CompPow: A Case for Component-level GPU Power Management

Shaizeen Aga and Mohamed Assem Ibrahim
June 26th, 2026

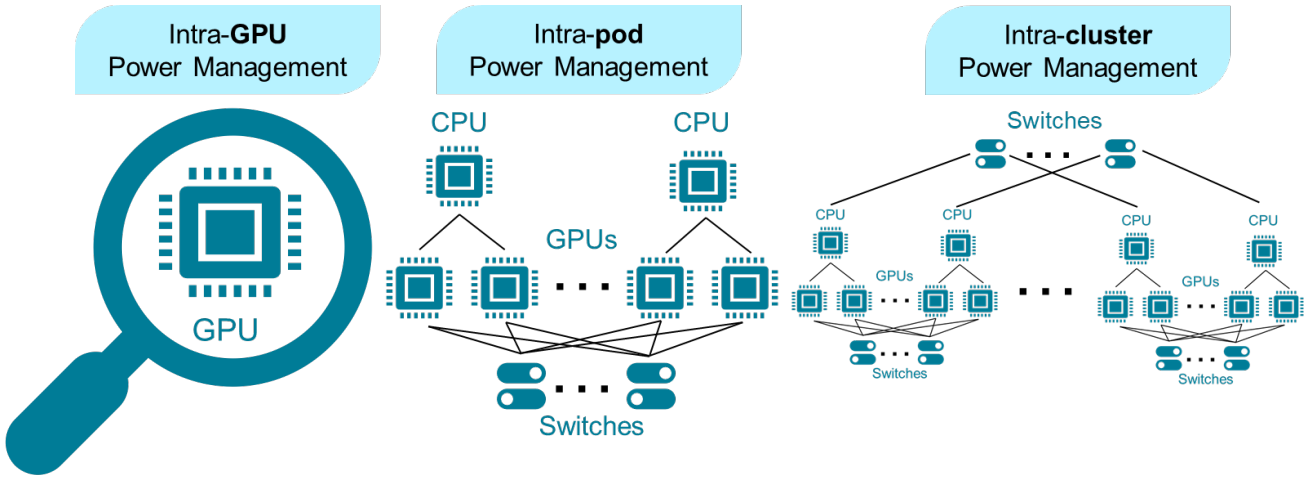
ISC High Performance 2026 - EESP Workshop

AMD
together we advance_

Key Message of ComPow

Component-aware, fine-grain power management inside of single GPU is crucial

Hierarchical approach to power management hinges on efficient intra-GPU power management



Component-awareness delivers ~10% energy savings for standalone executions at minimal performance loss

Component-awareness delivers ~4-5% execution uplifts for concurrent executions

Sustaining the AI Wave Calls for GPU Power Optimizations

AI is ubiquitous and will increasingly be so

Finance, healthcare, programmer productivity, chatbots, education, & more

"Electricity demand from AI-optimized data centers projected to more than quadruple by 2030." IEA "Energy & AI", Apr'25

"AI will propel datacenters to use 4.5% of global energy generation by 2030" versus 2% in 2025

SemiAnalysis "AI Datacenter Energy Dilemma", Mar'24

GPU = key workhorses of an AI datacenter

≥50% of power allocation in a typical AI data-center is to GPUs

SemiAnalysis "AI Datacenter Energy Dilemma", Mar'24

Performance = Power

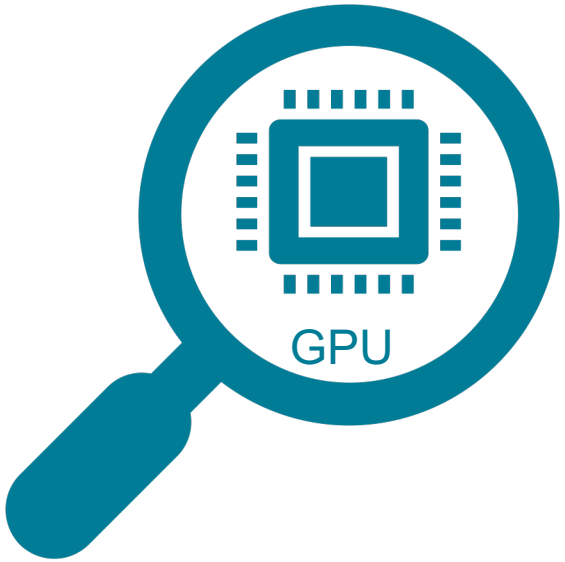
AI workloads are extremely power hungry & run at close to peak power

LBNL "Data Center Energy Usage", Dec'24;
SemiAnalysis "AI Datacenter Energy Dilemma", Mar'24

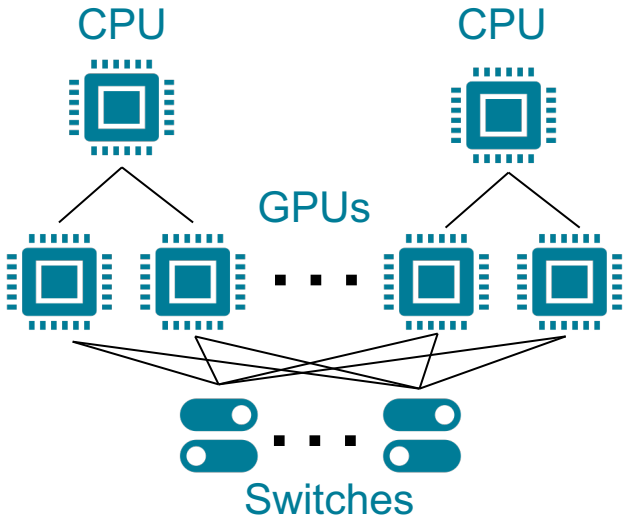
Hierarchical Approach to Power Management

- Hierarchical approach to power management widens aperture of impact
  - GPU → Pod → Cluster
- Intra-GPU smart power management key piece to enable efficient hierarchical power management
  - Controlling most power-hungry component aka GPU can deliver high ROI

