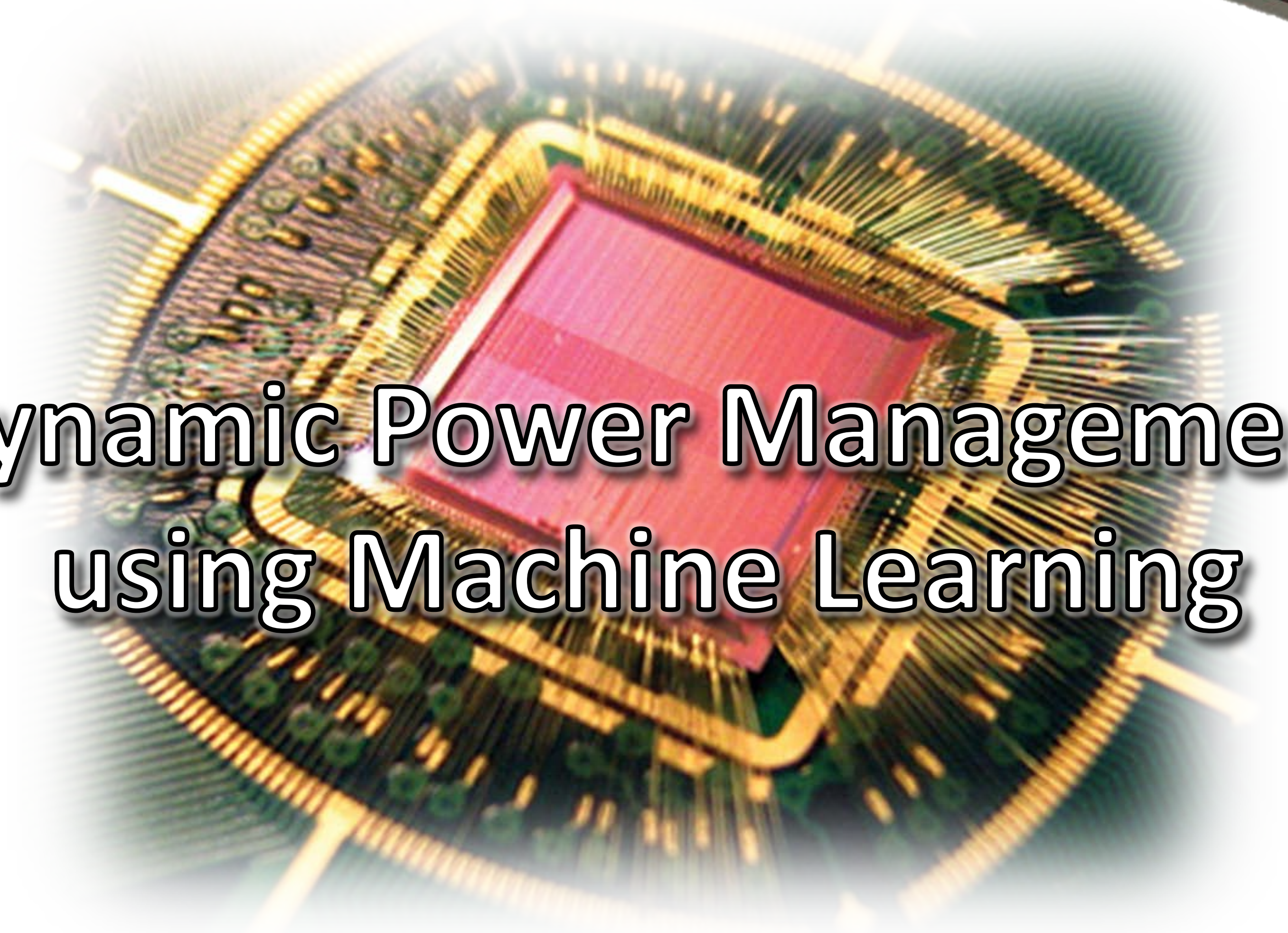


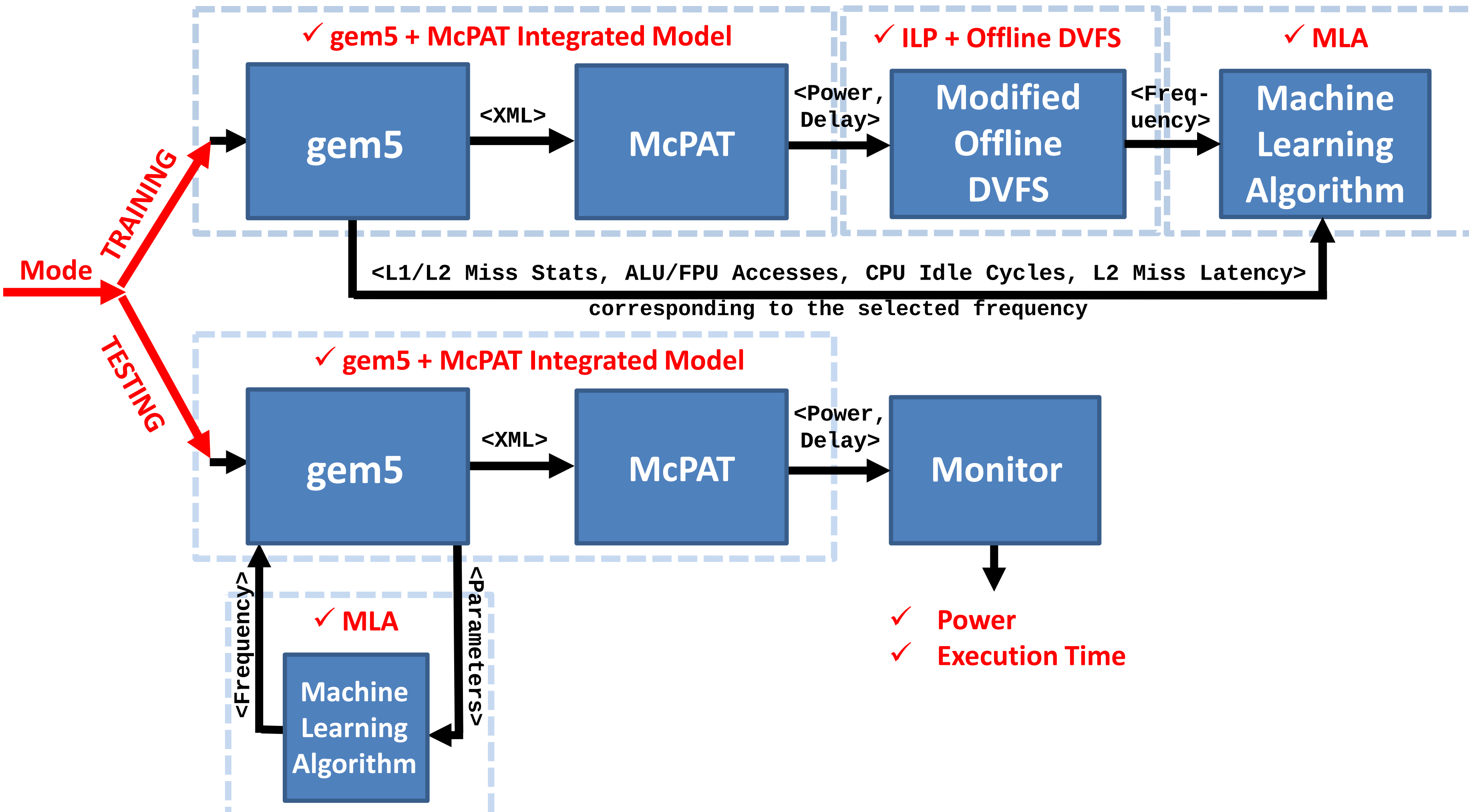
Dynamic Power Management using Machine Learning

Abhishek Pandey | Aman Chadha | Aditya Prakash

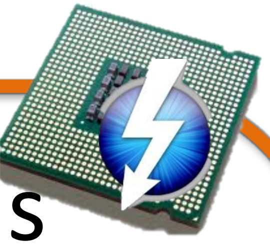
System: Building Blocks

- ✓ **Motivation:**
  - ✓ **Problem:** Determining when to scale down the frequency at runtime is an intricate task.
  - ✓ **Proposed Solution:** Use Machine learning algorithm to predict intervals where memory intensive tasks occur.
- ✓ **Phase 1: gem5 + McPAT**
- ✓ **Phase 2: ILP + Offline DVFS**
- ✓ **Phase 3: Machine Learning**

