



QUEEN'S
UNIVERSITY
BELFAST

ECIT

THE INSTITUTE
OF ELECTRONICS
COMMUNICATIONS AND
INFORMATION TECHNOLOGY

Workload-Aware DRAM Error Prediction using Machine Learning

**Lev Mukhanov, Konstantinos Tovletoglou, Hans Vandierendonck
Dimitrios S. Nikolopoulos and Georgios Karakonstantis**

Queen's University Belfast, United Kingdom

2019 IEEE International Symposium on Workload Characterization (IISWC)

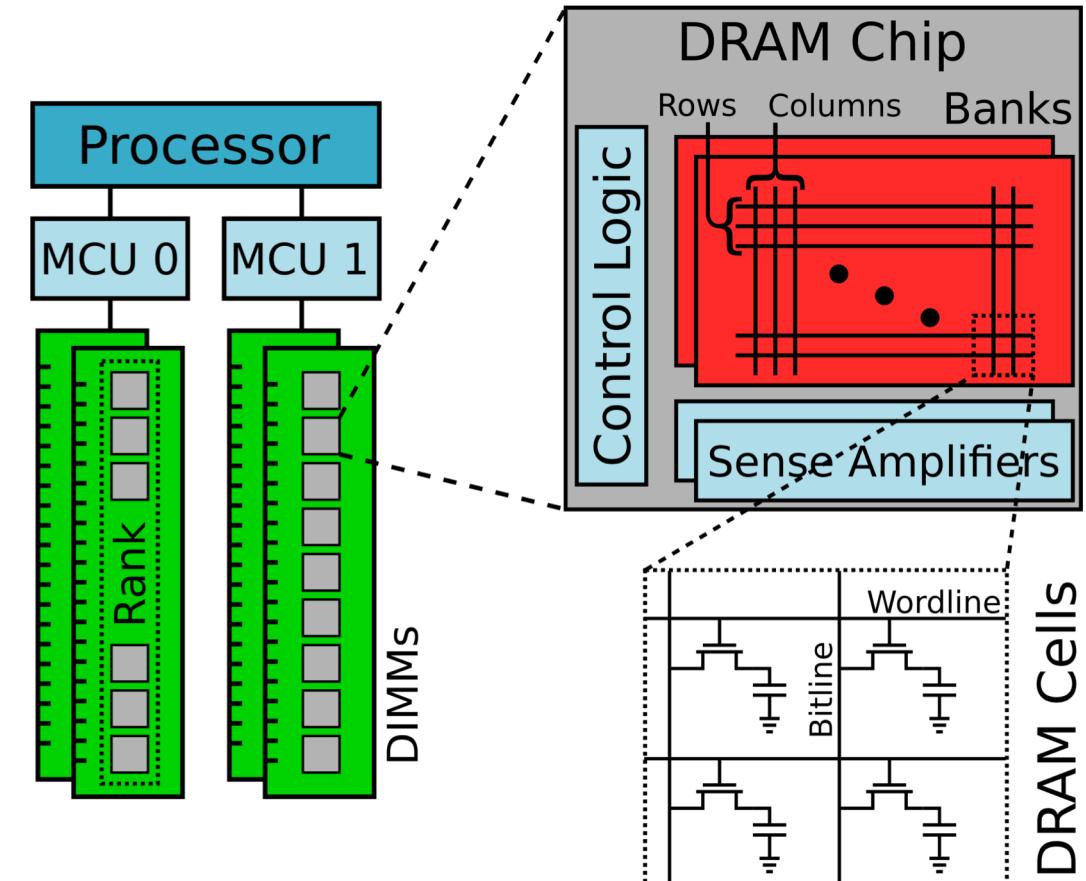
4 November 2019



Background - DRAM Organization



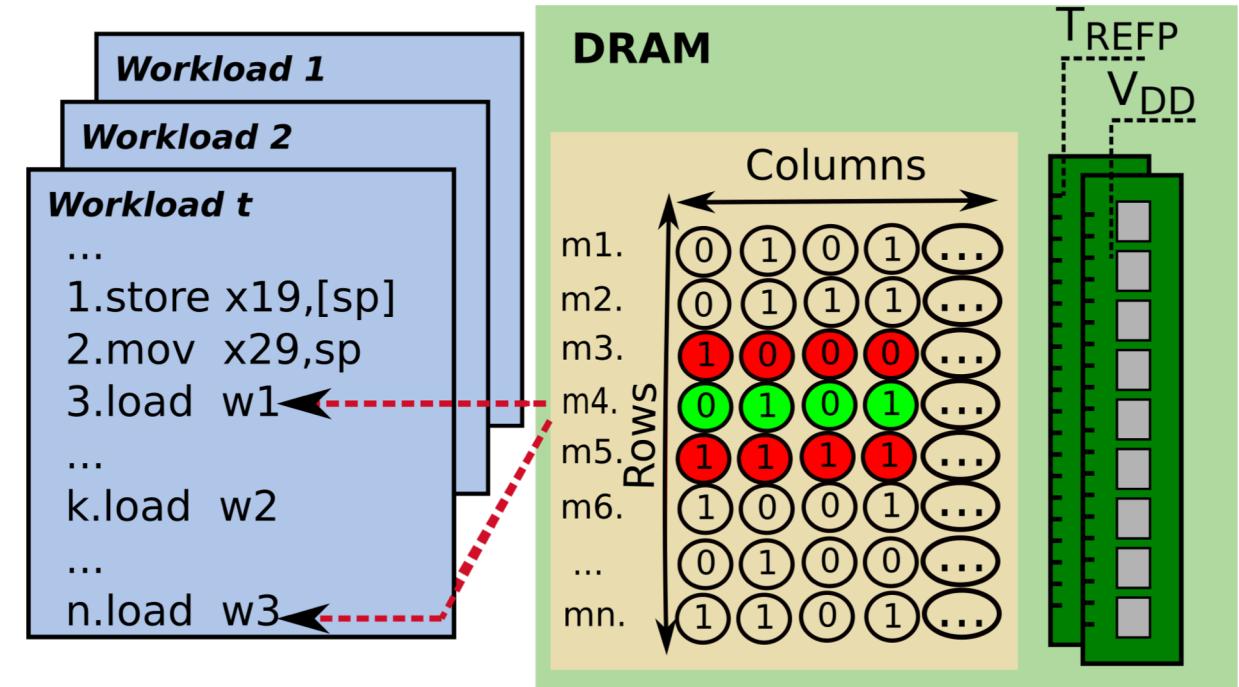
- Hierarchical DRAM organization
 - Memory Controllers (MCUs)
 - DIMMs, with two ranks of DRAM chips
 - Banks, two dimensional arrays (rows and columns) of cells
- DRAM cell consists of a:
 - Transistor
 - Capacitor