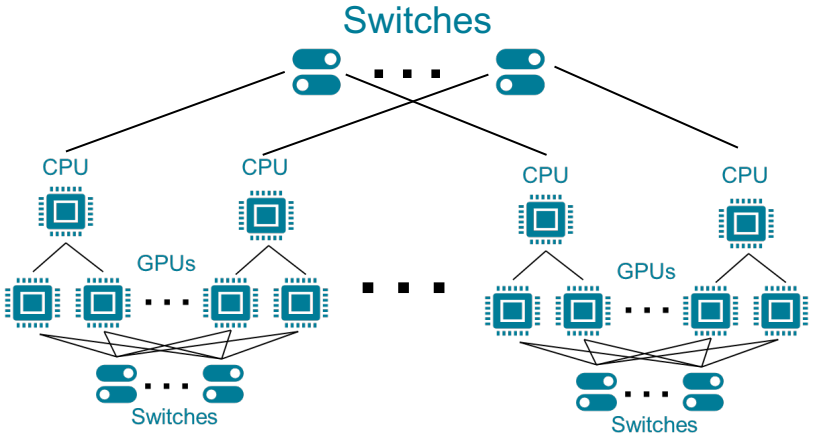
Intra-**GPU** Power Management



Intra-**pod** Power Management

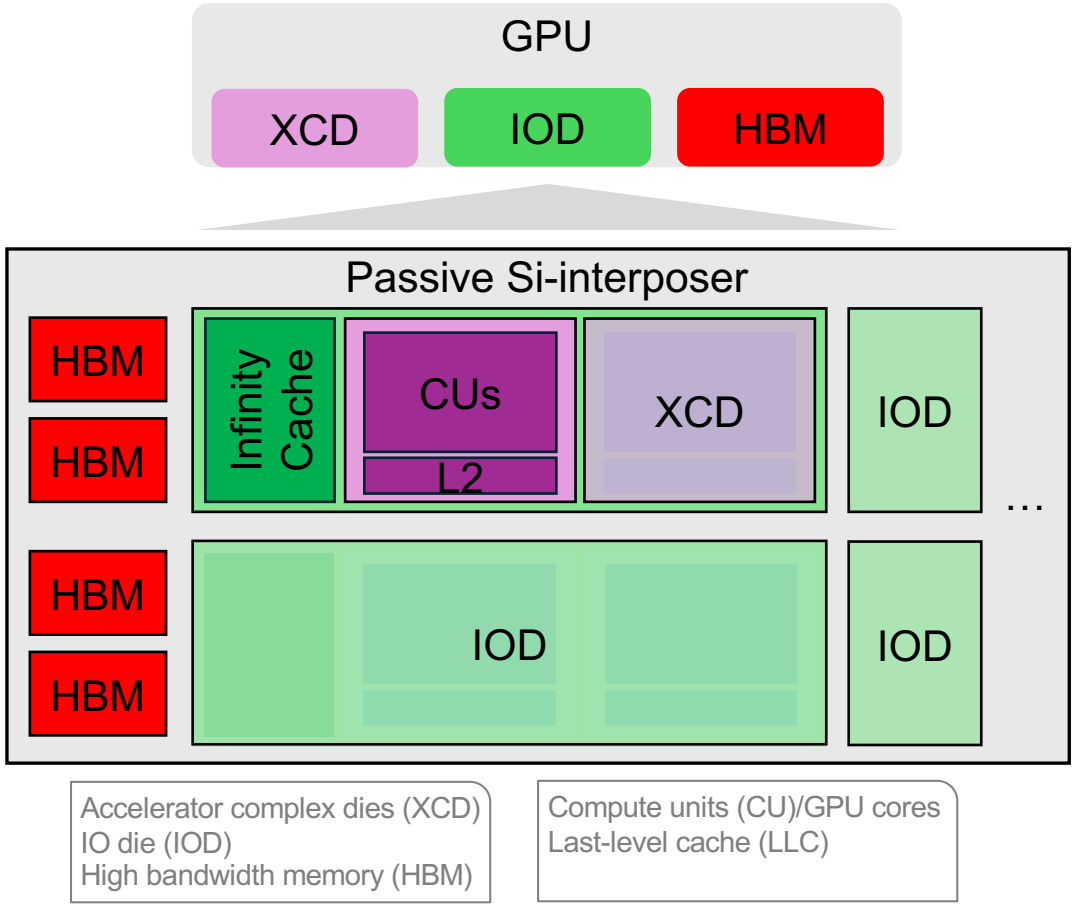


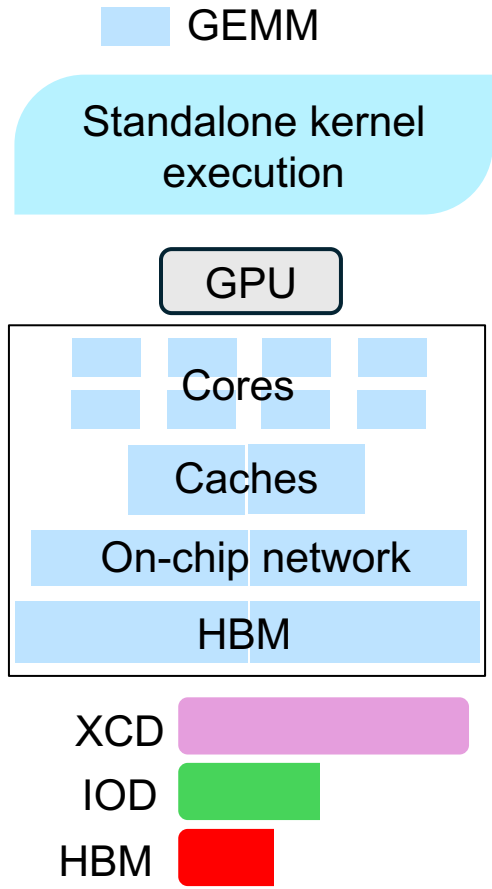
Intra-**cluster** Power Management

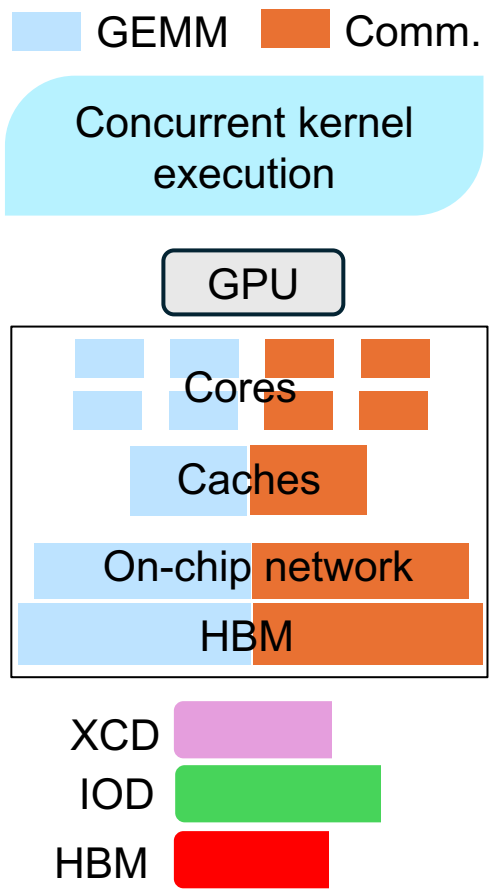


Component-awareness for Intra-GPU Power Management

- Different kernels and execution patterns can manifest different component-level power signatures
  - Component-level power controls & management can unlock additional avenues for smarter intra-GPU power allocation



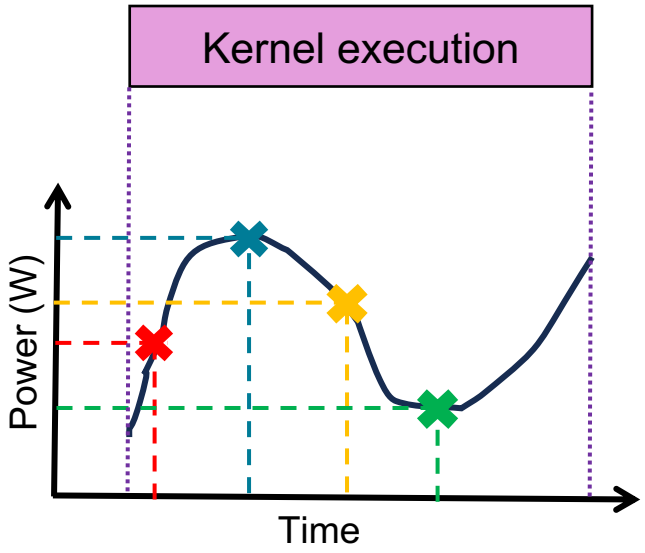




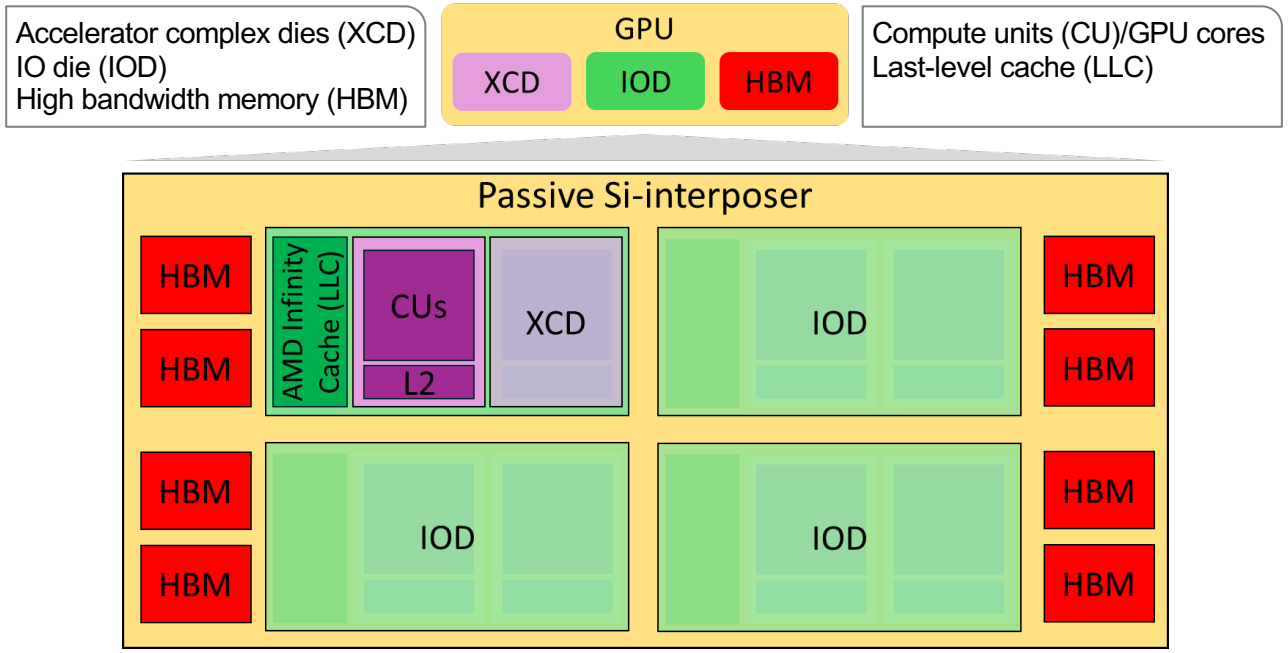
Fine-grain Power Profiling to Study Component-Awareness

- What is a fine-grain power profile?
  - Ideally, power at different points in a single kernel's execution (fine-grain in time dimension)
  - Component-level power breakdown (e.g., XCD, IOD, HBM; fine-grain in space dimension)
- Fine-grain power profiling can help maximize performance within power constraints
  - Examples: New algorithmic techniques to fully utilize power envelop, equip firmware/scheduler to power slosh at finer-granularity, etc.