System: Overview

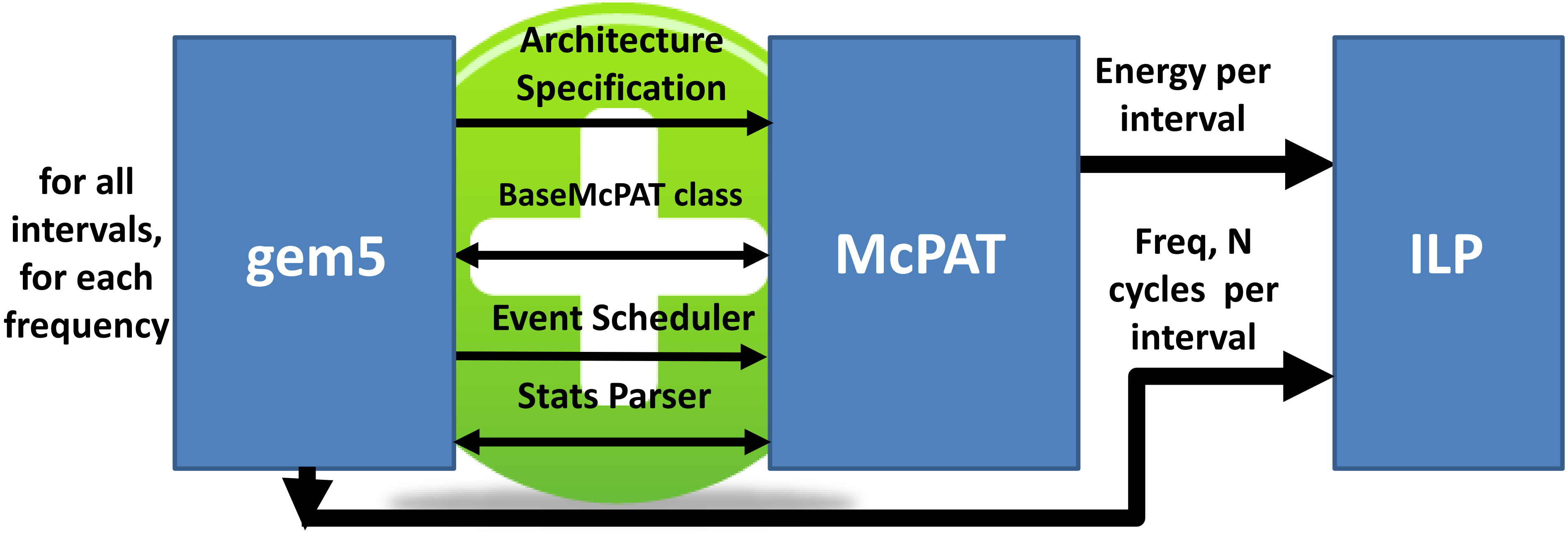


Testbed System Specifications

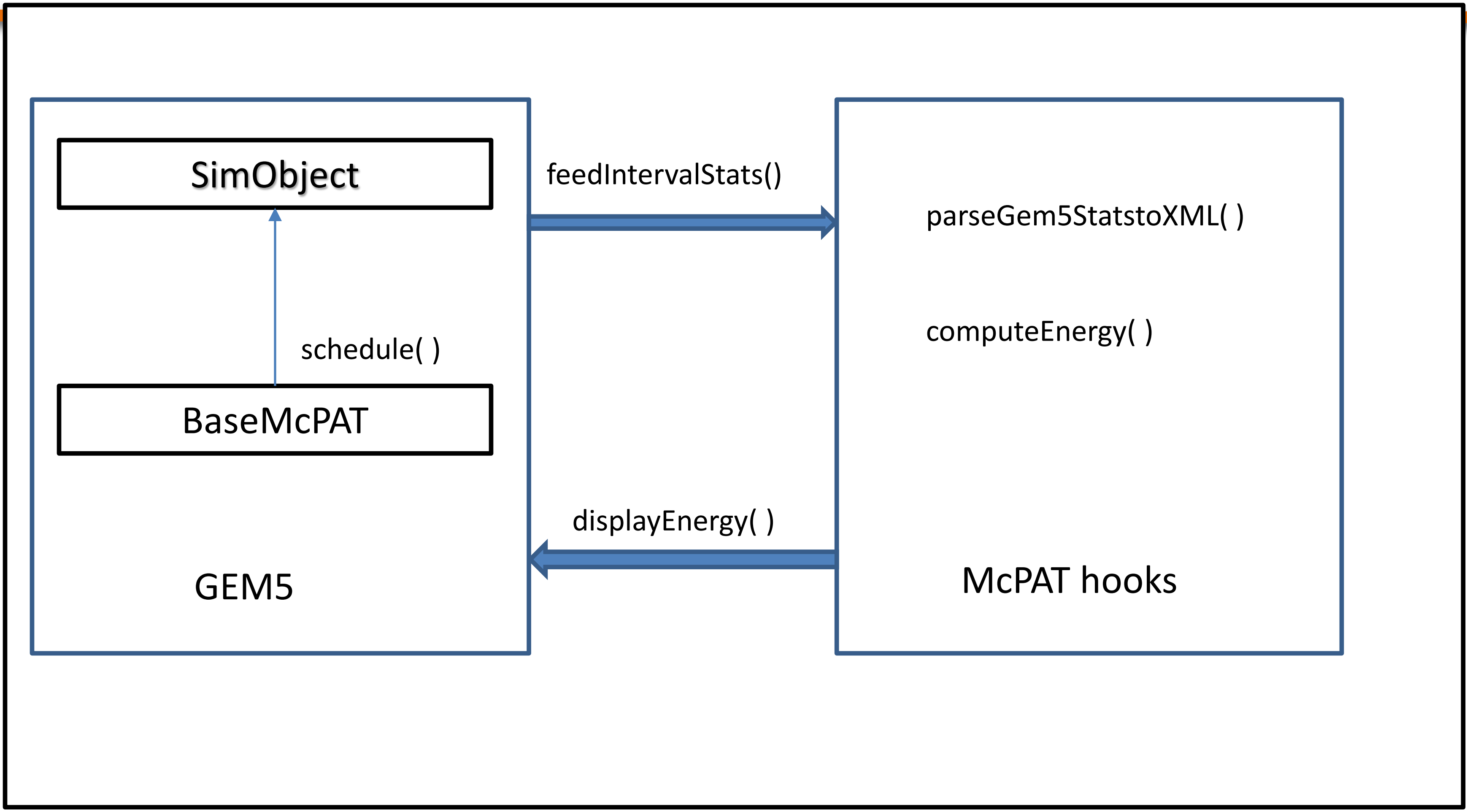


Parameter	Specification
ISA	ALPHA
Execution Mode	Out of Order Execution CPU
Base Frequency	2 GHz
L1 I-Cache and D-Cache Size	32 kB
L2 Cache Size	256 kB
Number of cores	1
Other parameters at their default values	

Phase 1: gem5 + McPAT
integration



Phase 1: gem5 + McPAT



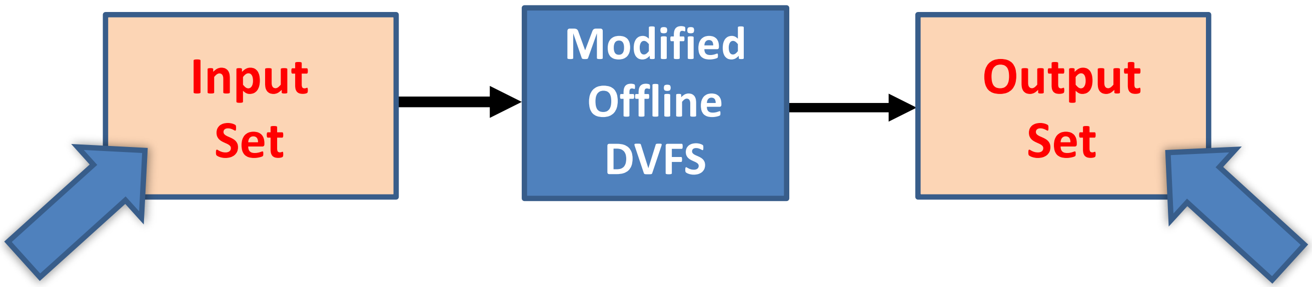
Phase 1: gem5 + McPAT

differentiating parameters chosen

- ✓ We chose the following parameters to ***profile memory characteristics*** of a workload
- ✓ Basically, we aimed at identifying the ***memory-boundedness*** of the workload in a particular interval of execution

Parameter	
L2 Miss Stats	Functional Unit Busy Rate
# of FPU Accesses	Miss Latency for L2
# of ALU Accesses	CPU Idle Cycles

Phase 2: ILP + Offline DVFS



- **Power value for all intervals and for all frequencies**
- **Delay value for all intervals for all frequencies**
- **Performance constraint = δ**
- **Execution time = 1/(Performance)**

- $\mathbf{P_{ij}}$ where ***i*** is the interval number and ***j*** is the frequency number
- Hence,
  - $\mathbf{P_{11}}$ means interval #1 and frequency number #1 (= 800 MHz)
  - $\mathbf{P_{12}}$ means interval #1 and frequency number #2 (= 1 GHz)