Background - DRAM Reliability Factors



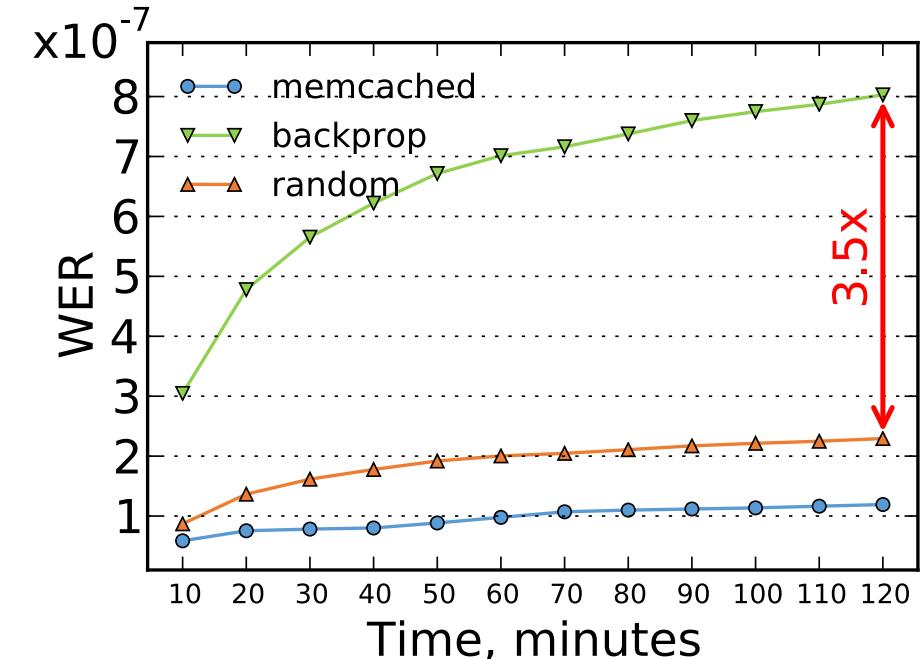
- DRAM error behavior depends on the:
 - i. circuit parameters (voltage (V_{DD}) and refresh rate (T_{REFP}))
 - ii. temperature
 - iii. specific DRAM device
 - iv. program inherent features
 - v. memory access patterns
 - vi. memory data patterns



Observations - DRAM Reliability Factors



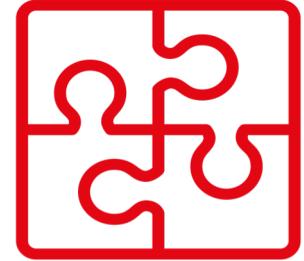
- DRAM error behavior may vary **across chips by 188x**
- The number of errors may differ by more than **3.5x between workloads** and micro-benchmarks
- The error rate can vary even between micro-benchmarks with **different data-pattern by 2x**
- **Frequent accesses** to memory may cause errors in neighbouring rows (Rowhammering)



Motivation for DRAM Error Behavior Model



- Evaluate how many errors will be manifested by a specific workload
- Measure the implicit impact of applied software optimizations (e.g. compiler, or thread level parallelism) on DRAM reliability
- Predict maintenance cycles
- Guide the adjustment of the circuit DRAM parameters for saving energy



Approach



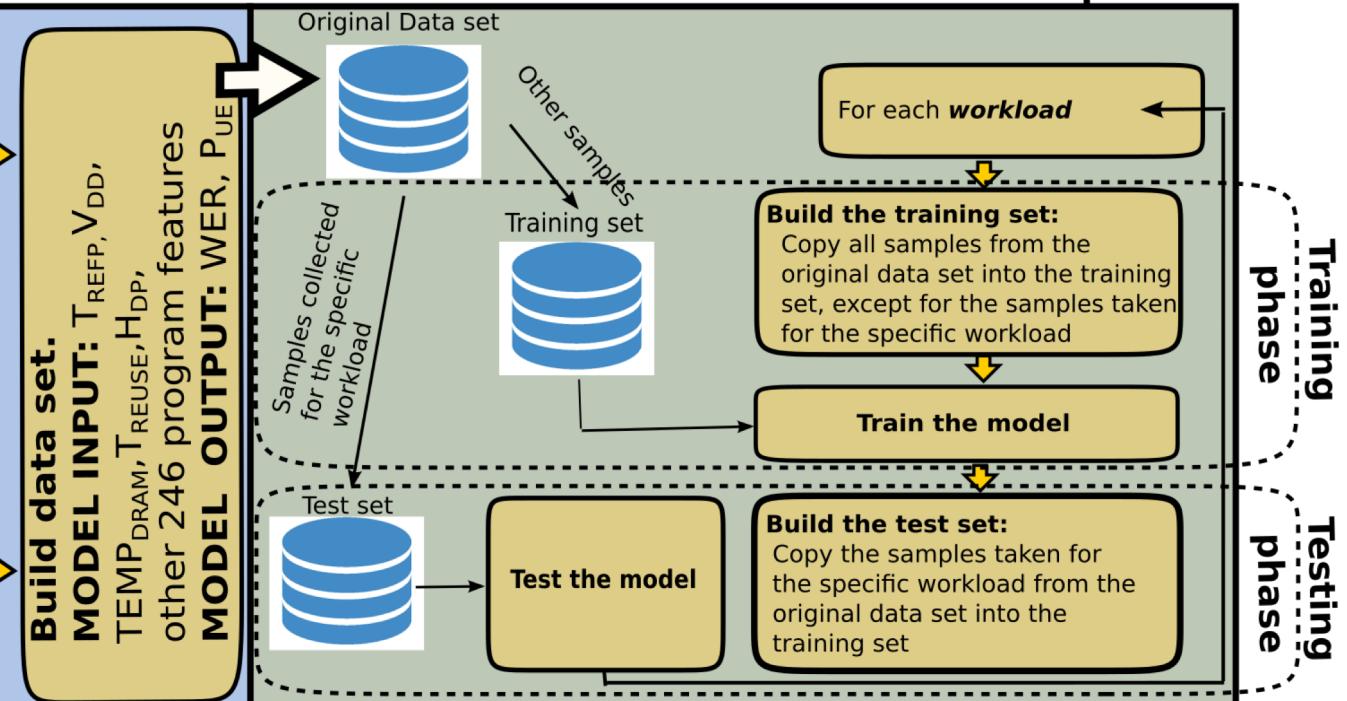
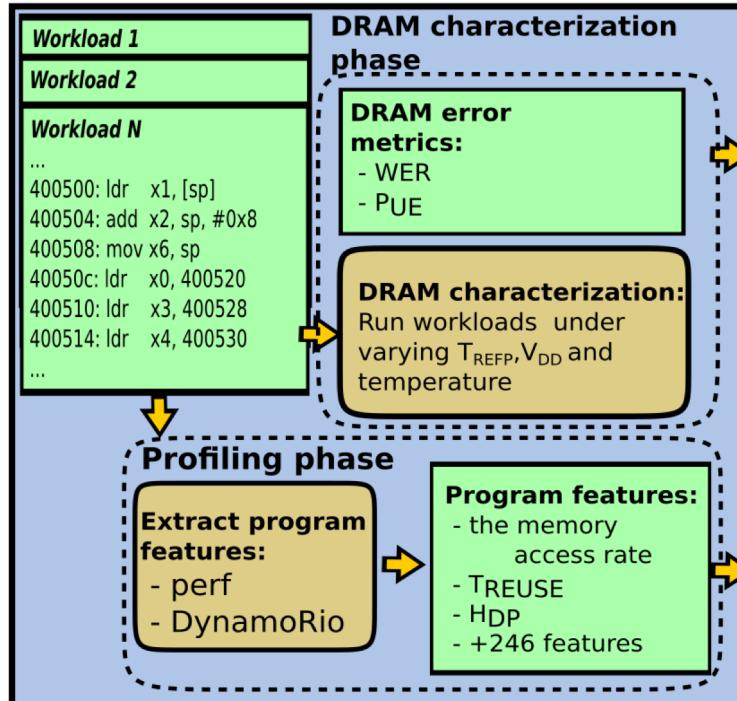
- Mathematical formulation of the problem space:
 $M_{ERR} = M(Features, DRAM\ Device, T_{REFP}, V_{DD}, TEMP_{DRAM})$
- Building an analytical model is extremely challenging due to the number of possible parameter combinations
- We use a supervised Machine Learning technique to automatically train such a model to predict DRAM error behavior