Fine-grain: snapshots during execution



Fine-grain: AMD Instinct™ MI300X GPU component breakdown



FinGraV: Methodology for Fine-Grain GPU Power Measurements

- FinGraV addresses many challenges with fine-grain power measurements

- Challenge-1: Kernel executions are sub-millisecond while power sampling is multi-milliseconds
  - FinGraV: Execute multiple kernel executions in a single run & stitch together multiple runs with random start delays

- Challenge-2: Kernel start/end events in CPU time-domain while power logging in GPU time-domain
  - FinGraV: Provide methodology to correlate CPU-GPU time

- Challenge-3: Kernels manifest execution time variation
  - FinGraV: Execution time binning and explicit outlier handling

2025 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)

FinGraV: Methodology for Fine-Grain GPU Power Visibility and Insights

Varsha Singhania
Advanced Micro Devices, Inc.
varsha.singhania@amd.com

Shaizeen Aga
Advanced Micro Devices, Inc.
shaizeen.aga@amd.com

Mohamed Assem Ibrahim
Advanced Micro Devices, Inc.
mohamed1.ibrahim@amd.com

***Abstract*—Ubiquity of AI makes optimizing GPU power a priority as large GPU-based clusters are often employed to train and serve AI models. An important first step in optimizing GPU power consumption is high-fidelity and fine-grain power measurement of key AI computations on GPUs. To this end, we observe that as GPUs get more powerful, the resulting sub-millisecond to millisecond executions make fine-grain power analysis challenging. In this work, we first carefully identify the challenges in obtaining fine-grain GPU power profiles. To address these challenges, we devise FinGraV methodology where we employ execution time binning, careful CPU-GPU time synchronization, and power profile differentiation to collect fine-grain GPU power profiles across prominent AI computations and across a spectrum of scenarios. Using the said FinGraV power profiles, we provide both, guidance on accurate power measurement and, in-depth view of power consumption on state-of-the-art AMD Instinct™ MI300X. For the former, we highlight a methodology for power differentiation across executions. For the latter, we make several observations pertaining to GPU sub-component power consumption and GPU power proportionality across different scenarios. We believe that FinGraV unlocks both an accurate and a deeper view of power consumption of GPUs and opens up avenues for power optimization of these ubiquitous accelerators.**

***Index Terms*—AI, fine-grain power analysis, GPU**

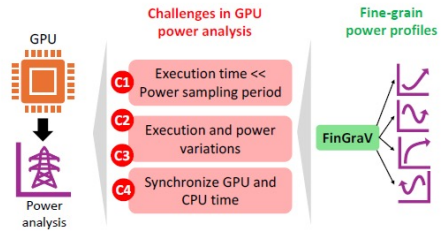


Fig. 1. FinGraV addresses challenges in fine-grain GPU power analysis.

power consumption into sub-components (e.g., compute cores, memory, etc.).

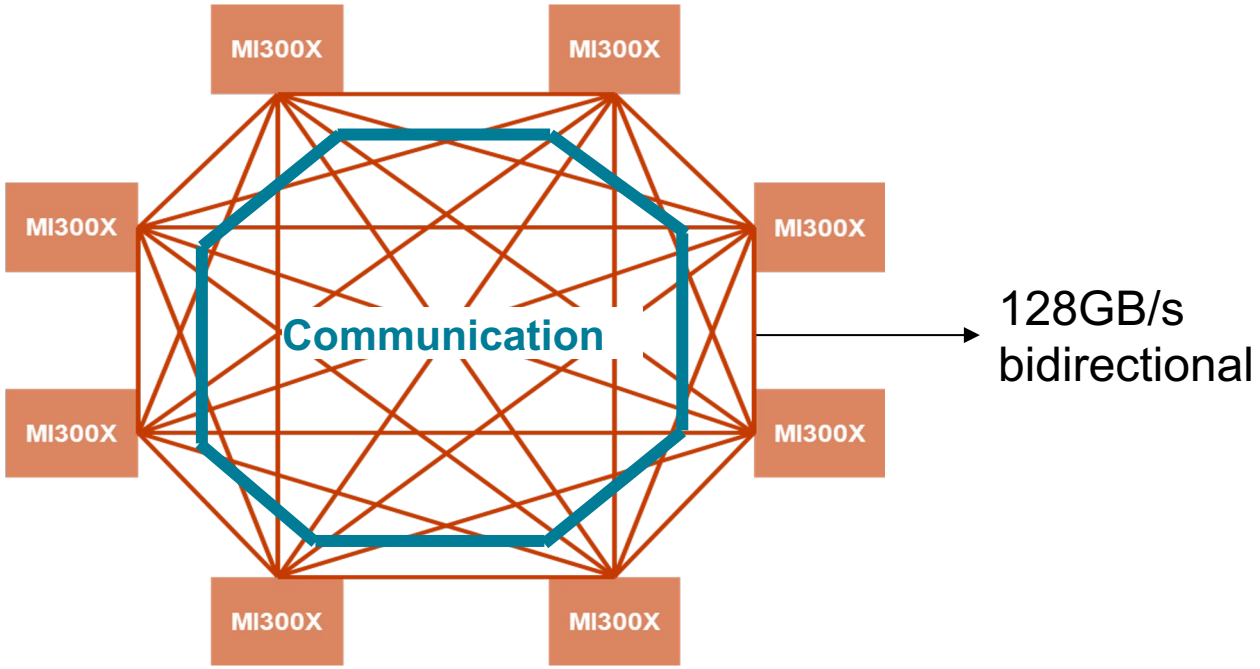
Note that, we focus on accurate kernel-level power profiles for several reasons. First, as our work and prior work [4] show, large-scale ML compute kernels can often be GPU power limited. That is, hitting GPU power limit during kernel execution causes frequency throttling [5] leading to lower performance.

https://ieeexplore.ieee.org/document/11096390

Experimental Setup & ML Scenarios Covered

- System evaluated
  - Hardware: 8× GPU AMD Instinct™ MI300X Platform in a fully-connected topology via AMD Infinity Fabric™ technology
  - Software stack: ROCm™ 6.4.0, RCCL 2.22.3 (collectives library), rocBLAS 4.4.0 (GEMM library)
- Coverage for key ML primitives of interest → Maximal execution time coverage
  - Primitives: Matrix-matrix multiplication (GEMM), Communication collectives (all-gather)
  - Execution mode: Standalone, concurrent