Considering 2 intervals I1 and I2:

I1: $\boxed{P_{11}}$, P_{12}, P_{13}, P_{14}
I2: P_{21}, P_{22}, $\boxed{P_{23}}$, P_{24}

I1: $\boxed{D_{11}}$, D_{12}, D_{13}, D_{14}
I2: D_{21}, D_{22}, $\boxed{D_{23}}$, D_{24}

✓ Select P_{11} for I1.
✓ **x_{11} = 1**. $\therefore$ $x_{12} = x_{13} = x_{14} = 0$.
✓ Select P_{23} for I2.
✓ **x_{23} = 1**. $\therefore$ $x_{21} = x_{22} = x_{24} = 0$.

. . . where x_{ij} is a binary variable.

Phase 2: ILP + Offline DVFS

- We adopt the ***offline DVFS algorithm*** proposed by Kim et. al. in "System Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators" (HPCA'08)
- The DVFS control problem is formulated as an ***Integer Linear Programming*** (ILP) optimization problem
- We seek to reduce the ***total power consumption*** of the processor within ***specific performance constraints*** (δ)
- The application runtime is divided into ***N intervals***
- A total of ***L = 2 frequency levels*** are considered
- For each runtime interval *i* and frequency *j*, the ***power consumption***, P_{ij}, is calculated. The ***delay*** for each interval and V/F level, D_{ij} is also calculated
- The equations below formulate the ILP, which is solved using ***Gurobi Optimizer***

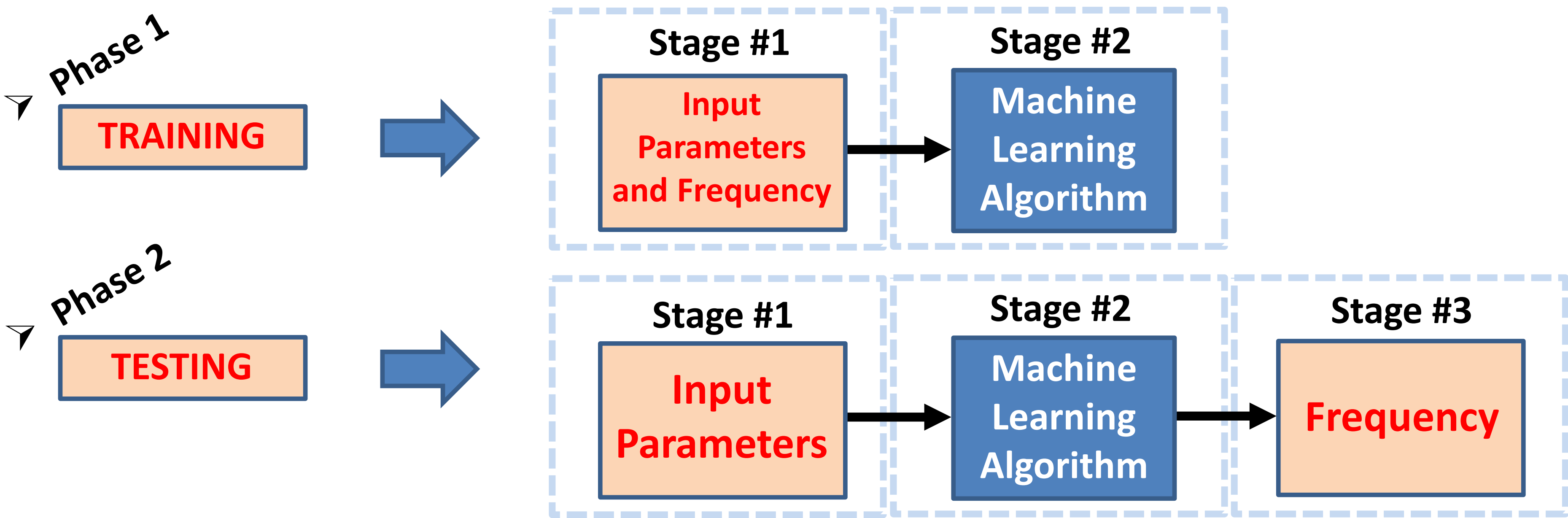
$$\min\left(\sum_{i=1}^{N}\sum_{j=1}^{L} P_{ij}x_{ij}\right)$$