Methodology



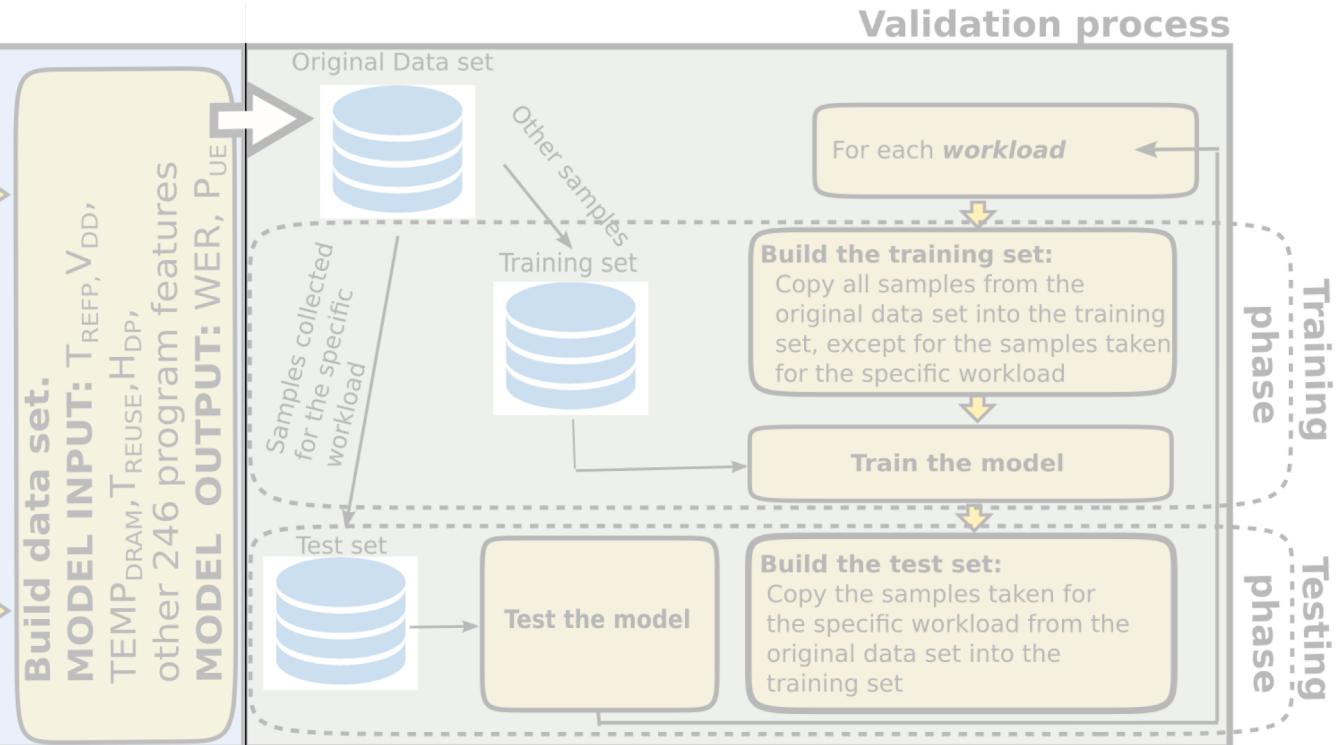
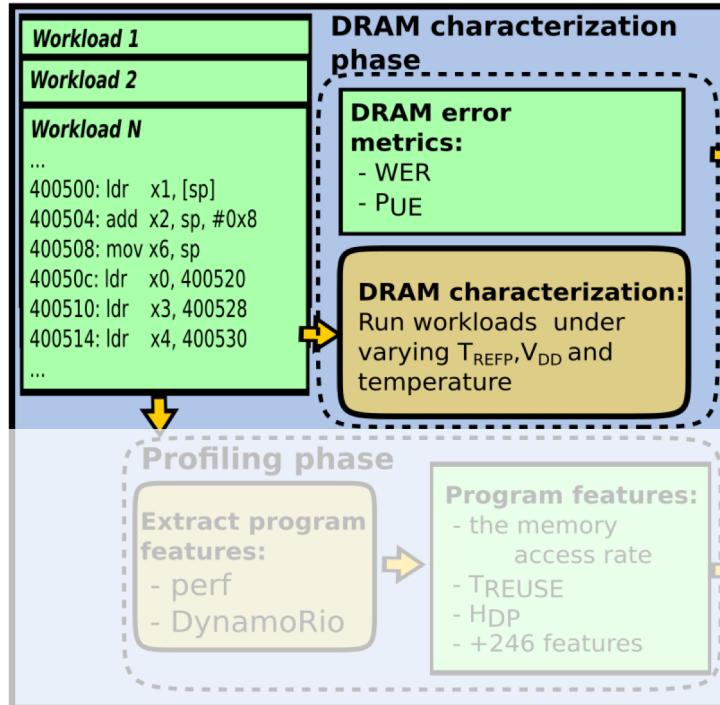
Data collection



