A Fail-Slow Detection Framework for HBM Devices

Zikang Xu, Yiming Zhang and Zhirong Shen

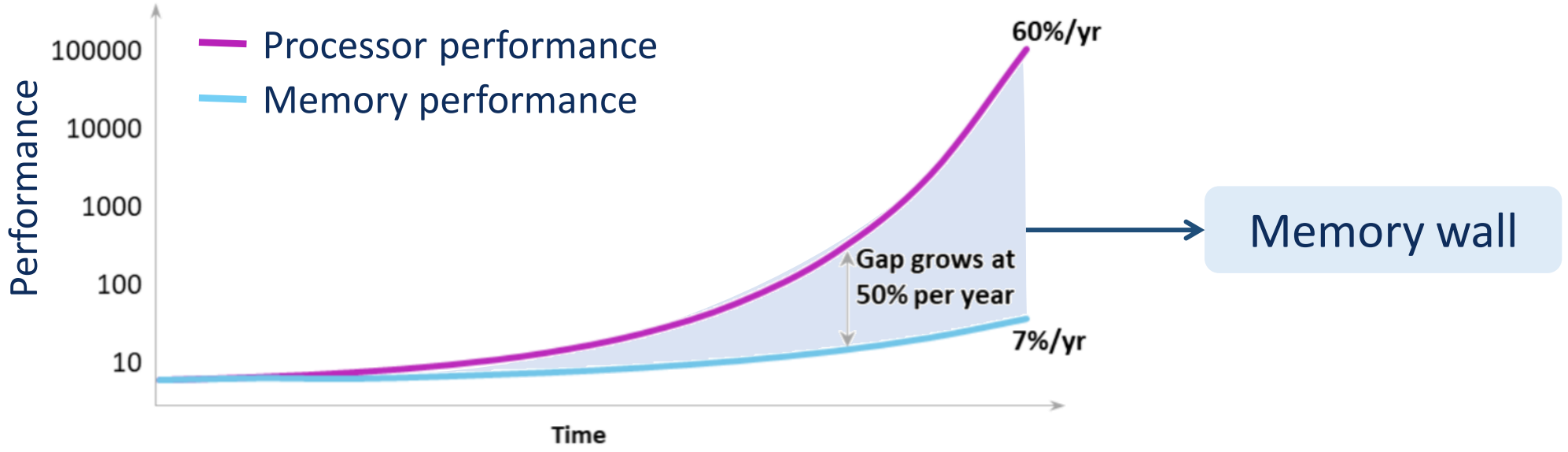
Outline

- Background
- Unsuccessful Attempts and Lessons
- A Fail-Slow Detection Framework for HBM Devices
- Conclusion

Background: Memory Wall

The gap between computing power and memory bandwidth is continuously widening in modern systems[1]

- Processors are improving exponentially, but memory bandwidth is increasing slowly



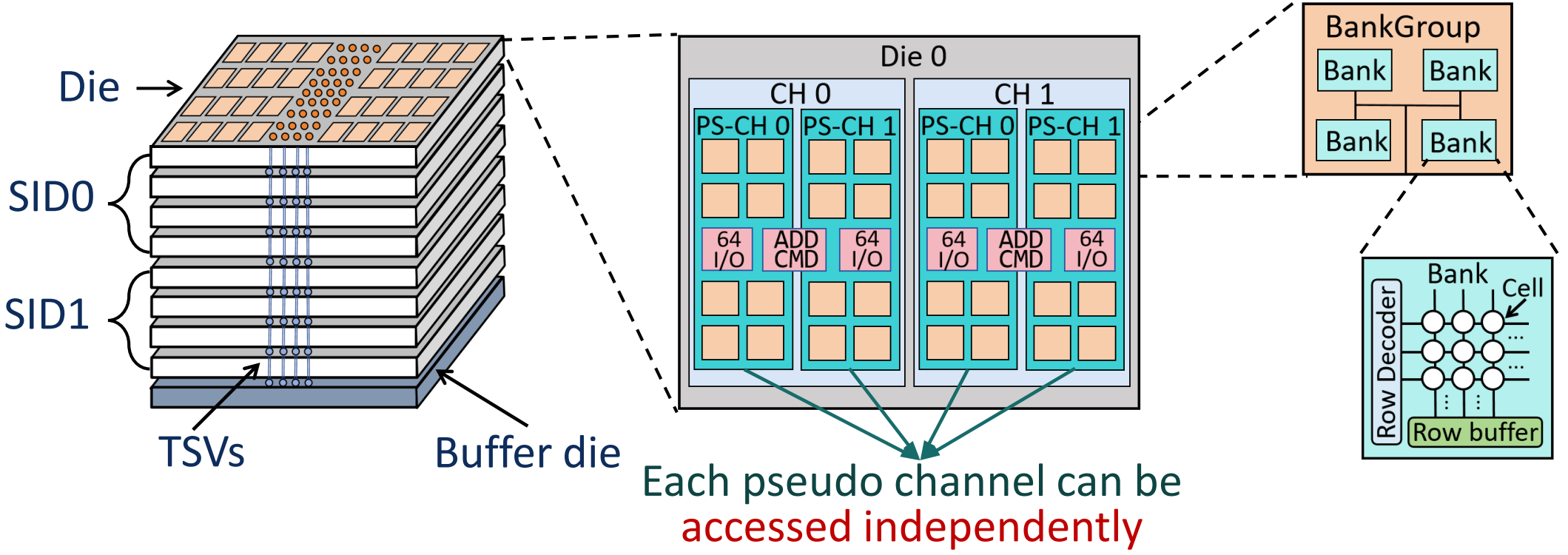
Memory wall becomes one of the major obstacles in training LLM models.