Single node with 8× AMD Instinct™ MI300X GPUs

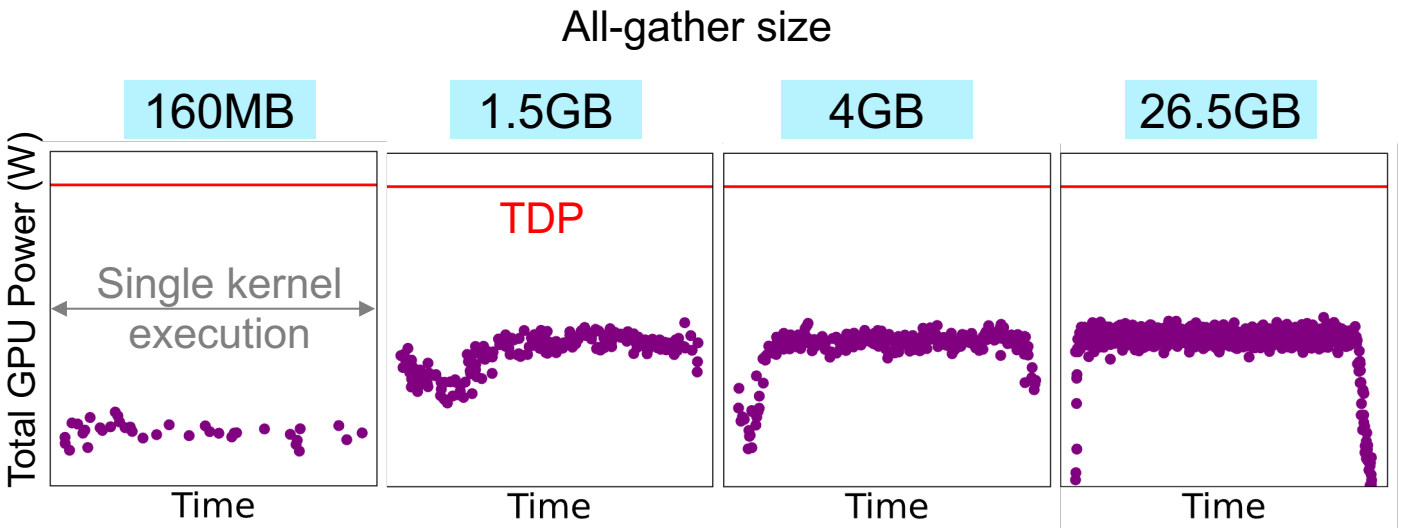


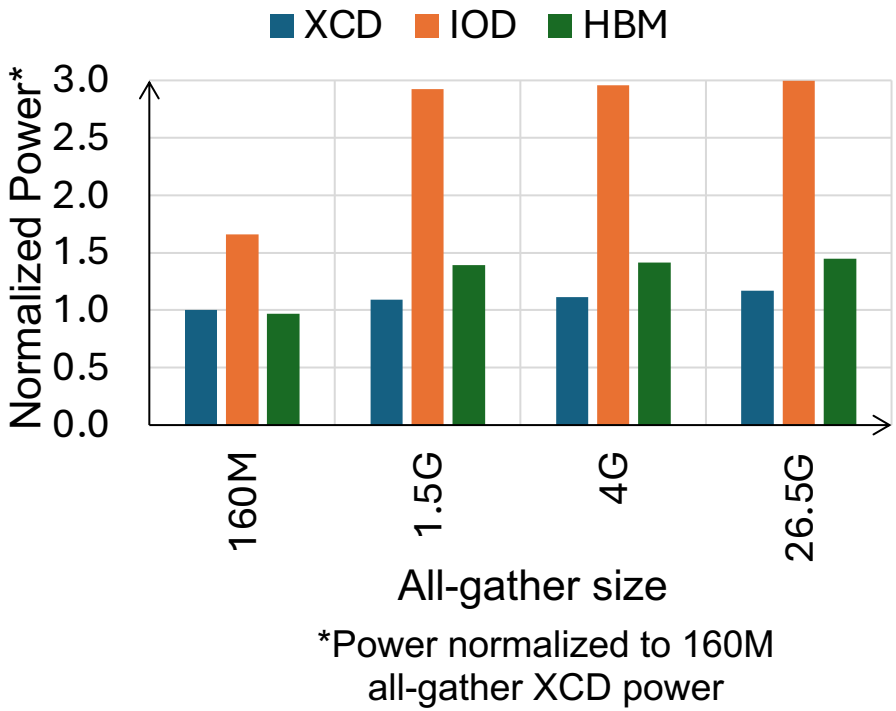
ML Training & Synthetic Scenarios

GEMM (M-N-K)	All-gather	Source
16384-106496-8192	4GB, 26.5GB	LLaMA-405B
18432-16384-16384	1.5GB, 3.5GB	LLaMA-405B
8192-57344-8192	7GB	LLaMA-70B
8192-8192-10240	160MB	Synthetic

ML Communication Stresses Data-movement Components

- Communication collective kernels launch just enough workgroups to saturate link bandwidth
- Data-movement (IOD, HBM) key power consumers for ML communication collectives



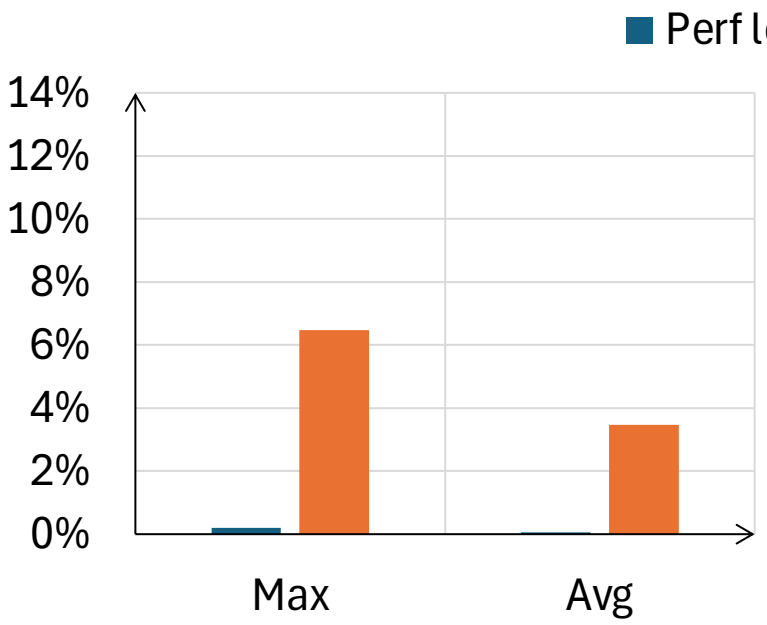


*Power normalized to 160M all-gather XCD power

ComPow for ML Communication: XCD Frequency Capping

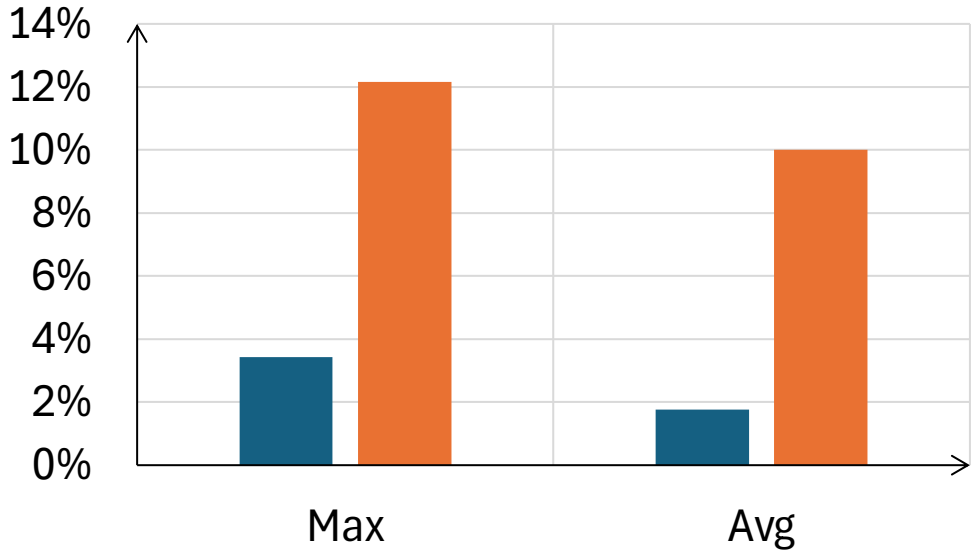
- Power capping is component-unaware & does not provide a good performance/power tradeoff
- Capping XCD frequency is component-aware and acknowledges that communication does not stress XCD power
- Component-aware frequency control provides a better lever than component-unaware power capping