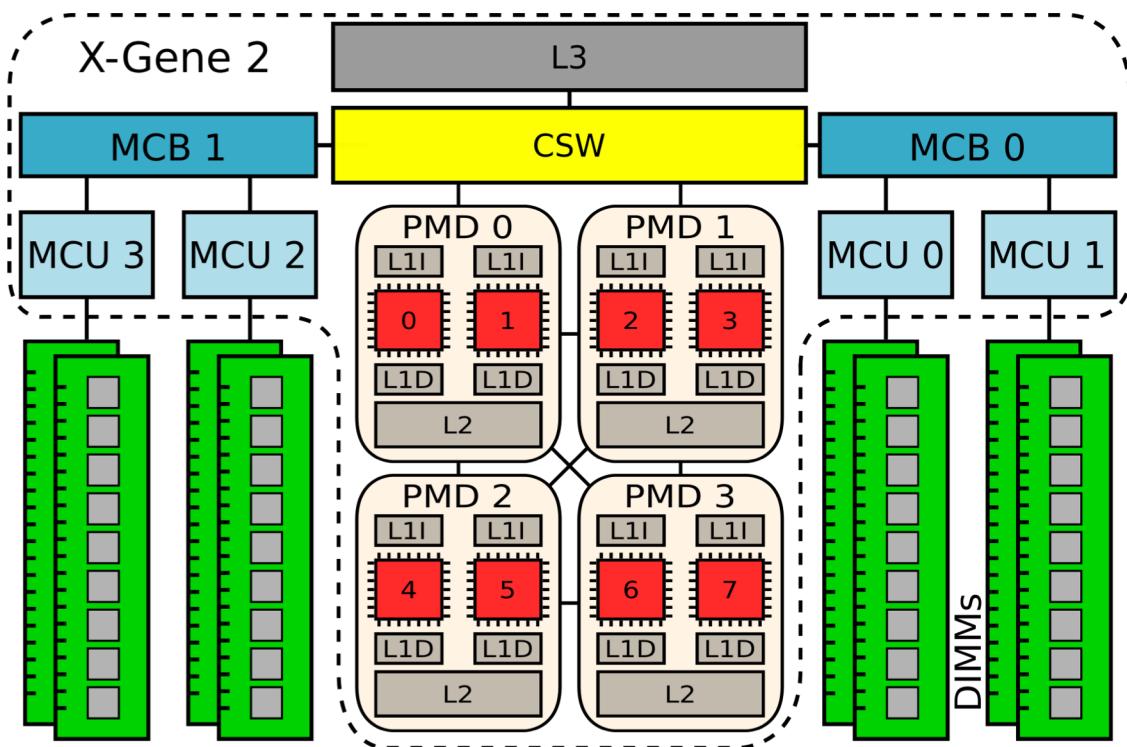
Methodology



Data collection



DRAM Characterization Framework - X-Gene 2

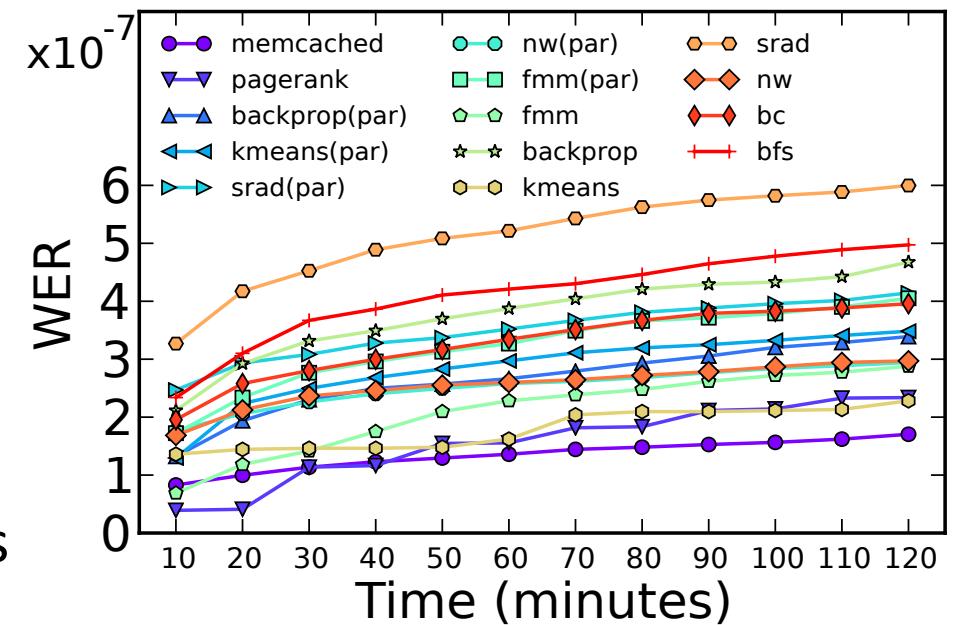


Parameter	Configuration
ISA	ARMv8 AArch64
CPU	8 cores
Core clock	2.4 GHz
L3 \$	8MB (SECDED)
Technology	28 nm
Max TDP	35 W
DRAM	4 x 8GB DDR3-1866
DRAM Chip Density	2Gb
Error Correction Code	SECDED ECC
Memory Characteristics	2 rank/MCU, 8 banks/rank, 64 K rows/bank, 64 B cache line

DRAM Characterization Framework - Workloads



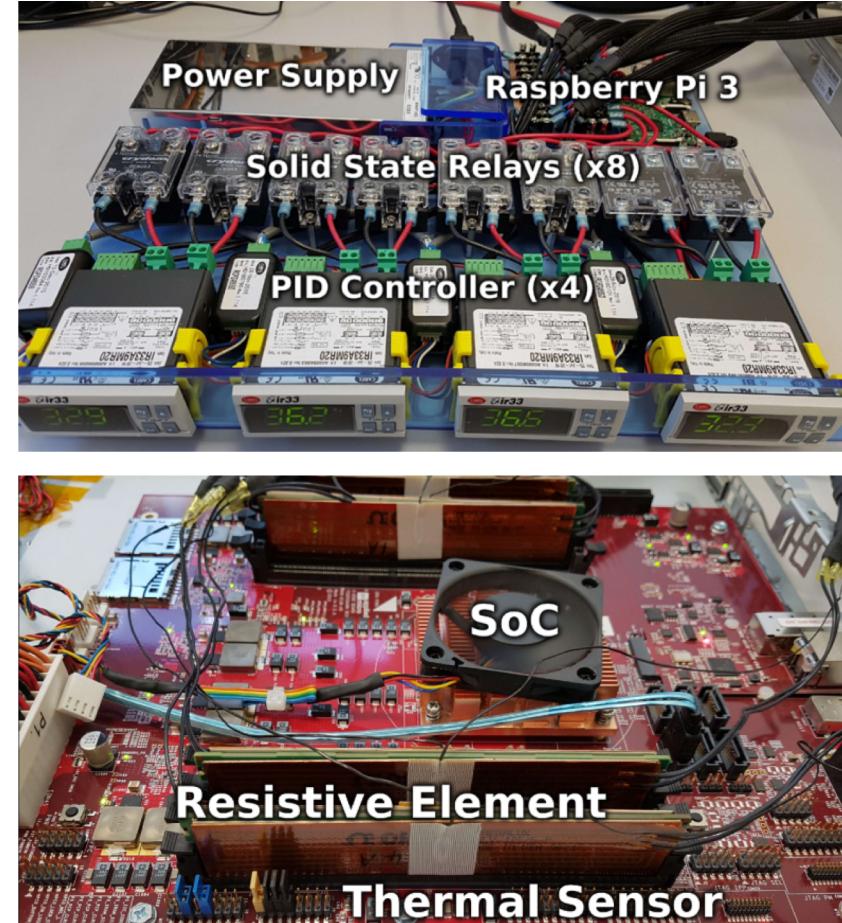
- We use various memory intensive computing, caching and data analytics workloads
- Each workload is executed in a loop for 2 hours
 - Mitigate VRT effects
 - We discover less than 3% of new error locations per 10 minutes of extra experimentation time
- Parallelism:
 - Single- and multi-threaded (1 / 8)
- Compiler optimization flags:
 - $-O2$ / $-F$



Suite	Benchmarks	Configuration
Rodinia/Parsec	Kmeans, nw, backprop, srad, fmm	1-thread/8-threads
Ligra	pagerank, bfs, bc	8-threads
CloudSuite	memcached	8-threads
Iulesh	Iulesh	$-O2$ / $-F$

DRAM Characterization Framework - Parameters

- Refresh Rate
 - Relax refresh period by 35x from 64ms up-to 2.283s, or refresh rate interval from 7.8us up-to 278.3us
- Voltage
 - Lower the supply voltage by 5%, from 1.5 V to 1.428 V
 - Very small variation of V_{min} across boards
- Temperature of DRAMs
 - Unique thermal testbed for controlling the temperature of each DIMM independently
 - Experiments across different controlled temperatures



DRAM Error Metrics



- DRAM type of errors that SECDED ECC can detect or correct:

# of bit-errors	Type of errors	Abbreviation
1	corrected	CE
2	uncorrected/detected	UE
>2	uncorrected/undetected	SDC

- We check for SDC by comparing the output with the golden output while executing the application under nominal parameters

- To characterize DRAM error behavior, we use two metrics:

$$WER = \frac{N_{CE}}{MEM_{SIZE}}$$

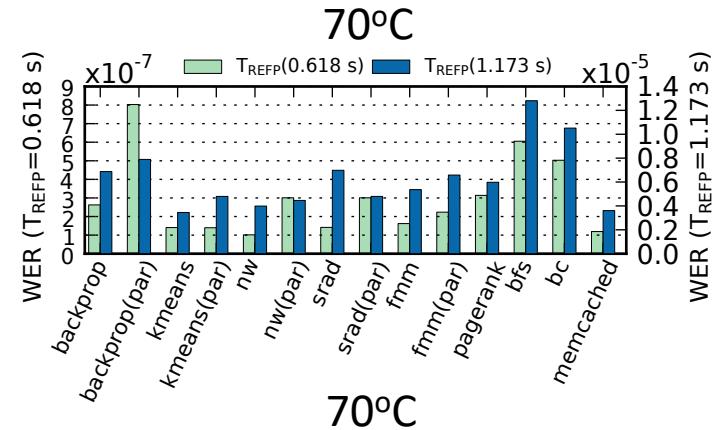
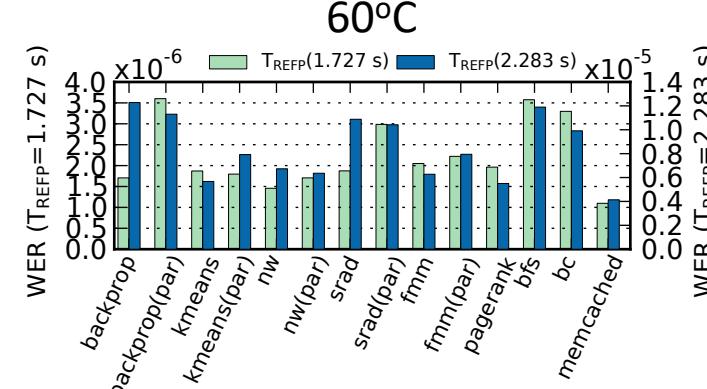
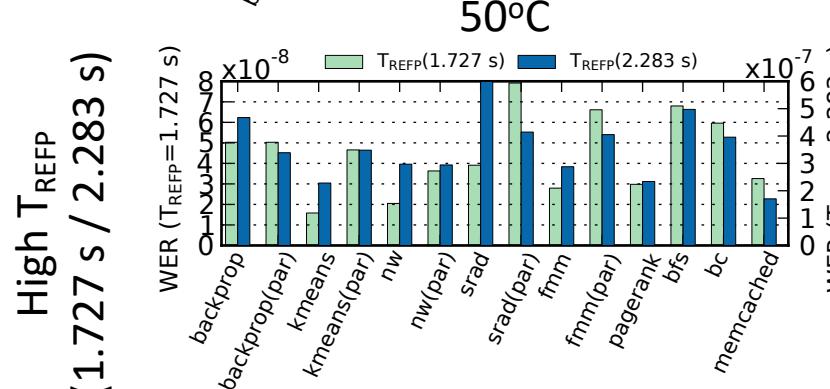
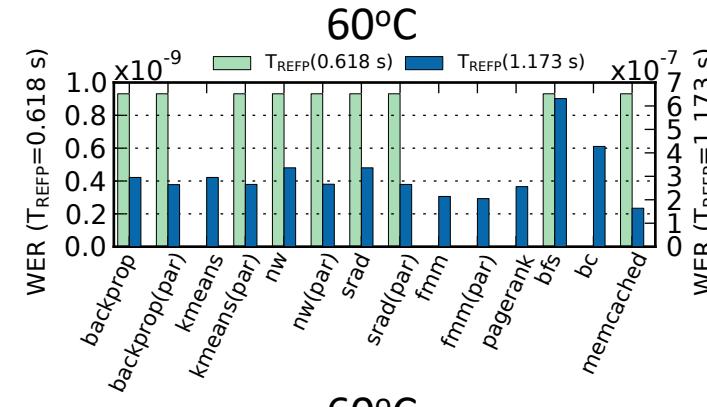
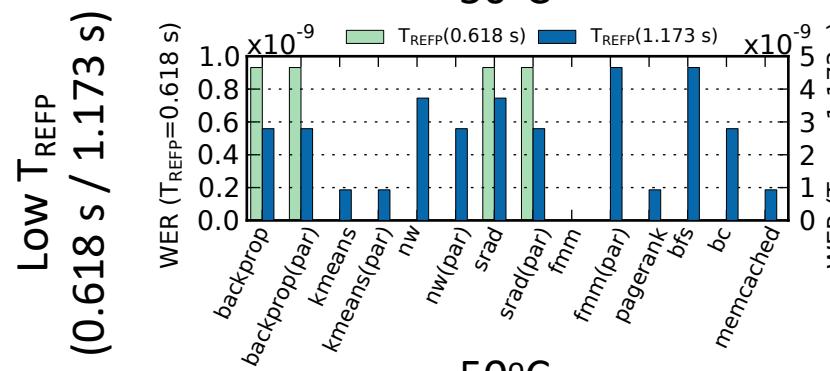
- WER shows the probability of a row to manifest CE

$$P_{UE} = \frac{N_{UE}}{N_{EXP}}$$

- P_{UE} shows the probability of an execution to manifest UE and thus crash



DRAM Characterization Campaign - WER per Application



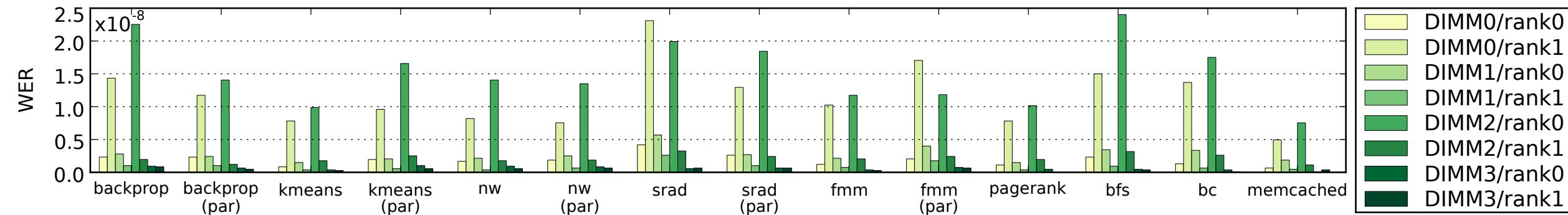
The system crashes much earlier than 2 hours and WER does not reach the maximum value

- Voltage scaling has a negligible effect on DRAM reliability
- WER may vary by 8x between realistic workloads
- The number of threads affect DRAM reliability

DRAM Characterization Campaign - WER per Device

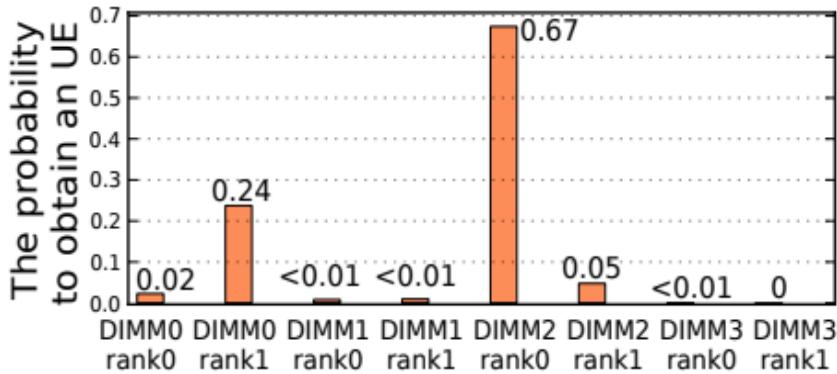
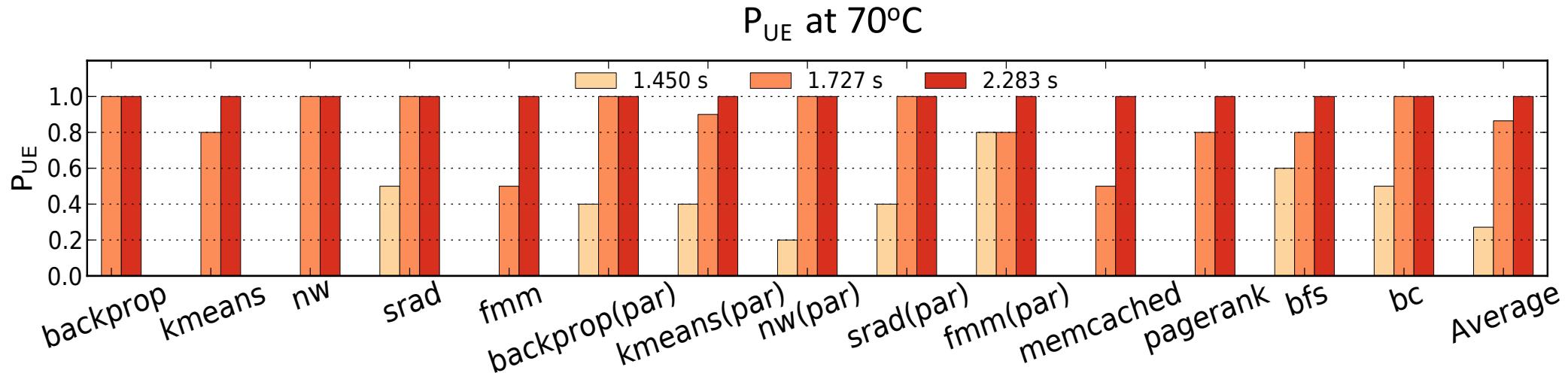