[1] Micron Inc., Micron's Perspective on Impact of CXL on DRAM Bit Growth Rate

Background: High-Bandwidth Memory

HBM is a hopeful technology to overcome the memory wall

- Save massive physical space by stack vertically
- Offer significantly higher data transfer rates
- Introduce reduced power consumption



Background: Fail-slow Faults

Recent Studies of Fail-slow Faults

- A survey of Fail-slow faults
- Researching and detecting fail-slow faults in HDDs and SSDs
- A fail-slow detection framework for cloud storage systems

Fail-slow faults in memory have been less studied. Existing studies basically focus on theoretical speculations but lack robust validation, replication, and detection tools.

Outline

- Background
- Unsuccessful Attempts and Lessons
- A Fail-Slow Detection Framework for HBM Devices
- Conclusion

Design Goals

A practical HBM fail-slow detection framework should have several properties.

- General. Due to the diversity of HBM devices, our framework aims to be applicable to all HBM devices with little or no modifications
- Non-intrusive. If possible, we do not wish to modify or affect the user code. We prefer to use existing workloads and external performance statistics for testing
- Accurate. This framework should be able to find some real fail-slow incidents and reveal some HBM fail-slow patterns

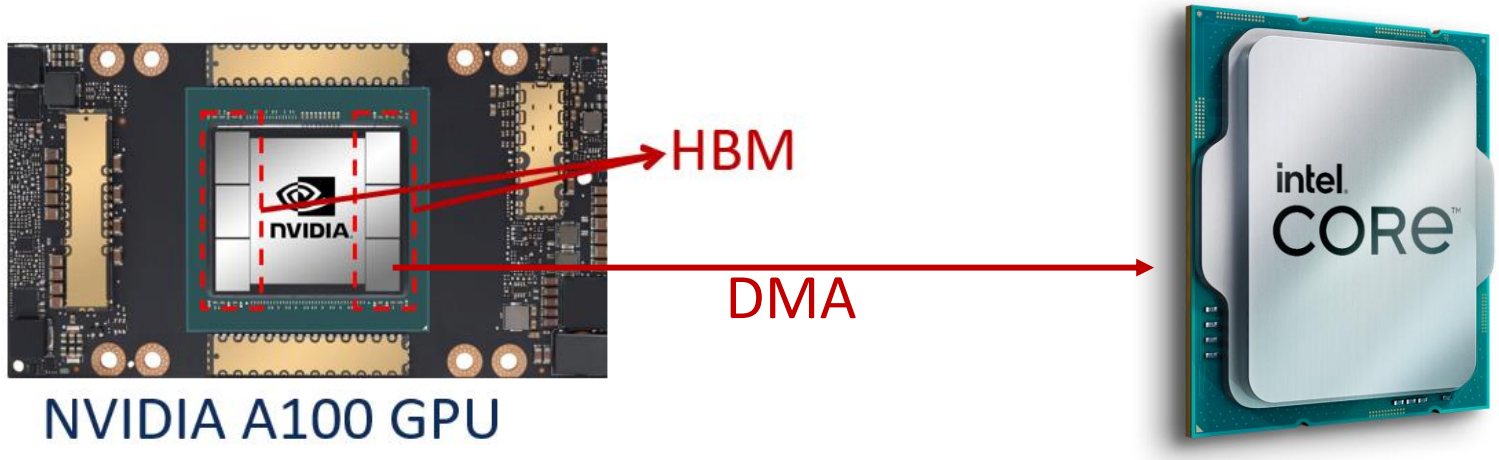
Attempt 1: Peer Evaluation

Existing memory testing tools

- Intel® Memory Latency Checker (MLC)
- A set of implementations based on the design of MLC

