AMD กับ vLLM

มีอะไรใหม่ในปี 2026



Passin Pornvoraphat

AMD Champion Hackathon: ACT II (Unicorn Track)

เนื้อหาเรียบเรียงจากการบรรยายของ Doug Lehr, AMD Fellow และ maintainer ของ vLLM

หัวข้อ

1. AMD ใน vLLM วันนี้
2. 2026 ปีที่ย้ายไป upstream
3. Kernel: FlyDSL, AITER, ATOM
4. เครื่องมือ: Quark และ TraceLens
5. vLLM บน Navi
6. สิ่งที่กำลังจะมา
7. Repo ที่แนะนำ


AMD
NYSE Wired
Google DeepMind
theCUBE
Gemma 4
LAB LAB
PRIZE POOL
$20 000+
AMD DEVELOPER HACKATHON
WINNERS ANNOUNCEMENT
20,728 BUILDERS.
THESE ARE OUR CHAMPIONS.

1

หัวข้อที่ 1 จาก 7

AMD ใน vLLM วันนี้

1 **AMD ใน vLLM วันนี้**

2 2026 ปีที่ย้ายไป upstream

3 Kernel: FlyDSL, AITER, ATOM

4 เครื่องมือ: Quark และ TraceLens

5 vLLM บน Navi

6 สิ่งที่กำลังจะมา

7 Repo ที่แนะนำ

AMD ใน vLLM: ตัวเลขสำคัญ

ส.ค. 2025 ถึง ก.ค. 2026

1,041

PR ของ AMD และ ROCm ที่ merge เข้า vLLM

14.2%

สัดส่วนของ PR ทั้งหมดใน vLLM เดือน ก.ค. 2026

174

ผู้เขียนโค้ดที่ส่ง PR เหล่านี้

73%

PR ที่เขียนโดยวิศวกรของ AMD

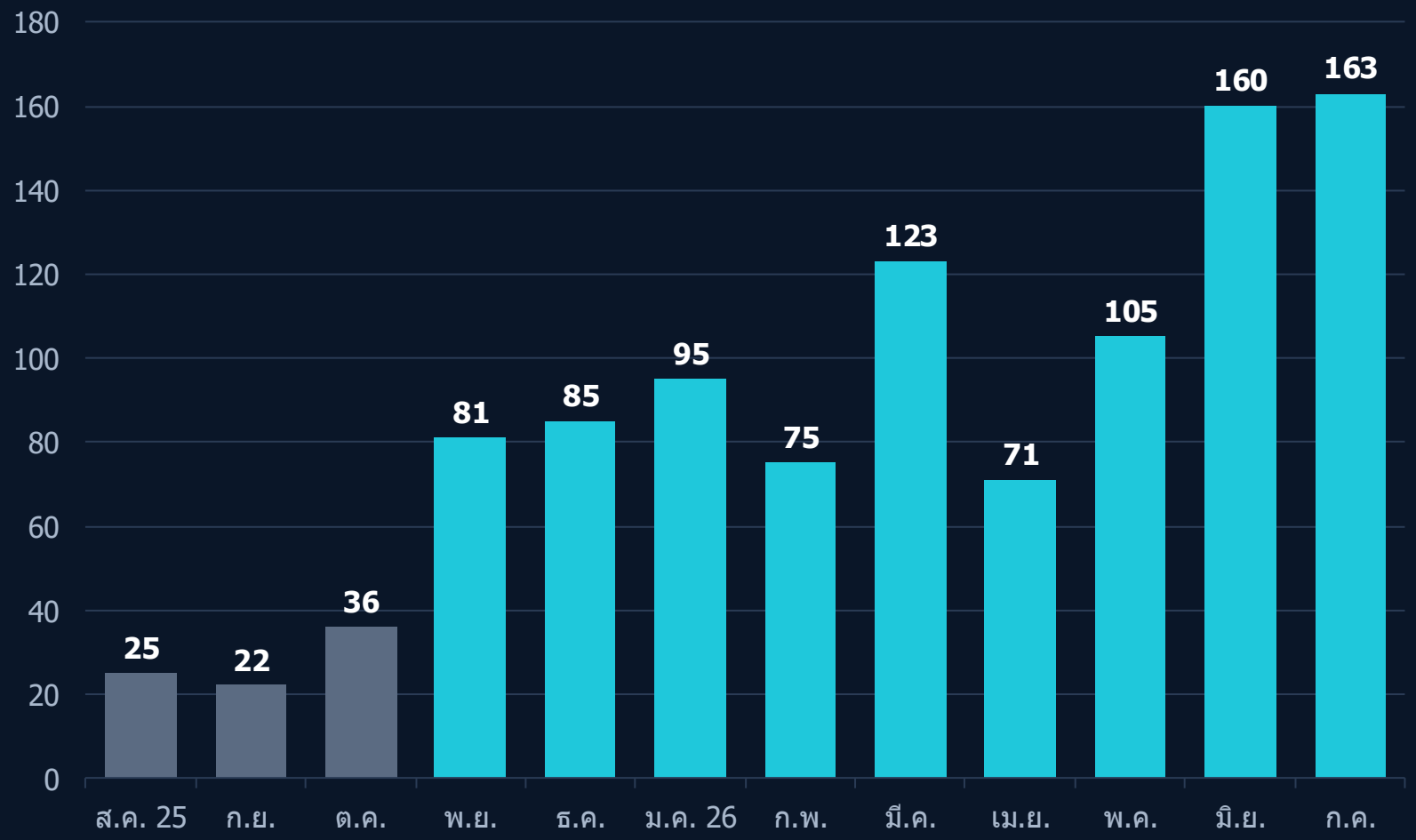