DIMM-to-DIMM variation at 50°C



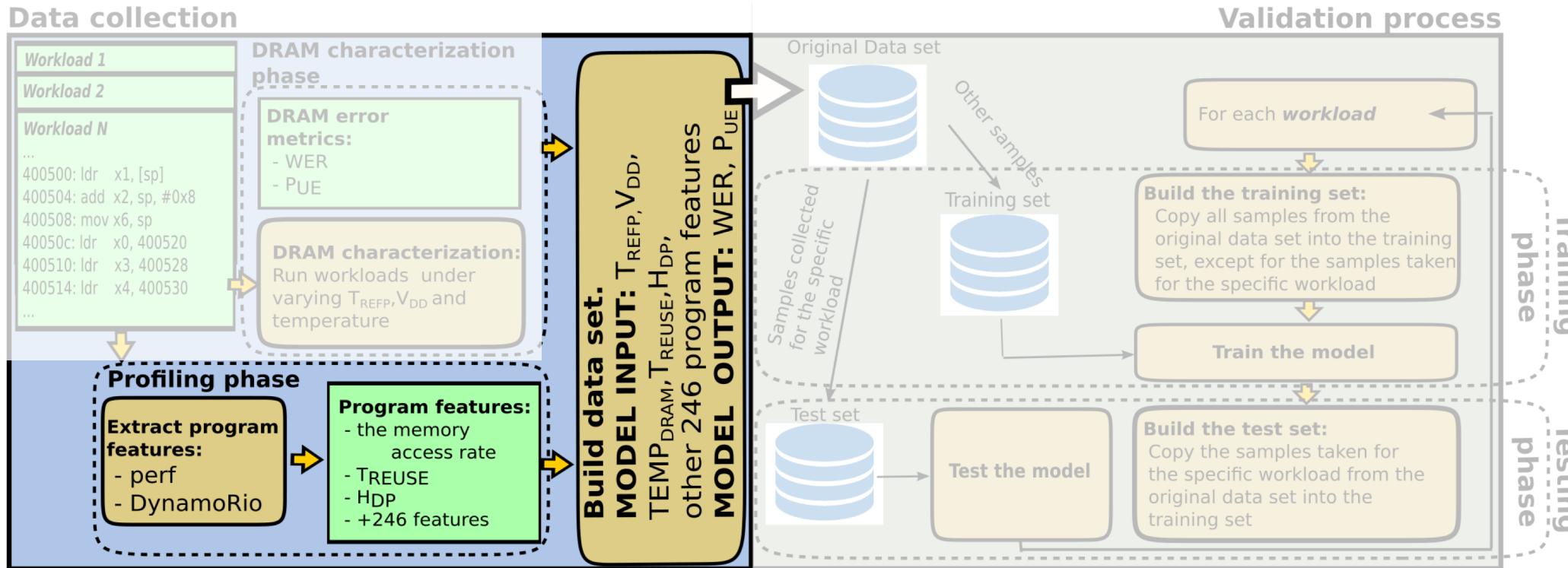
- WER may vary across DIMMs by up-to 188x
- WER across all workloads has the same tendency, when observing from which device errors are manifesting
- DRAM reliability is highly depended on the DRAM devices
 - Need to train the model for each DIMM

DRAM Characterization Campaign - P_{UE} per Application and Device



- Uncorrectable errors are obtained only at 70°C
- P_{UE} varies significantly across DIMMs
(DIMM2/rank0 $P_{UE}=0.67$, DIMM2/rank0 $P_{UE}=0$)

Methodology



Program Feature Extraction



- 247 program features measured via the performance hardware counters, (e.g. memory/cache read/write access rate, wait cycles, IPC, CPU utilization, branch instructions, branch mispredictions, page faults, ...)

- Memory access patterns
 - Measure the average memory reuse time

$$T_{REUSE}^i = CPI * D_{REUSE}^i$$

$$T_{REUSE} = \frac{\sum T_{REUSE}^i}{N}$$

- Memory data pattern
 - Measure the data entropy

$$H_{DP} = - \sum_{i=0}^{2^{32}-1} P(x_i) * \log_2(P(x_i)); \quad P(x_i) = \frac{N_{WR}(x_i)}{N_{TOT}}$$

- Measurement of performance counters with Linux perf

```
21.812281 task-clock          # 0.912 CPUs utilized
      15 context-switches       # 0.001 M/sec
        2 CPU-migrations        # 0.000 M/sec
      2,805 page-faults         # 0.129 M/sec
    62,025,623 cycles           # 2.844 GHz
    6,299,287 stalled-cycles-frontend # 10.16% frontend cycles idle
  24,456,020 stalled-cycles-backend # 39.43% backend cycles idle
  12,655,619 instructions       # 0.20 insns per cycle
                                # 1.93 stalled cycles per insn
  3,552,630 branches          # 162.873 M/sec
    51,348 branch-misses        # 1.45% of all branches

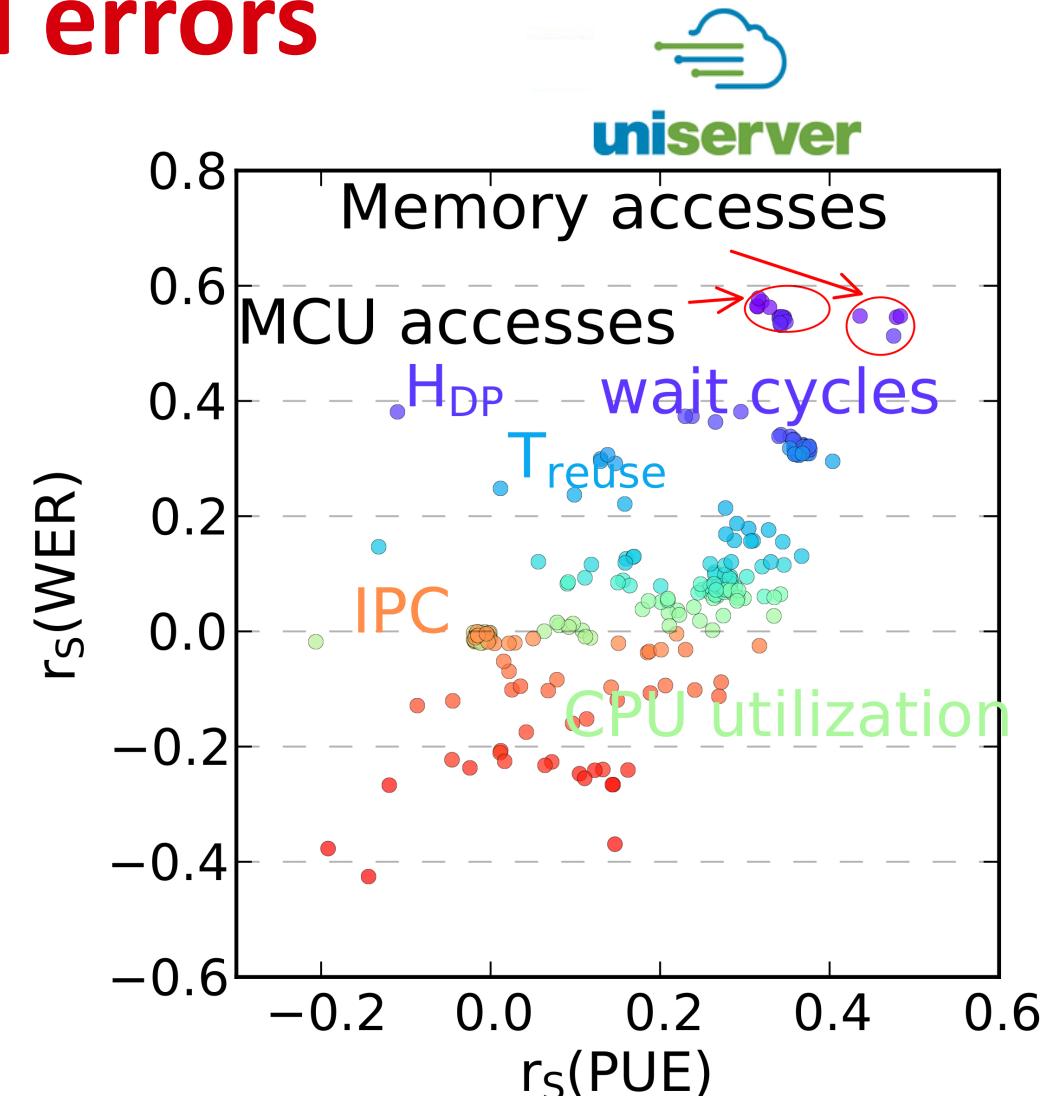
0.023914596 seconds time elapsed
```