Limitation

- Most existing CPUs cannot directly communicate with HBM
- Memory testing is intrusive



Attempt 2: Fine-grained program tracking

Methodology

- Track and count the latency and bandwidth of each access step of the program in real time
- Existing mature tools for tracing programs at the granularity of kernel function

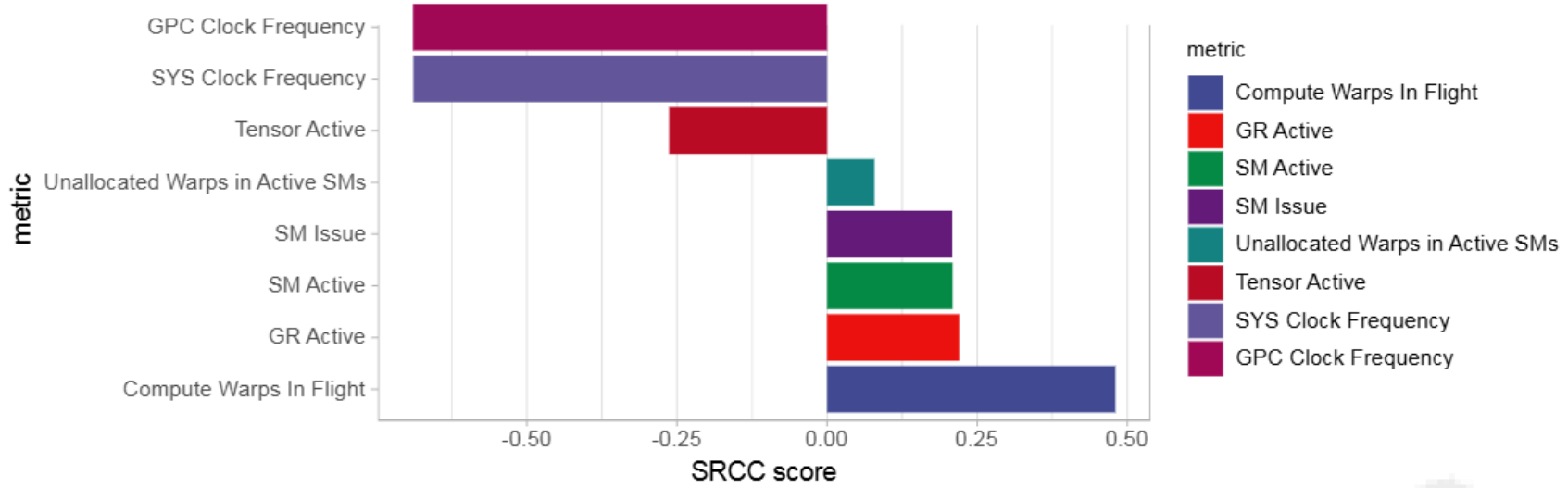
Limitation

- This type of tool can greatly impact performance and accuracy while introducing additional errors
- Too-fine granularity can overemphasize fluctuations and produce a lot of invalid data

Lessons

We want an offline, non-intrusive HBM fail-slow detection tool:

- construct polynomial regressions based on performance metric distributions at the node level to automatically derive adaptive thresholds
- define fail-slow events and a scoring mechanism
- metric distributions rather than single metric: <Throughput, Compute Warps in flight> pairs