48

วิศวกร AMD ที่ส่งงานเข้า vLLM ครั้งแรกในปี 2026

6.5x

การเติบโตของจำนวน merge ต่อเดือน

PR ของ ROCm ที่ merge ต่อเดือน



นับจาก PR ที่ merge แล้วและมีคำว่า rocm, aiter, amd, instinct หรือ gfx9 ในชื่อ

25 → 163

PR ต่อเดือน ใน 12 เดือน

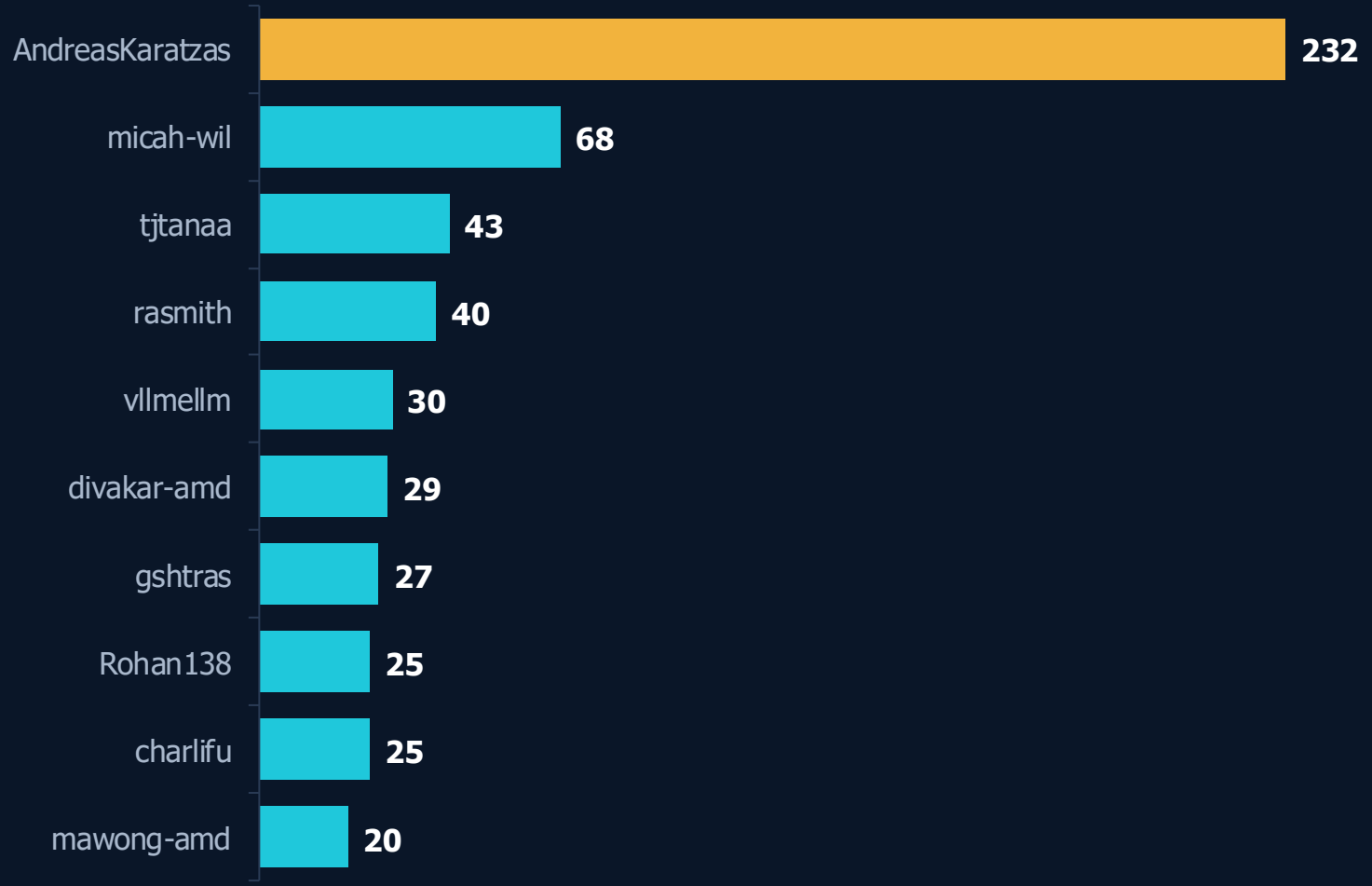
2.8% → 14.2%

สัดส่วนต่อ PR ทั้งหมดของ vLLM

ส.ค. ถึง ต.ค. 2025 ซึ่งเป็นแท่งสีเทา ยังต่ำ แล้วพุ่งขึ้นตั้งแต่ พ.ย.

ใครเป็นคนทำงาน

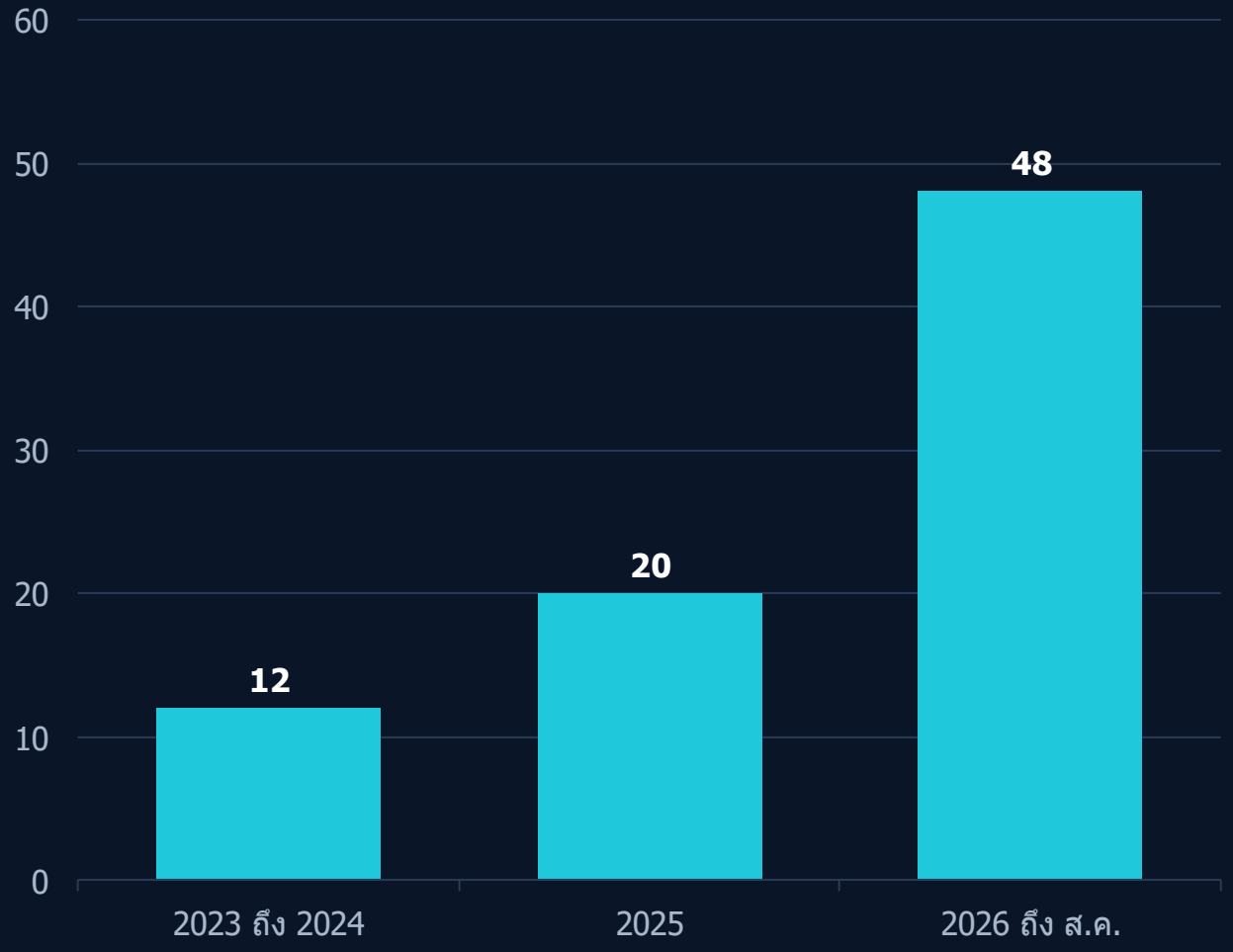
วิศวกร 10 อันดับแรกที่ส่ง PR ของ ROCm เข้า vLLM