Component-unaware Power Capping



750W vs. 500W

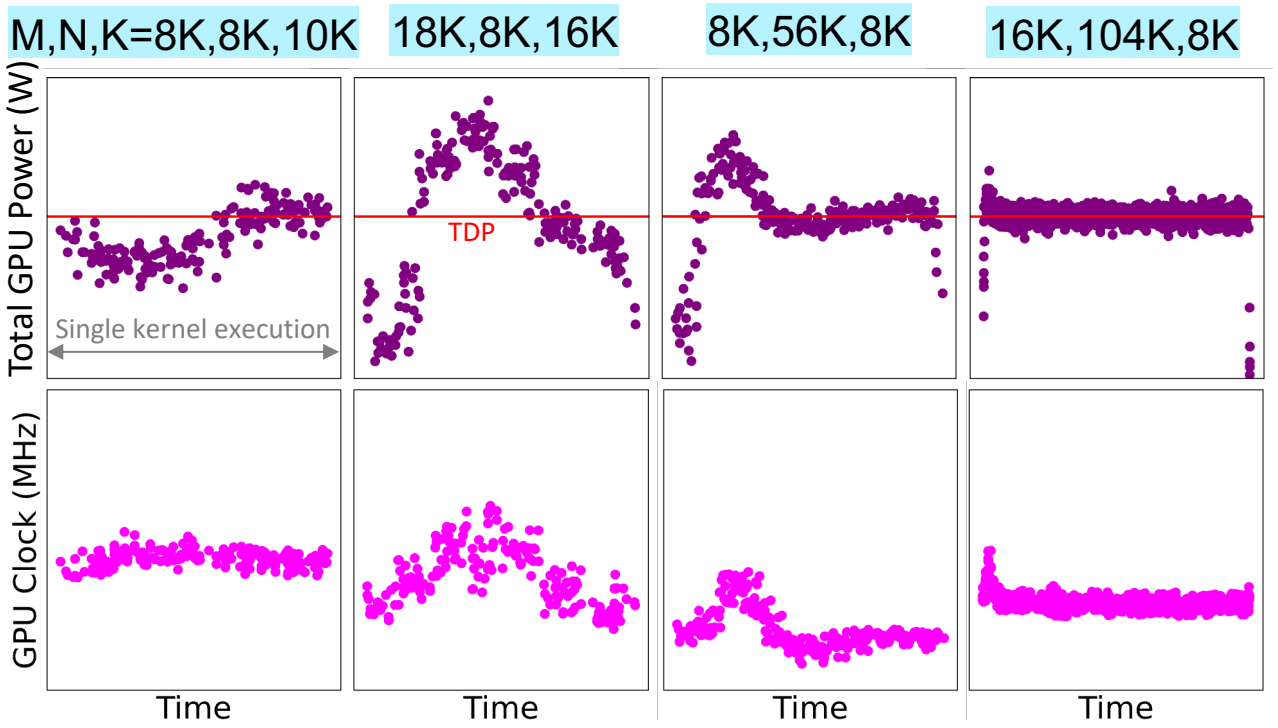
Component-aware Frequency Control

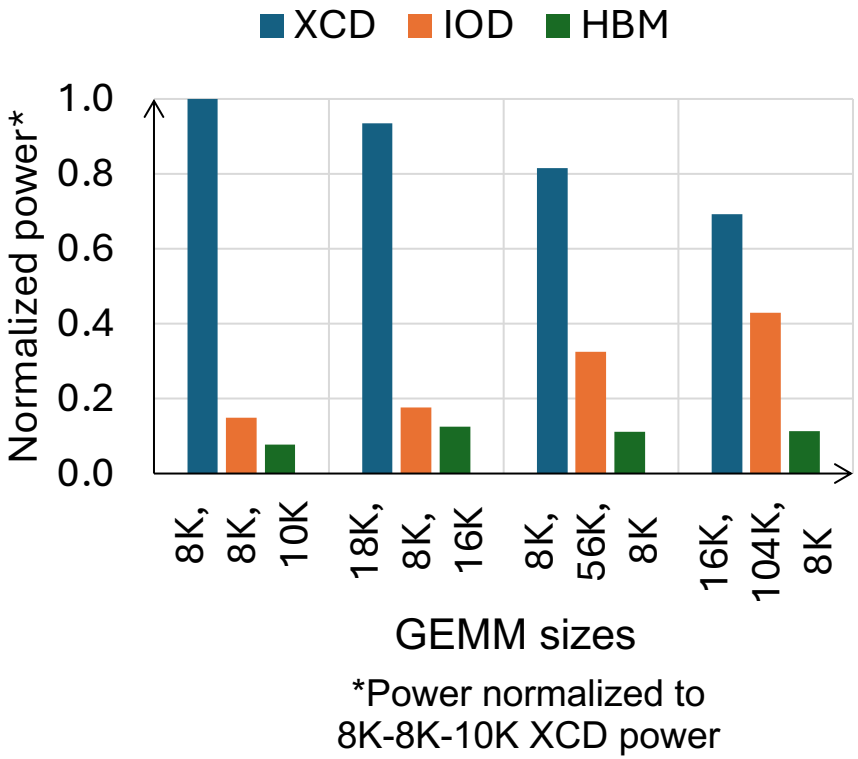


2100MHz vs. 1600MHz

GEMM Is Power-constrained & Can Stress All Components

- GEMM kernels stress GPU power
- GEMM kernels typically stress XCD power
- GEMMs with lower measured arithmetic intensity can stress data-movement components (IOD, HBM)

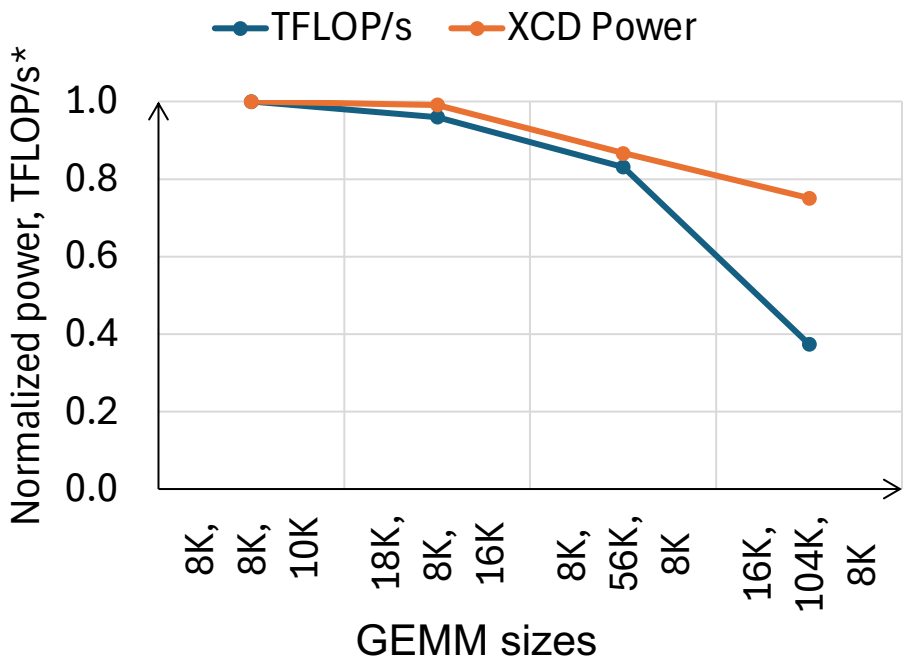




ComPow Potential for GEMMs: Phase-level Power Sloshing

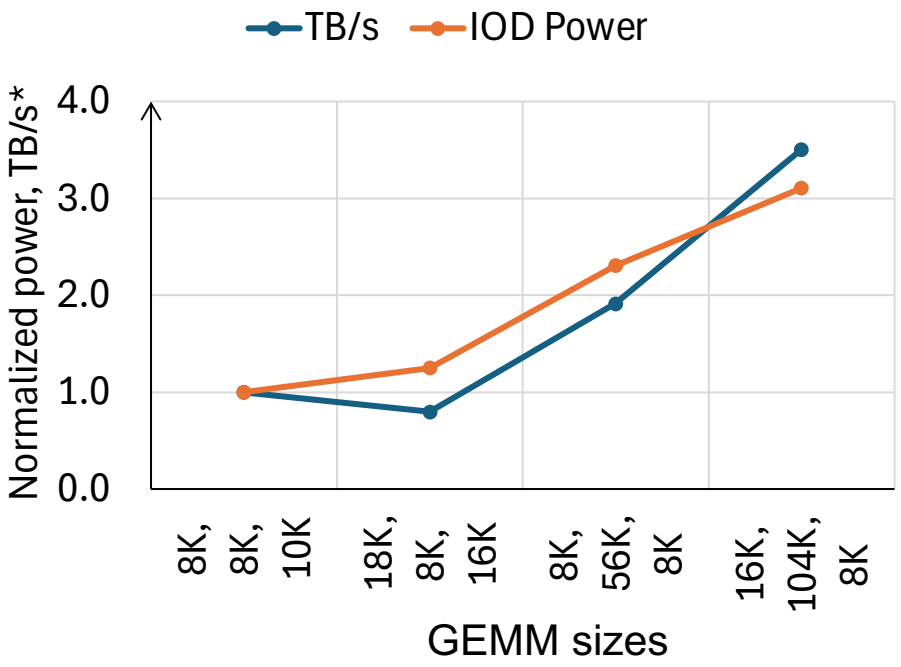
- Today: Limited component-level knobs in GPUs
- ComPow potential: Key utilization statistics track well with power utilization
  - GEMM phase-level software hints can allow power manager to do intelligent intra-GPU power sloshing across components