- Measurement of memory reuse time and data entropy with DynamoRIO

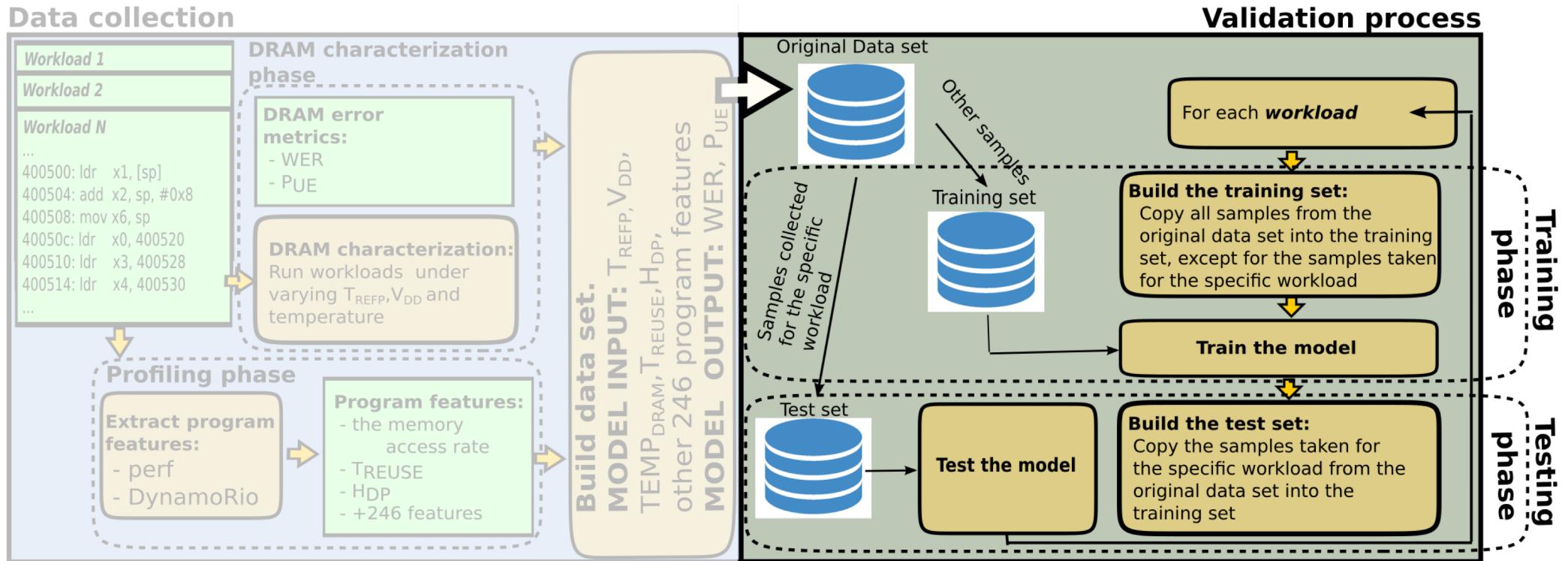


Correlation Analysis of DRAM errors with Program Features

- We use the Spearman's coefficient to correlate WER and P_{UE} with program inherent features
- By applying this analysis, we identify the most critical features that affect DRAM error behaviour
- The highly correlated features are: the memory access rate, the MCU access rate, wait cycles
- Features that are highly correlated to WER are also correlated to P_{UE}



Methodology



Machine Learning Model



ML models: Support Vector Machines (SVM), K-Nearest Neighbors (KNN) and Random Decision Forests (RDF)

Training Phase

- We use three sets of program inherent features to train the model
- The accuracy will depend significantly on the feature set used for training

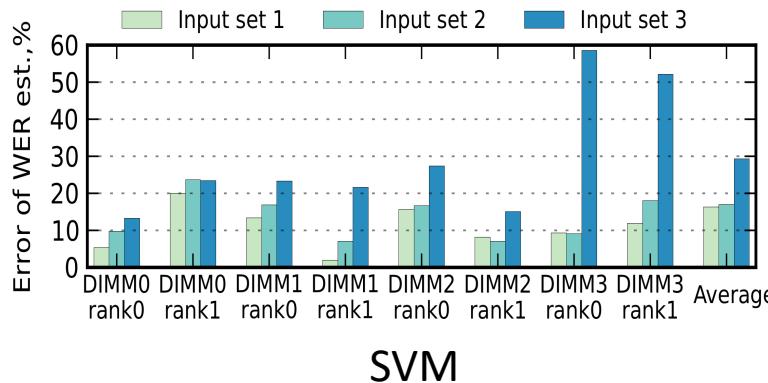
Set	Features
Set 1	DRAM temperature, refresh rate, wait cycles, memory access rate, data entropy, reuse time
Set 2	DRAM temperature, refresh rate, wait cycles, memory access rate
Set 3	DRAM temperature, refresh rate, all inherent program features

Testing Phase

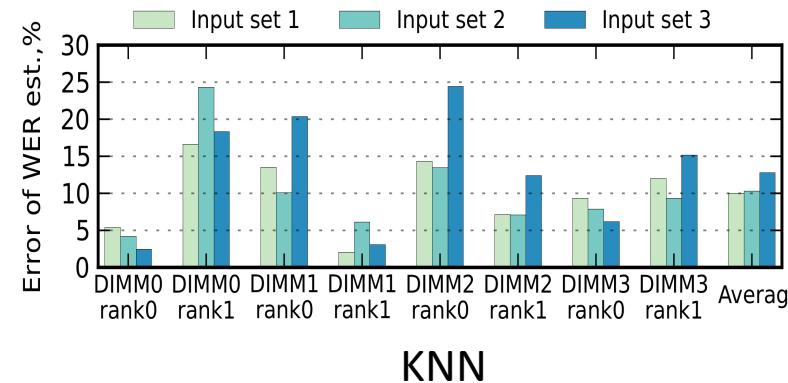
- We use the Leave-One-Out partitioning for evaluating the accuracy of the trained ML model



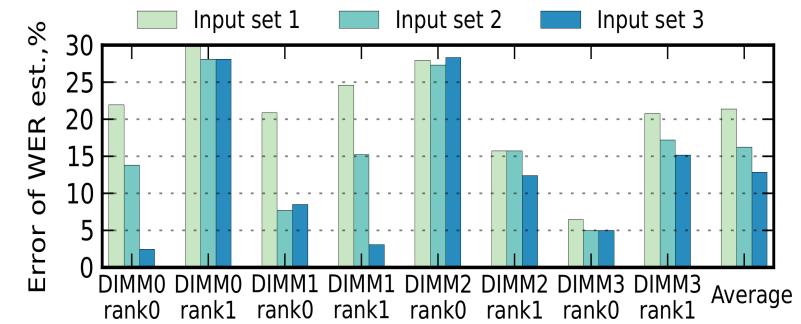
Evaluation - Modeling Accuracy per Device