Outline

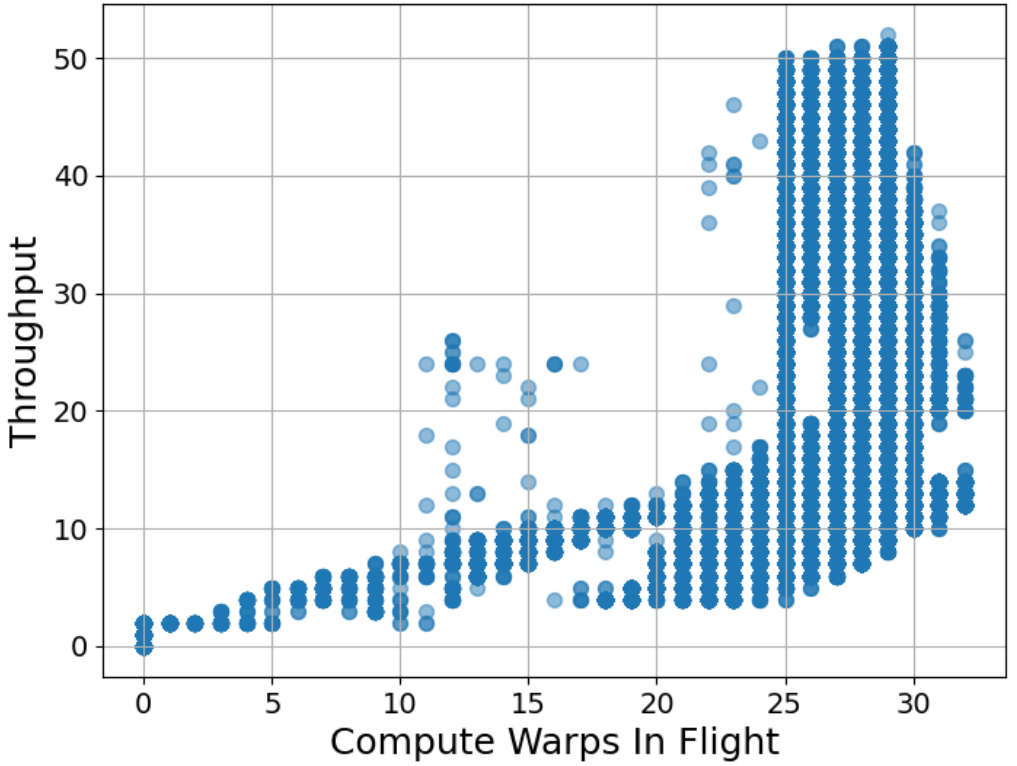
- Background
- Unsuccessful Attempts and Lessons
- A Fail-Slow Detection Framework for HBM Devices
- Conclusion

Dataset: A Single Node

A fixed load: a real-world neural network model

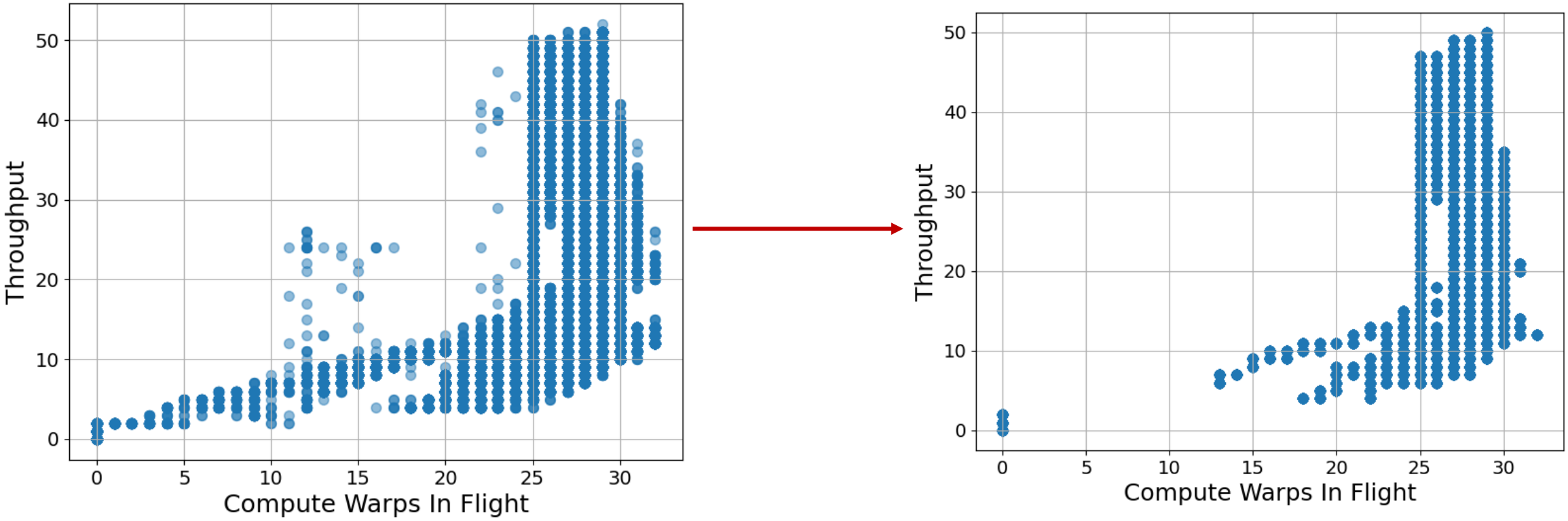
Items: <Throughput, Compute Warps in flight> pairs

- Using **Nvidia Nsight System** to collect GPU performance data at a frequency of 10Hz over a week. The dataset consists of roughly 600,000 entries (= 7 days × 24 hours × 60 minutes × 60 seconds × 10 entries/second).