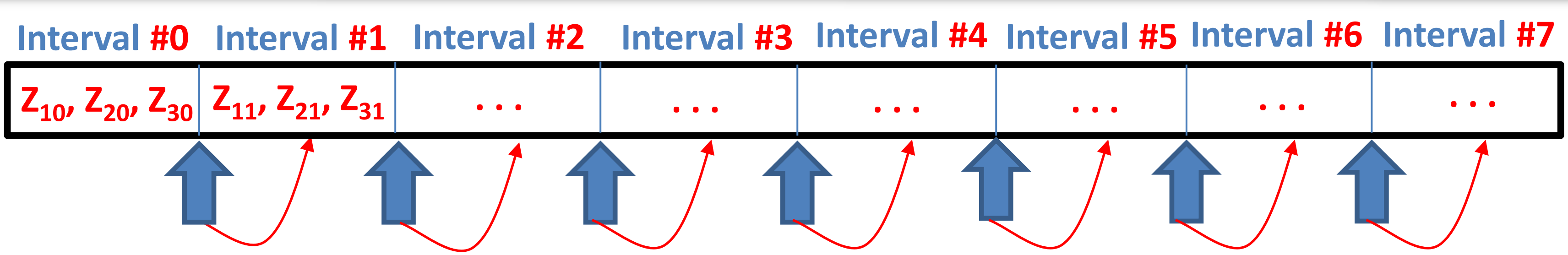
$$\left(\sum_{i=1}^{N}\sum_{j=1}^{L} D_{ij}x_{ij}\right) < \delta$$

$$\sum_{j=1}^{L} x_{ij} = N \quad \forall \;\; i = 1 \; to \; N$$

. . . where x_{ij} is a binary variable having values 1 or 0.

Phase 3: Machine Learning





Z_{ij} . . . where *i* is the parameter number (there are *n* such parameters) and *j* is the interval