XCD power and TFLOP/s



*Power & TFLOP/s normalized to
8K-8K-10K XCD power & TFLOP/s respectively

IOD power and TB/s

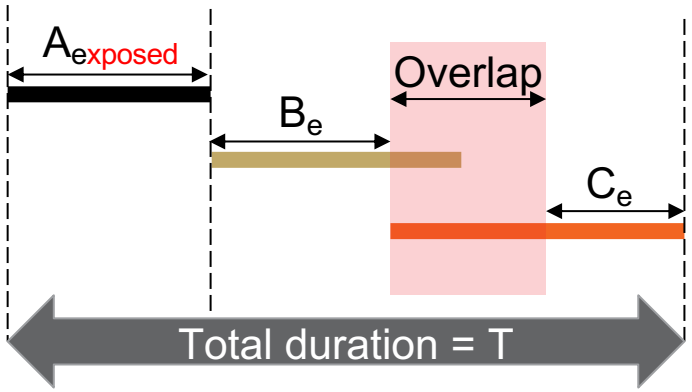


*Power & TB/s normalized to
8K-8K-10K IOD power & TB/s respectively

Computation-Communication Concurrency: Crucial ML Paradigm

- ML algorithms & system-level optimizations aim to overlap both data-independent & data-dependent communication
- State-of-the-art ML training demonstrates 38-47% computation & communication overlap