- มีคนหนึ่งที่ส่งงานมากเป็นพิเศษ คือ 232 PR
- 10 อันดับแรกรวมกันได้ราวครึ่งเดียว อีกครึ่งมาจากผู้เขียนอีก 163 คน
- ทั้ง 10 คนอยู่ที่ AMD หรือ Embedded LLM
- งานหลักคือ CI ความเร็วของโมเดล และการแก้ bug

วิศวกร AMD หน้าใหม่ใน vLLM

จำนวนวิศวกร AMD ที่ส่งงานเข้า vLLM เป็นครั้งแรก



95 จาก 174

ผู้เขียน ROCm PR ที่ส่งงานเข้า vLLM ครั้งแรกในปีนี้

7 ถึง 10 คน

คนใหม่จาก AMD ต่อเดือน ตั้งแต่เดือน มี.ค.

ราว 60 คน

จำนวนคนใหม่ที่คาดไว้เมื่อสิ้นปี 2026

Code owner ฝั่ง AMD ใน vLLM มี 5 คน

ปีก่อนมี 2 คน ลดเหลือ 1 แล้วเพิ่มอีก 4 ในปี 2026 ทำให้ PR ของ AMD ได้รับการตรวจเร็วขึ้น

TJ

Tun Jian Tan

Embedded LLM ดูแลโค้ด AMD นานที่สุด

@tjtanaa

DL

Doug Lehr

Fellow ด้าน ML Frameworks

@dllehr-amd

AK

Andreas Karatzas

MTS ดูแล vLLM และ CI

@AndreasKaratzas

HY

Hongxia Yang

PMTS ดูแล vLLM และโมเดลใหม่

@hongxiayang

SS

Shanshan Shen

Sr Software ดูแล vLLM และ kernel

@shen-shanshan

2

หัวข้อที่ 2 จาก 7

2026 ปีที่ย้ายไป upstream

1 AMD ใน vLLM วันนี้

2 **2026 ปีที่ย้ายไป upstream**

3 Kernel: FlyDSL, AITER, ATOM

4 เครื่องมือ: Quark และ TraceLens

5 vLLM บน Navi

6 สิ่งที่กำลังจะมา

7 Repo ที่แนะนำ

จาก fork สู่ upstream

	สมัยใช้ fork	ตอนนี้
การ build และ release	AMD build Docker image และ wheel เอง	vLLM เป็นผู้ build และปล่อย nightly จาก CI ของ vLLM
ค่าตั้งของโมเดล	ค่าที่ดีที่สุดอยู่กับ AMD ใครอยากได้ต้องขอ	อยู่ใน vllm-project/recipes และ recipes.vllm.ai
การทดสอบ	CI ของ AMD แยกอยู่ upstream มองไม่เห็น	193 test group ใน CI ของ vLLM และ 73 กลุ่มกั้นการ merge
การแก้โค้ด	แก้ใน fork ก่อน แล้วค่อยส่งกลับทีหลัง	ส่งเข้า upstream ก่อนเสมอ