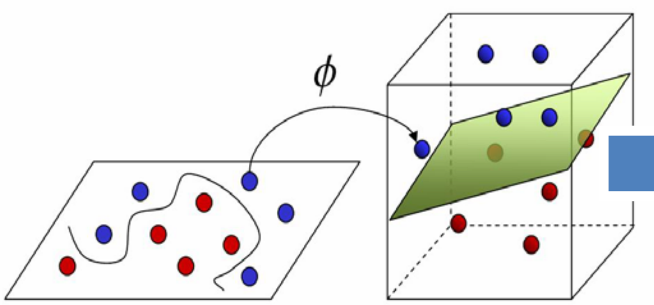
Phase 3: Machine Learning

learning and classification

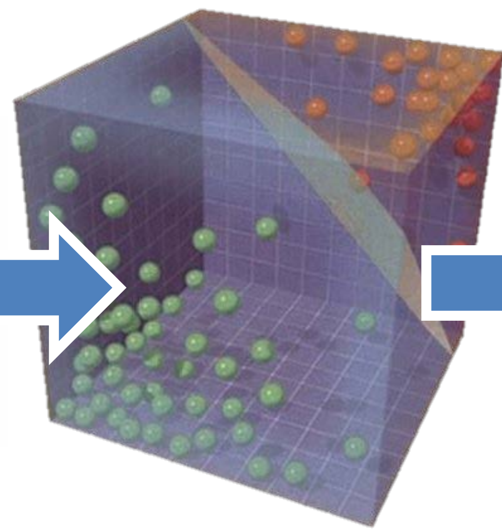
1 FEATURE EXTRACTION

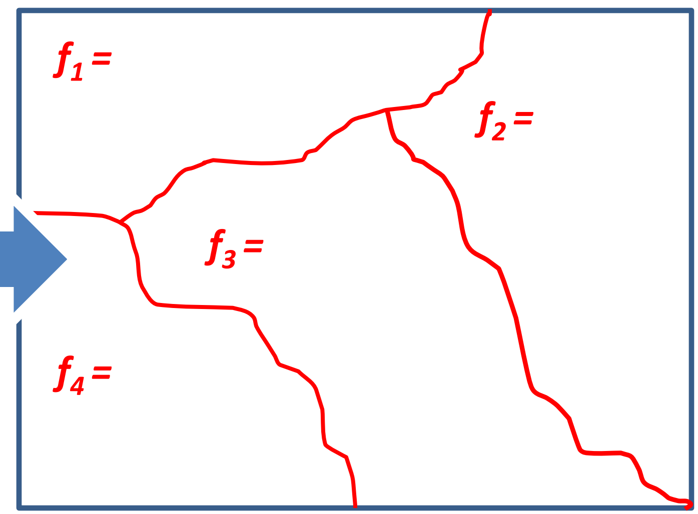
2 CLASSIFICATION

3 FREQUENCY OUTPUT



Input Space → Feature Space



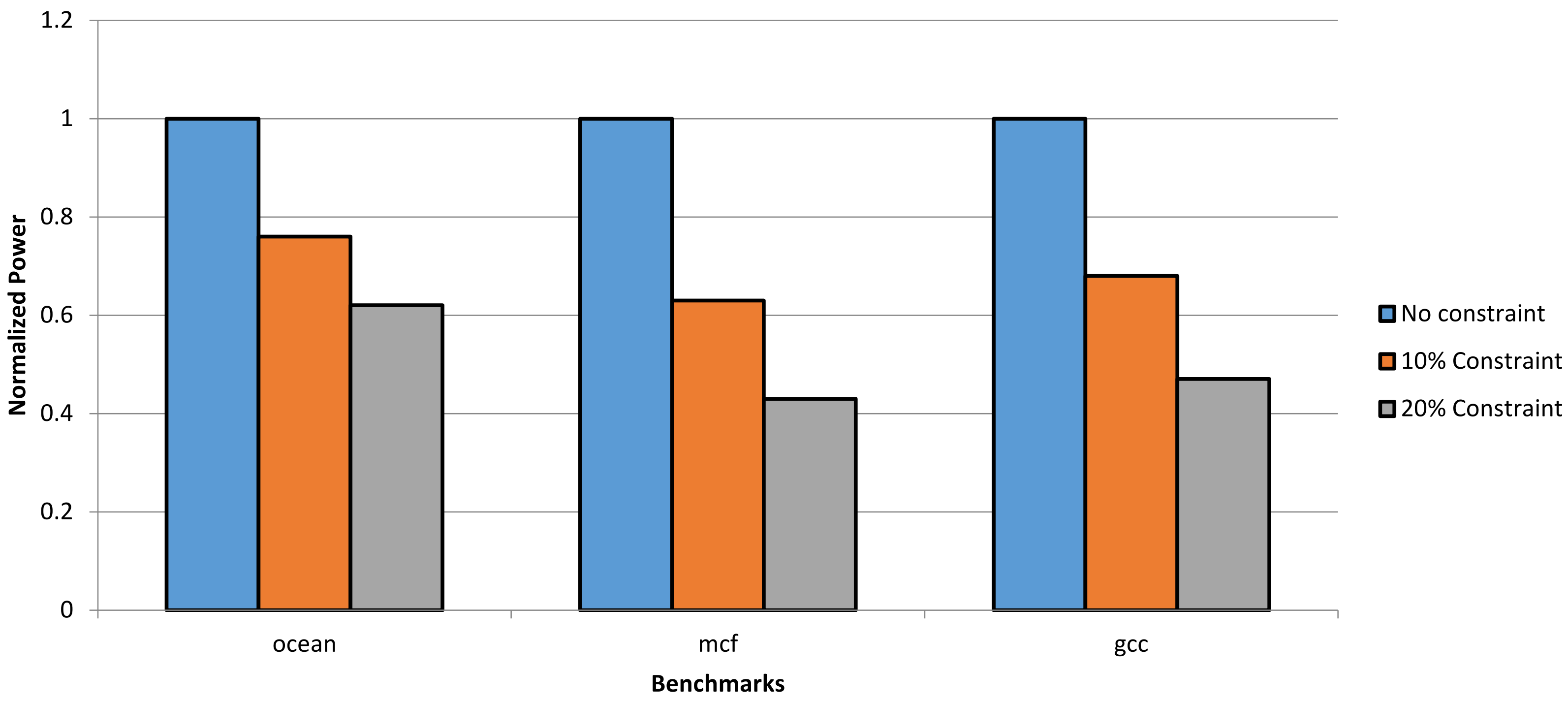


Contributions from our end. . .