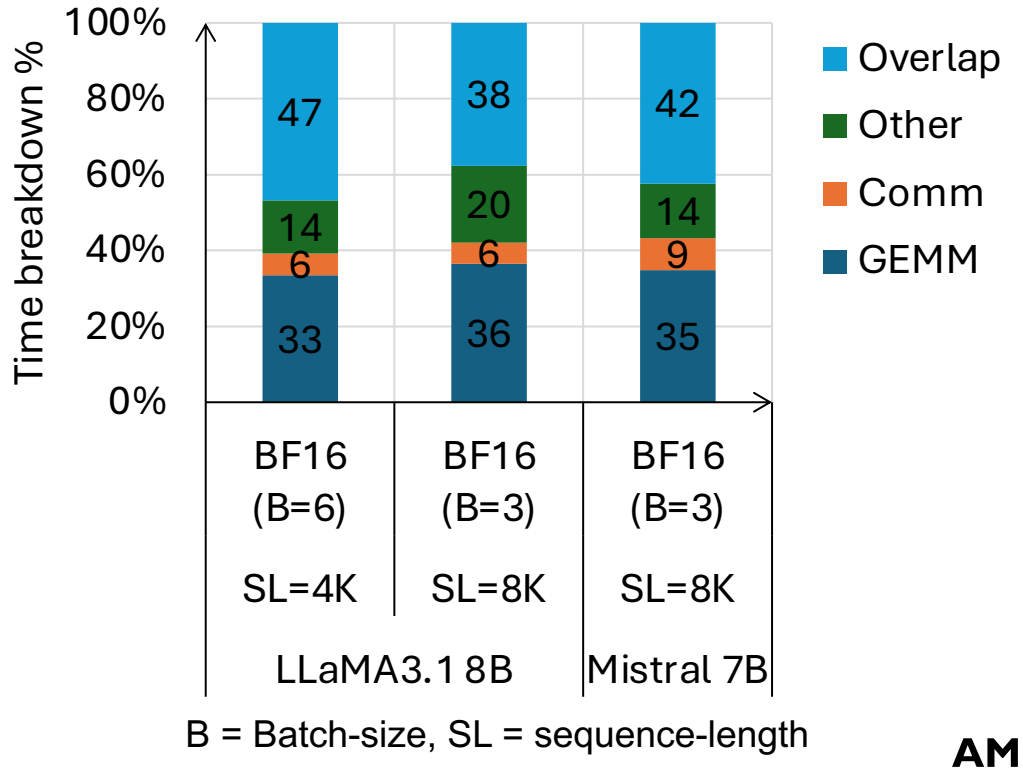
Execution Time Breakdown Visualization



$A_e\% = A_e / T$
$B_e\% = B_e / T$
$C_e\% = C_e / T$
$O\% = Overlap / T$

Sum(%) = 100%

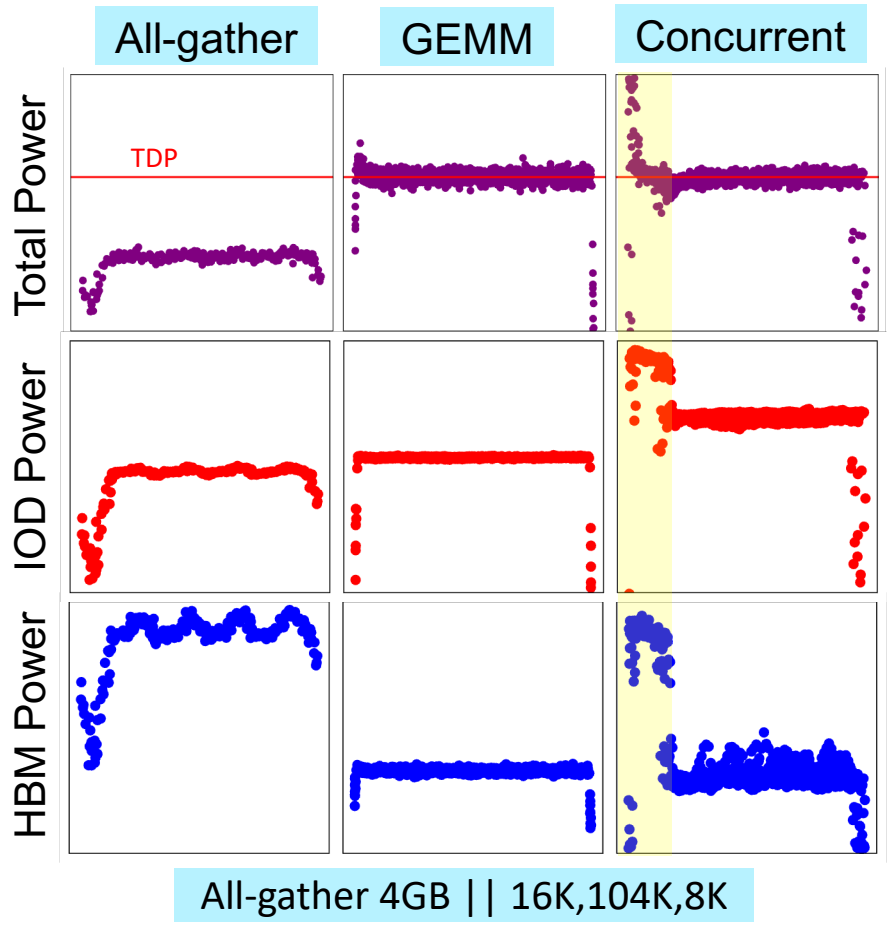
ML Training Execution Time Breakdown

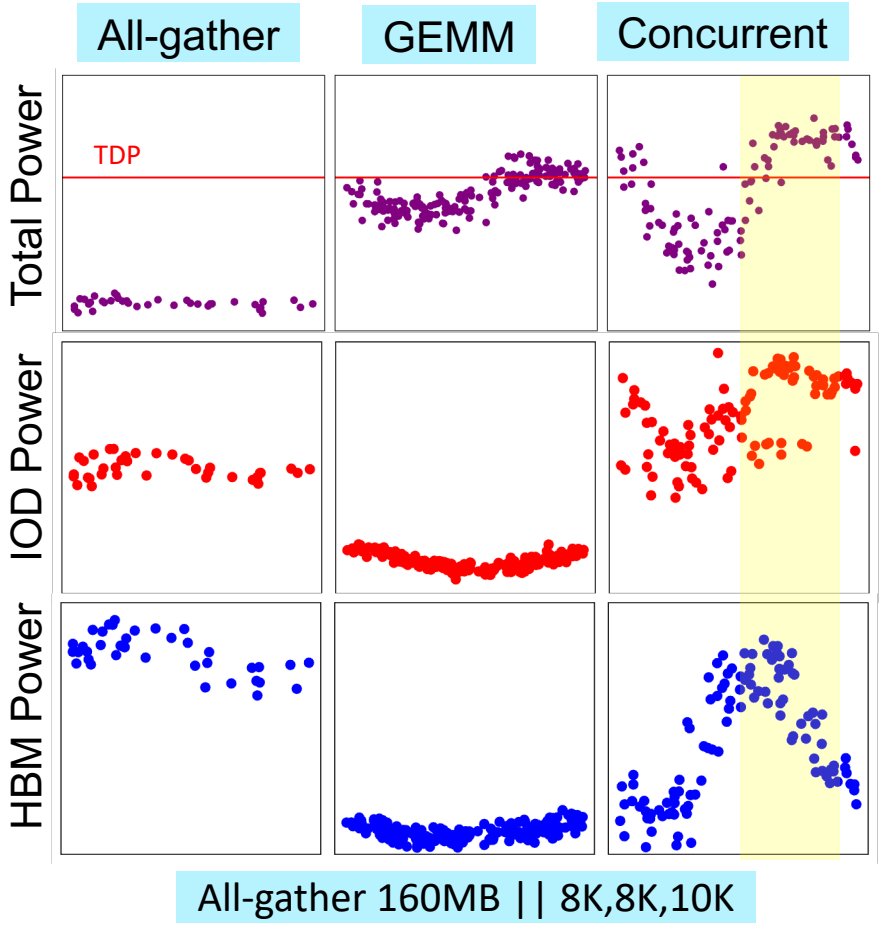


B = Batch-size, SL = sequence-length

Concurrency Stresses Data-movement Components

- IOD and HBM power during concurrency is higher than standalone GEMM or communication IOD/HBM components
  - GEMM and communication collectives both stress data-movement components

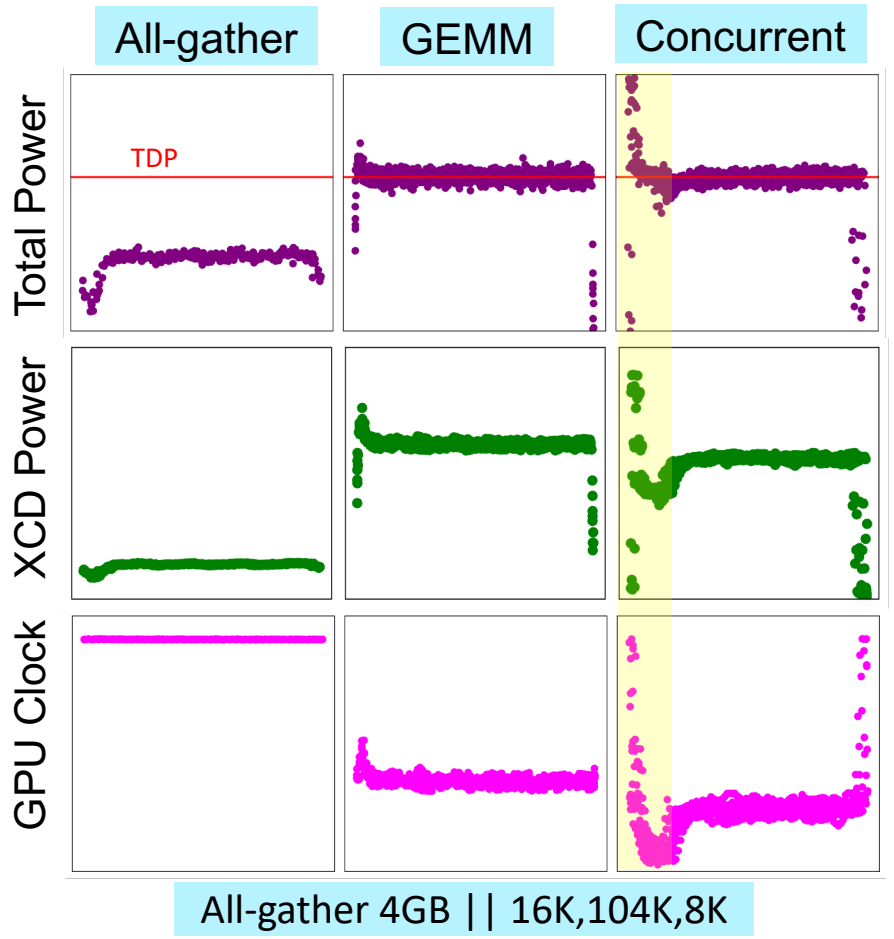


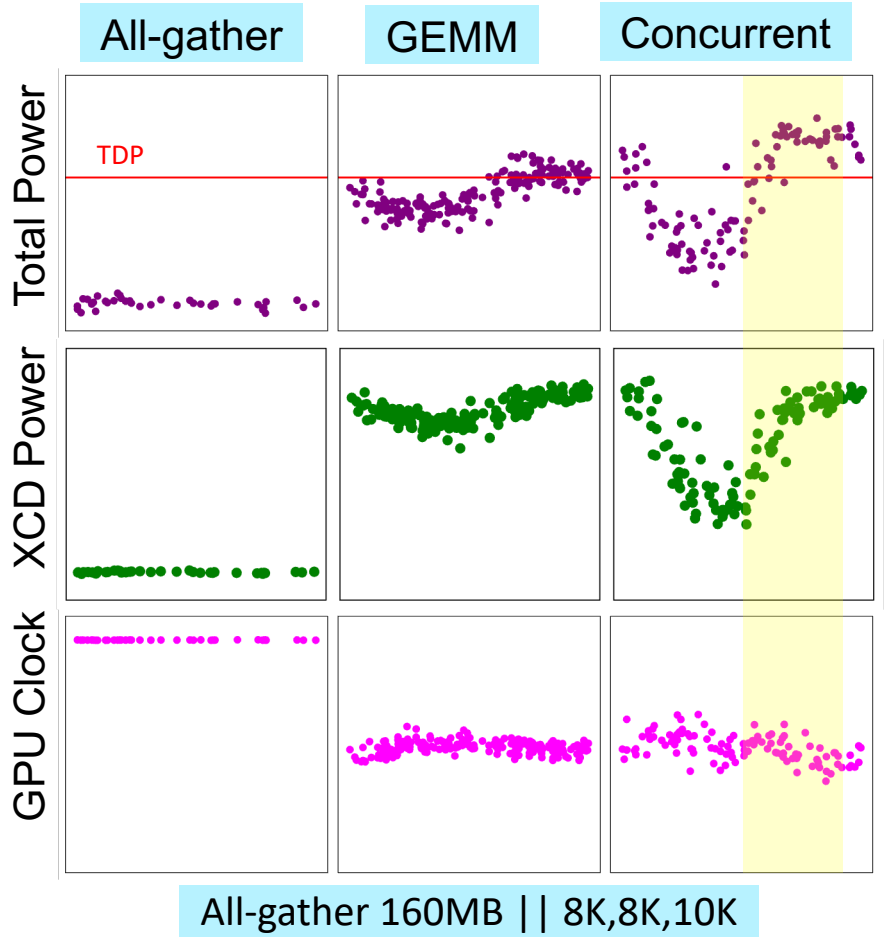


Concurrent kernel execution

Concurrency Trades-off XCD Power

- Concurrency IOD/HBM rise is compensated by lowering XCD power
  - XCD power during concurrency is lower than GEMM XCD power





