, the differences are smaller. In our model, we assume that a transaction completes in t_{RC} cycles. Figure 3 shows that, if a transaction does not complete in t_{RC} cycles, a bank remains *active* for an additional time equal to $t_{penalty}$ and hence dissipates extra background energy. Furthermore, overlapping transactions that complete in t_{RC} cycles dissipate more background energy, which we do not consider. The behaviour of the application and the memory scheduler are captured in a command trace, information that is unavailable to our model.

The power estimation of MeSAP and DRAMPower is similar with up to 0.65% difference for the four transaction traces. DRAMPower also assumes a close-page policy for a transaction-level trace, which leads to similar accuracy and correlation. However, MeSAP has a constant analysis time of 0.18 seconds, while the average analysis time of DRAMPower is 9.94 seconds, which increases linearly with the trace size.

V. Case study

We use MeSAP, integrated with an analytic performance model [2] and DRAMSpec [12], to optimize a memory design for a specific application to show how it can be employed to improve energy efficiency of computing systems. The target application computes large 8k×8k 2D FFTs, used as important kernels in various scientific and image processing domains. Both DDR3 and hybrid memory cube (HMC) [18] memory technologies are explored. We use the methodology presented by [2] to generate memory-access statistics of the 2D FFT

TABLE IV
PARAMETERS PROVIDED AS INPUT TO DRAMSPEC TO GENERATE DIFFERENT MEMORY DESIGNS.

Parameter	**DDR3 / HMC**
3D die stacks	off (DDR3), on (HMC)
Vaults per layer	8, or 16 (only for HMC)
Number of banks	16, 32, or 64
Interface pins per vault	4, 8, 16, or 32
I/O bus frequency [MHz]	933, 1066, 1200, 1250, 1333, 1466, or 1600
Core frequency [MHz]	0, 200, or 320
Row buffer size [kB]	0.5, 1, 2, 4, or 16
Sub-array row buffer factor	0.5 or 2 (fixed to 2 for HMC)

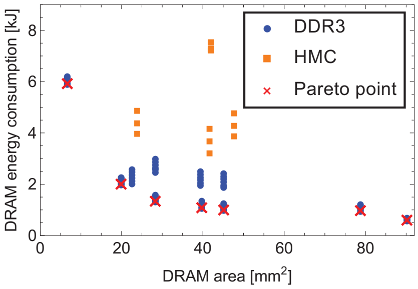

Fig. 8. Pareto graph of the design-space exploration of a 256-GB memory for a 2D FFT application.

kernel executing on an Intel® Xeon® E5-2697 v3 platform with 14 cores and 256 GB memory.

By varying the architectural parameters in Table IV, DRAMSpec generated a total of 87 valid configurations of 4-GB DDR3 and HMC memory chips that meet timing constraints. A total of 64 of such chips make up the 256-GB memory. A core frequency of 0 means that the frequency is determined by the DRAMSpec tool in a way that fits the architecture configuration. We used the 22-nm technology parameters provided with the tool.

To calculate the available DRAM bandwidth, we assume a dual memory interface. To consider the changes in the access latency t_{RC} for different designs, we start with a base DRAM latency of 70 ns plus the L3 latency of the processor for one DRAM design point and add the relative difference in t_{RC} between all DRAM designs.

The memory design-space exploration using our methodology is fast and takes approximately 52 seconds to explore 87 designs. Figure 8 shows a clear trade-off between DRAM energy consumption and area. We show the area for a 4-GB memory chip, as generated by DRAMSpec. While the total area depends on the system memory, this does not affect the trade-offs. In the design-space under consideration, the most energy-efficient memory design corresponds to a x4 DDR3, with an area of 90.15 mm^2, a 0.5-kB row-buffer size, a 320-MHz core-frequency, 64 banks, and a 1466-MHz I/O bus frequency. The results show that DDR3 architecture is more energy-efficient than a HMC solution. However, a single HMC can provide more than 15x the bandwidth of a DDR3 module while consuming 70% less energy-per-bit [18]. The 2D FFT application executing on the E5-2697 processor cannot exploit this bandwidth, which will lead to an inefficient design.