fork ROCm/vllm ใช้ตั้งแต่ พ.ย. 2023 และ commit สุดท้ายคือ 9 ก.ย. 2025 · ลูกค้าต้องการ vLLM upstream ที่ไม่ผูกกับผู้ผลิตรายใด

ติดตั้ง vLLM สำหรับ AMD จาก upstream

```
# Wheel จากคลัง ROCm ของ vLLM
uv pip install vllm \
  --extra-index-url https://wheels.vllm.ai/rocm/

# Docker image จากกระบวนการ release ของ vLLM
docker pull vllm/vllm-openai-rocm:v0.27.1

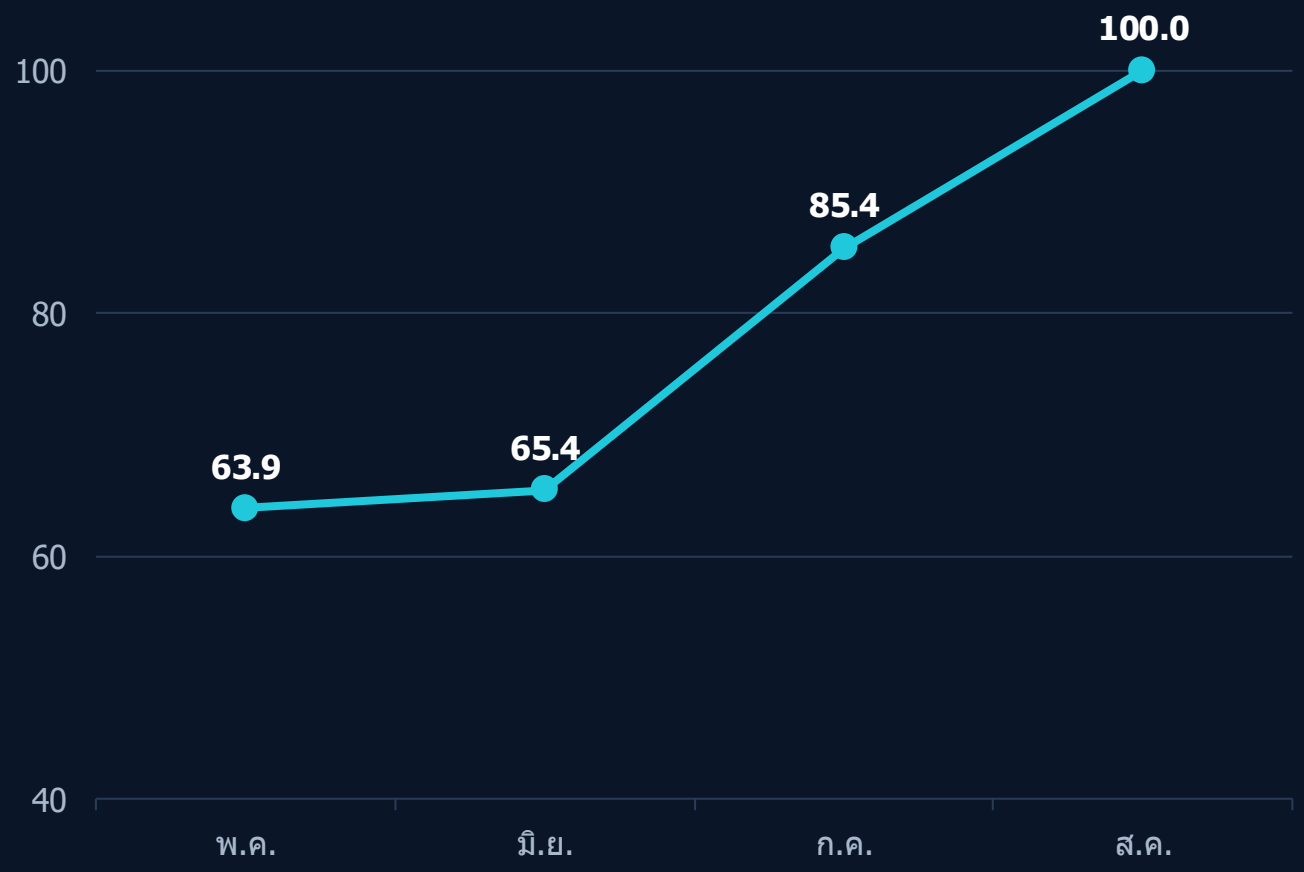
# รันโมเดล
export VLLM_ROCM_USE_AITER=1
vllm serve Qwen/Qwen3-235B-A22B-Instruct-2507 -tp 8
```