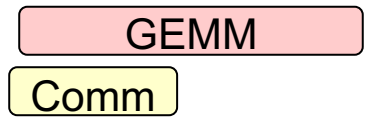
Concurrent kernel execution

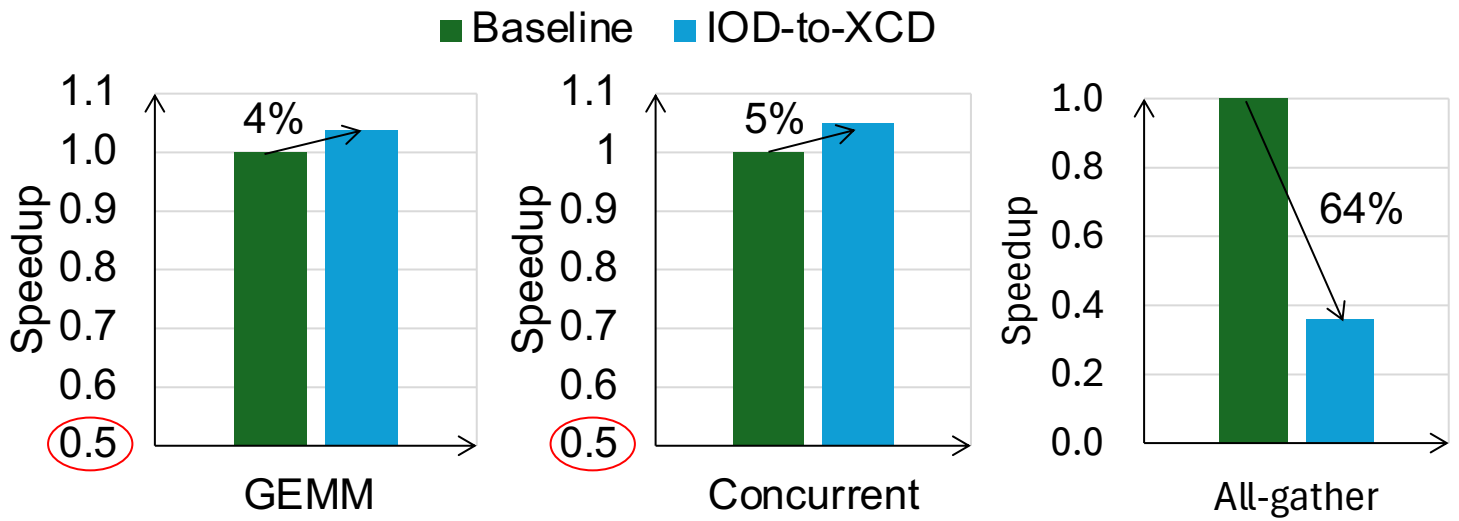
ComPow for Concurrency: Power Reallocation for Performance

- ComPow Emulation: Use static resource partitioning to allocate fewer cores to Comm (slosh power to XCD and GEMM)
  - Power reallocation delivers 4% higher GEMM performance and 5% higher concurrent performance
  - Beneficial for scenarios when GEMM is on critical path

All-gather 160MB || 8K,8K,10K

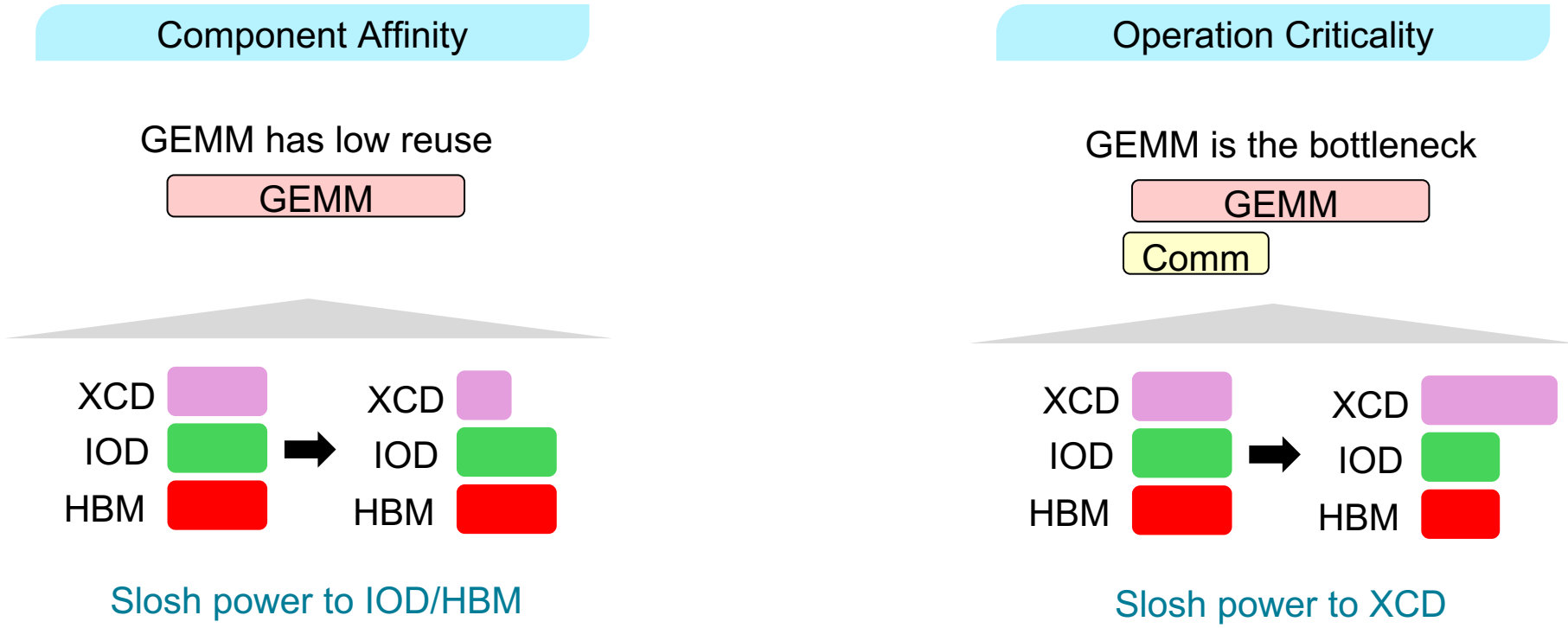
GEMM is the bottleneck





Hardware/Software Co-design to Realize ComPow

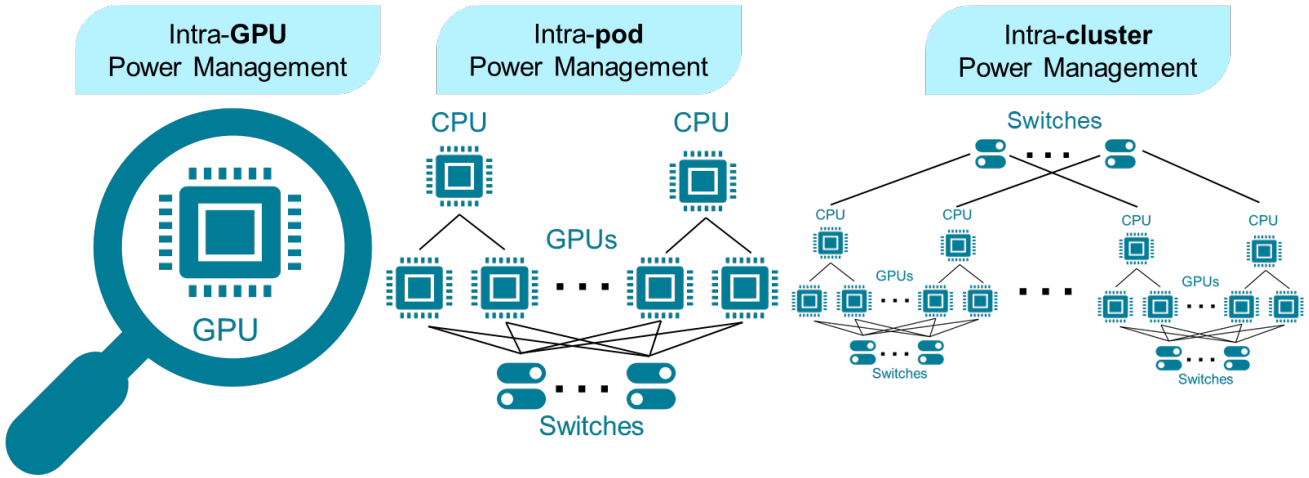
- Software hints can better guide power managers to slosh power amongst GPU components
- Repetitive & iterative nature of ML & HPC workloads can also provide guidance to power managers



Conclusion

Component-aware, fine-grain power management inside of single GPU is crucial

Hierarchical approach to power management hinges on efficient intra-GPU power management



Component-awareness delivers ~10% energy savings for standalone executions at minimal performance loss

Component-awareness delivers ~4-5% execution uplifts for concurrent executions

CompPow: A Case for Component-level GPU Power Management

Shaizeen Aga and Mohamed Assem Ibrahim
June 26th, 2026

ISC High Performance 2026 - EESP Workshop

AMD
together we advance_

COPYRIGHT AND DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED “AS IS”. AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD Arrow logo, Instinct, Infinity Fabric, ROCm, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.


AMD