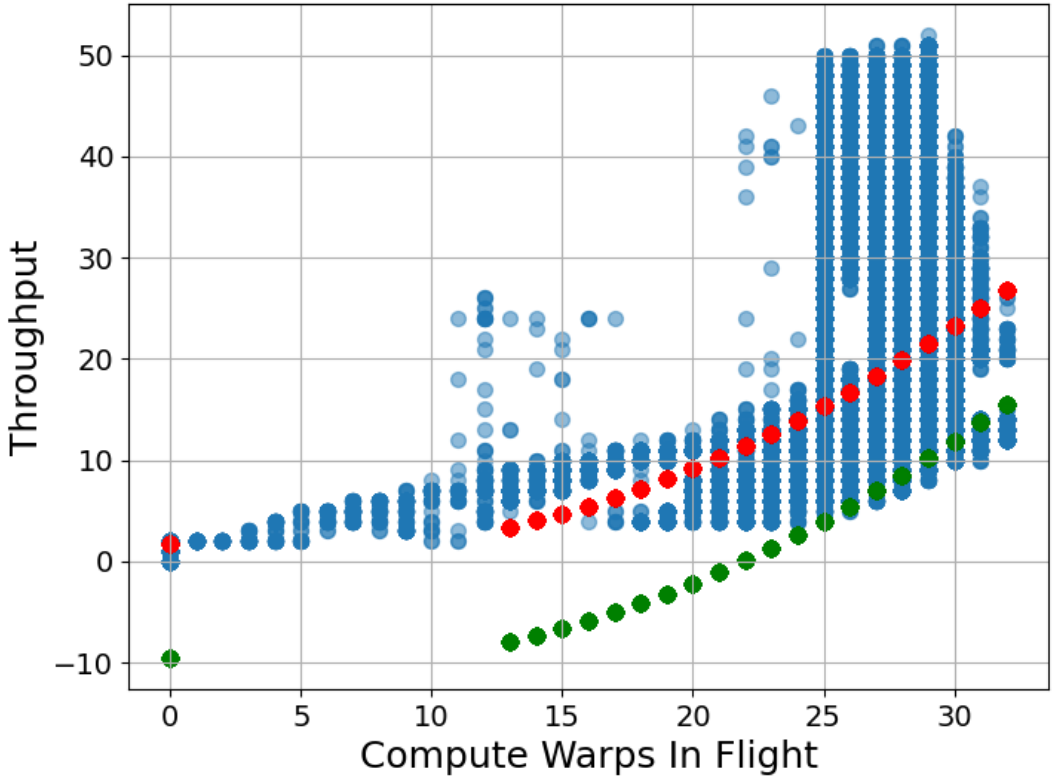
Outlier Detection

Filter out outliers:



Seek Fail-slow Faults

1.Building the Regression Model: Polynomial regression



2.Identifying Fail-slow Events

3.Risk Scoring Mechanism

Generally, We seek those time windows that are particularly slow for more than half the time and mark them as fail-slow events.

Finding: We find that slow items are highly concentrated, occurring in less than 2% of the windows. Meanwhile fail-slow events, that is, fail-slow windows, account for about 1% of the total time

Outline

- Background
- Unsuccessful Attempts and Lessons
- A Fail-Slow Detection Framework for HBM Devices
- Conclusion

Conlusion

Key Finding

- Fail-slow events exist in HBM devices
- ……

More details in the paper

Our Fail-slow Detection Framework

- Effectively identifies fail-slow events and slow items at the granularity of individual HBM device
- Under the premise of executing the same load, our method can effectively identify the slowest nodes. This testing scheme could be integrated into the factory testing process for products

Huawei Dataset

Over 15K DSA devices

- 60K+ HBM2 memory chips
- 460M+ errors
- 19 data centers: diverse services

Two years BMC log

- ErrLog_Cycle log: concise but fully recordable
- ErrLog_Occurrence log: detailed but difficult to fully retain
- Sensor log: temperature, power, etc.

Test dataset released

- https://github.com/wrl297/Calchas

Thanks & QA

A Fail-Slow Detection Framework for HBM Devices

NICEXLAB
Networked Intelligent Computing at EXascale

NICEXLAB: http://nicexlab.com/
Contact email: Zikang Xu, xuzikang@stu.xmu.edu.cn