เหมือนกันทุกอย่าง

- Docker tag ใช้แบบเดียวกับของทั่วไป แค่ชื่อ image ลงท้ายด้วย -rocm
- มีทั้ง nightly และ release
- คำสั่งและฟีเจอร์ทำงานเหมือนเดิม
- ไม่มี fork และไม่มี patch เพิ่ม

CI ของ AMD ใน vLLM ปี 2026

อัตราผ่านการทดสอบของ AMD เป็นเปอร์เซ็นต์



96 → 193

test group ที่รันใน CI ของ vLLM

73 กลุ่ม

กั้นการ merge เพิ่มในปีนี้ 41 กลุ่ม

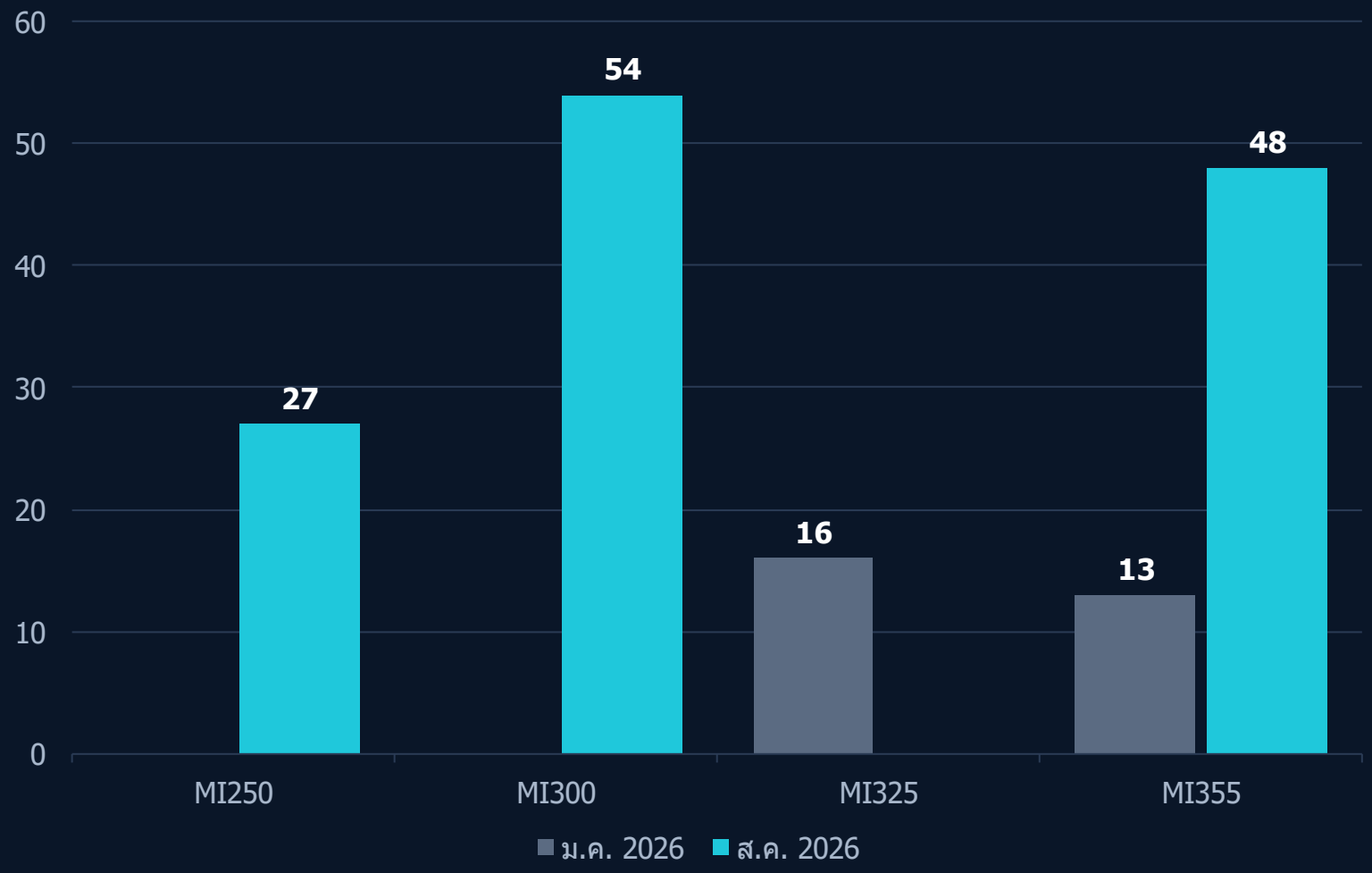
40 → 5 นาที

เวลา build Docker ต่อ PR

ถัดไป

ทดสอบ P/D, MoRI-IO, WideEP และ multi-node

เครื่องทดสอบ CI เพิ่มจาก 29 เป็น 128 node



29 → 128

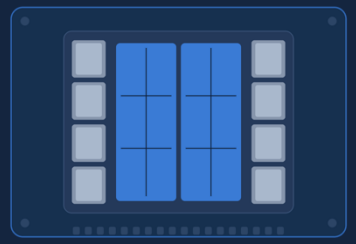
node สำหรับ CI บน GPU Instinct

- ครอบคลุม 3 สถาปัตยกรรม gfx
- ทุก PR ทดสอบบนเครื่องเหล่านี้