- ✓ **ILP Formulation and Solving**
  - ✓ Python implementation for solving the ***ILP formulation*** using Gurobi Optimizer
- ✓ **Gem5 modifications**
  - ✓ ***McPAT*** integration
  - ✓ ***Dynamic Frequency Scaling*** in gem5
  - ✓ Integrating the ***SVM Model*** with gem5
- ✓ **Developed scripts**
  - ✓ Parsing the ***output of gem5*** in a format acceptable to the ***Machine Learning Algorithm***
  - ✓ C implementation to fetch execution statistics corresponding to the down-scaled frequency obtained after performing ***Dynamic Frequency Scaling***
  - ✓ In progress is a single script to ***automate the entire process***

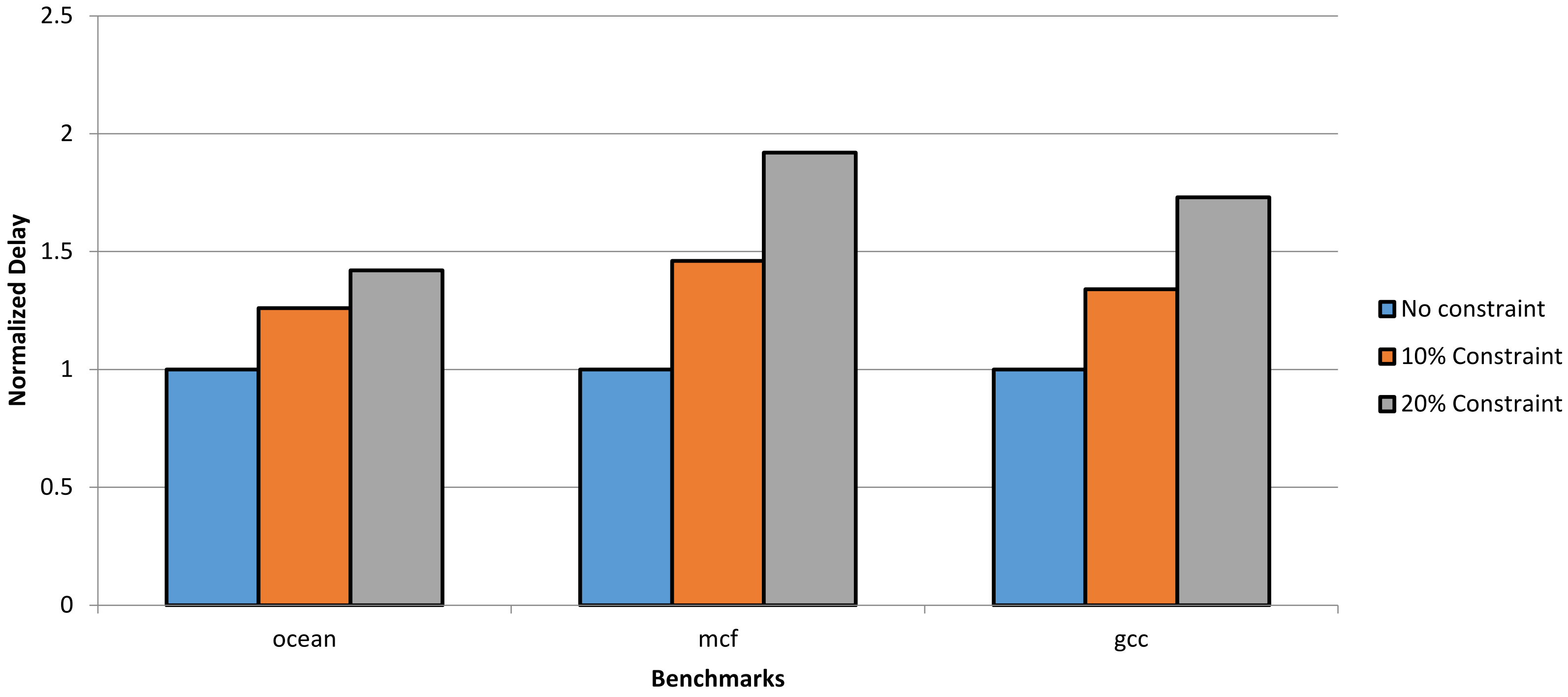
Experimental Results

power



Experimental Results

delay



Future Work

- ✓ Avenues for future optimization:
  - ✓ Finding the optimum ***interval size*** and ***number of intervals*** for training
- ✓ Other MLAs can be evaluated and a ***comparative analysis of all algorithms*** can be presented
- ✓ Several ***steps of frequencies*** can be introduced
- ✓ System performance can be evaluated with the ***permutation-combination*** of training and testing with some more benchmarks
- ✓ Additional ***memory/CPU parameters*** can be explored which can help to better train the MLA
- ✓ ***Memory profile of each benchmark*** can be analyzed to perform selective training for the MLA

Thank You . . .

. . . questions are welcome