The results of the case study show a difference in energy consumption of up to 8×, showing how a good design of the memory subsystem can improve energy efficiency. Furthermore, the case study shows how MeSAP can be employed to explore different memory designs.

VI. CONCLUSION

This paper has presented a new analytic memory-scheduler-agnostic power model for DRAM, referred to as MeSAP. Our model is fast to analyze design points, two orders of magnitude faster than a trace-based power model, and enables the design-space exploration of the memory subsystem of exascale computing systems.

The model achieves a good correlation of 0.8 to 0.9 with an absolute average error of around 20%. We achieve a similar accuracy as trace-based approaches, while we are more accurate than CACTI, especially for DDR4.

As future work, we plan to extend the model for an open-page policy to increase the accuracy of the model for applications with high temporal and spatial locality.

ACKNOWLEDGMENT

The authors would like to thank Benny Åkesson for his valuable input to this work.

REFERENCES

[1] D. Unat *et al.*, "ExaSAT: An exascale co-design tool for performance modeling," *Int. J. High Perform. Comput. Appl.*, vol. 29, no. 2, pp. 209–232, May 2015.

[2] R. Jongerius *et al.*, "Analytic processor model for fast design-space exploration," in *Int. Conf. Comput. Des.*, ser. ICCD '15. IEEE, October 2015, pp. 411–414.

[3] S. Van den Steen *et al.*, "Analytical processor performance and power modeling using micro-architecture independent characteristics," *IEEE Trans. Comput.*, vol. PP, no. 99, pp. 1–1, March 2016.

[4] K. Chandrasekar, "High-level power estimation and optimization of DRAMs," Ph.D. dissertation, Delft University of Technology, 2014.

[5] N. Binkert *et al.*, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[6] Y. Joo *et al.*, "Energy exploration and reduction of SDRAM memory systems," in *Proc. Des. Autom. Conf.*, ser. DAC '02. ACM, 2002, pp. 892–897.

[7] S. Thoziyoor *et al.*, "A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies," in *Int. Symp. Comput. Archit.*, ser. ISCA '08, June 2008, pp. 51–62.

[8] T. Vogelsang, "Understanding the energy consumption of dynamic random access memories," in *Int. Symp. on Microarch.*, ser. MICRO '43. IEEE Computer Society, 2010, pp. 363–374.

[9] A. Joshi *et al.*, "Power modeling in SDRAMs," Tech. Rep. TR-040126-02, January 2004.

[10] Micron, "Calculating memory system power for DDR3 introduction," Tech. Rep. TN-41-01, 2007.

[11] H. C. Shih *et al.*, "DArT: A component-based DRAM area, power, and timing modeling tool," *IEEE Trans. on Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 9, pp. 1356–1369, September 2014.

[12] O. Naji *et al.*, "A high-level DRAM timing, power and area exploration tool," in *Int. Conf. on Embed. Comput. Syst.: Arch., Model., and Sim.*, ser. SAMOS '15, July 2015, pp. 149–156.

[13] J. Bruce, S. Ng, and D. Wang, *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann Publishers Inc., 2007.

[14] S. Goossens *et al.*, *Memory Controllers for Mixed-Time-Criticality Systems*. Springer International Publishing, 2016.

[15] J. Treibig, G. Hager, and G. Wellein, "LIKWID: A lightweight performance-oriented tool suite for x86 multicore environments," in *Int. Conf. Parallel Process. Workshops*, ser. ICPPW '10. IEEE, September 2010, pp. 207–216.

[16] L. Pouchet, "PolyBench/C: The polyhedral benchmark suite," 2012. [Online]. Available: http://web.cs.ucla.edu/ pouchet/software/polybench/

[17] J. McCalpin, "STREAM: Sustainable memory bandwidth in high performance computers," 2013. [Online]. Available: https://www.cs.virginia.edu/stream/

[18] "Hybrid memory cube consortium," 2016. [Online]. Available: http://www.hybridmemorycube.org/