SVM



KNN



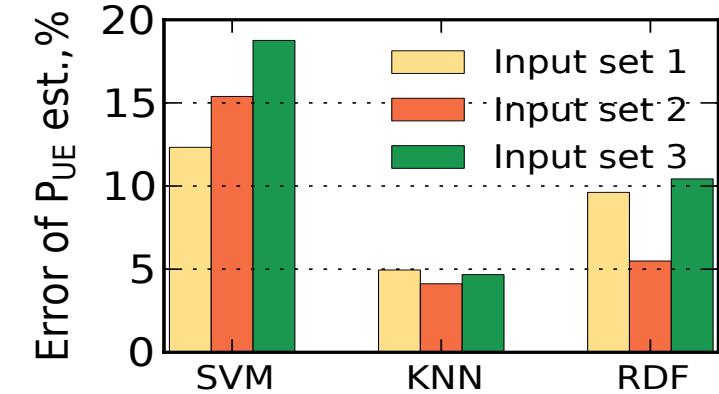
RDF

WER Estimation:

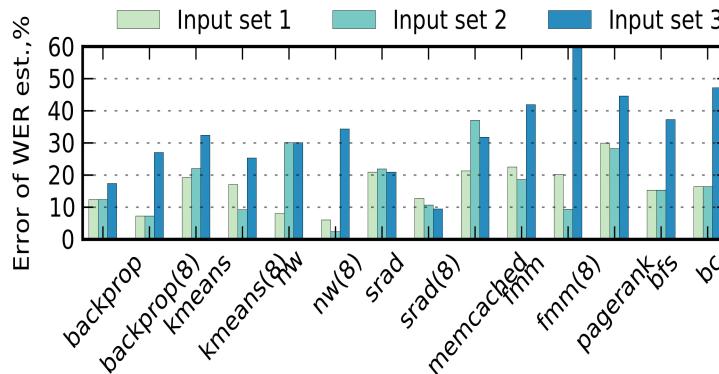
- The highest accuracy is observed for KNN and the input set 1
- The input set 1 for both SVM and KNN has the best accuracy
- The lowest accuracy is observed for the input set 3 but it has the best accuracy for RDF

P_{UE} Estimation:

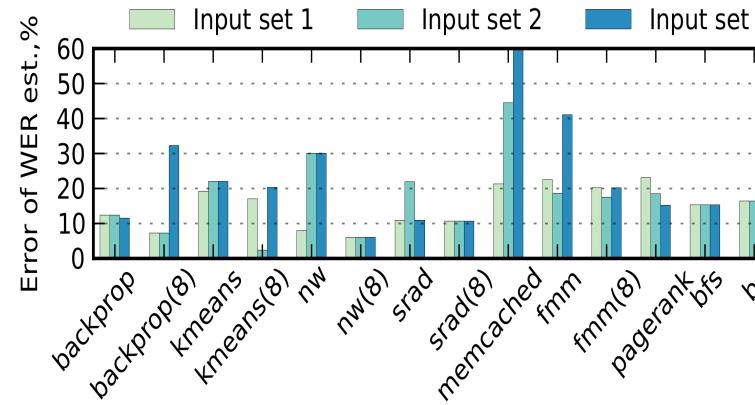
- The highest accuracy for P_{UE} estimation is observed for KNN



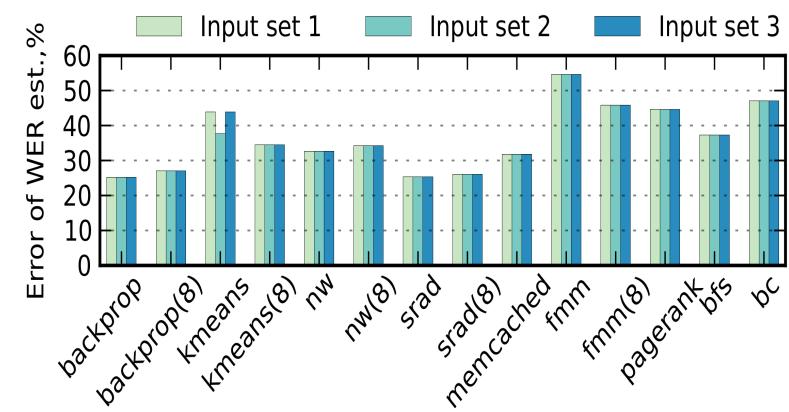
Evaluation - Modeling Accuracy per Application



SVM



KNN



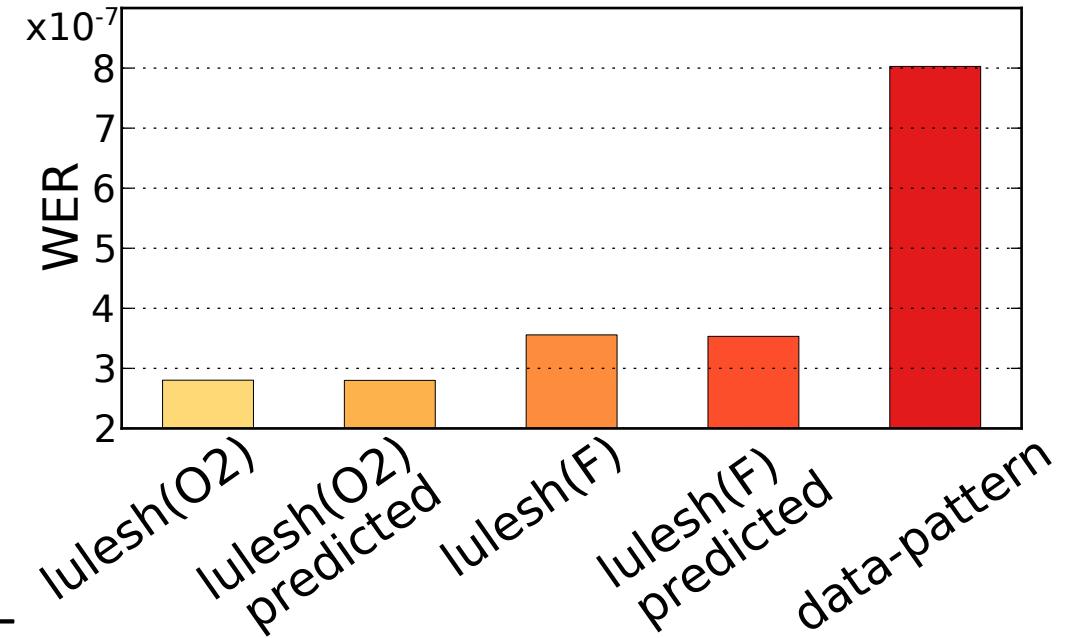
RDF

- The highest accuracy is observed for KNN and the input set 1
- The input set 1 for both SVM and KNN has the best accuracy
- The lowest accuracy is observed for RDF

Advantages of Workload-aware Modeling



- DRAM error model is crucial for understanding the effect on DRAM reliability of:
 - workload characteristics
 - software optimizations
- The model can predict WER and P_{UE} in milliseconds based on the characteristics, while characterization campaigns may take months
- The model enables us to predict the worst/best-case error behavior for specific DRAM chips
 - Find DRAM guardbands (Future research)



Conclusions



- The number of errors may vary across workloads by up-to 8x, as the program inherent features affect the DRAM reliability.
- DRAM error behavior of each DRAM device can be modeled based on the most significant program features, temperature and circuit parameters.
- We use Machine Learning to automatically train such a model to predict the expected number of errors and the probability of a crash.
- The KNN-based model provides the best accuracy with an average error of 10.5%.
- The model enables us to accurately predict the effect of a particular workload on DRAM reliability without long-running characterization campaigns.

Acknowledgments



UniServer Project

Led by Queen's University Belfast

(Grant Agreement 688540, <http://www.uniserver2020.eu>)

OpreComp Project

(Grant Agreement 732631, <http://oprecomp.eu>)



Thank You

Have you any questions?