- ใช้วัดความเร็วทุกคืนร่วมกับ Perf Eval ของ vLLM

ตัวเลขรายรุ่นปัดเศษตามกราฟต้นฉบับ

รู้จัก GPU ตระกูล Instinct

GPU สำหรับศูนย์ข้อมูลและงาน AI ของ AMD เรียงจากรุ่นเก่าไปรุ่นใหม่



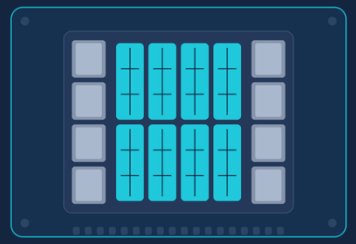
MI250

ออกปี 2021

CDNA 2 · gfx90a

128 GB HBM2e

3.2 TB/s



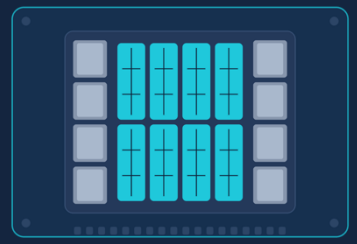
MI300X

ออกปี 2023

CDNA 3 · gfx942

192 GB HBM3

5.3 TB/s



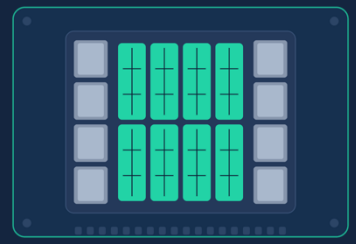
MI325X

ออกปี 2024

CDNA 3 · gfx942

256 GB HBM3E

6 TB/s



MI355X

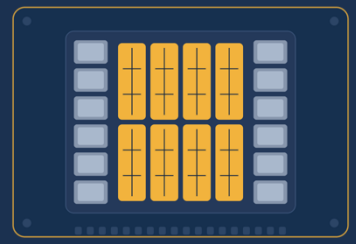
ออกปี 2025

CDNA 4 · gfx950

288 GB HBM3E

8 TB/s

รองรับ FP4 และ FP6 ในตัว



MI450

ออกปี 2026

รุ่นถัดไป · gfx1250

432 GB HBM4

19.6 TB/s

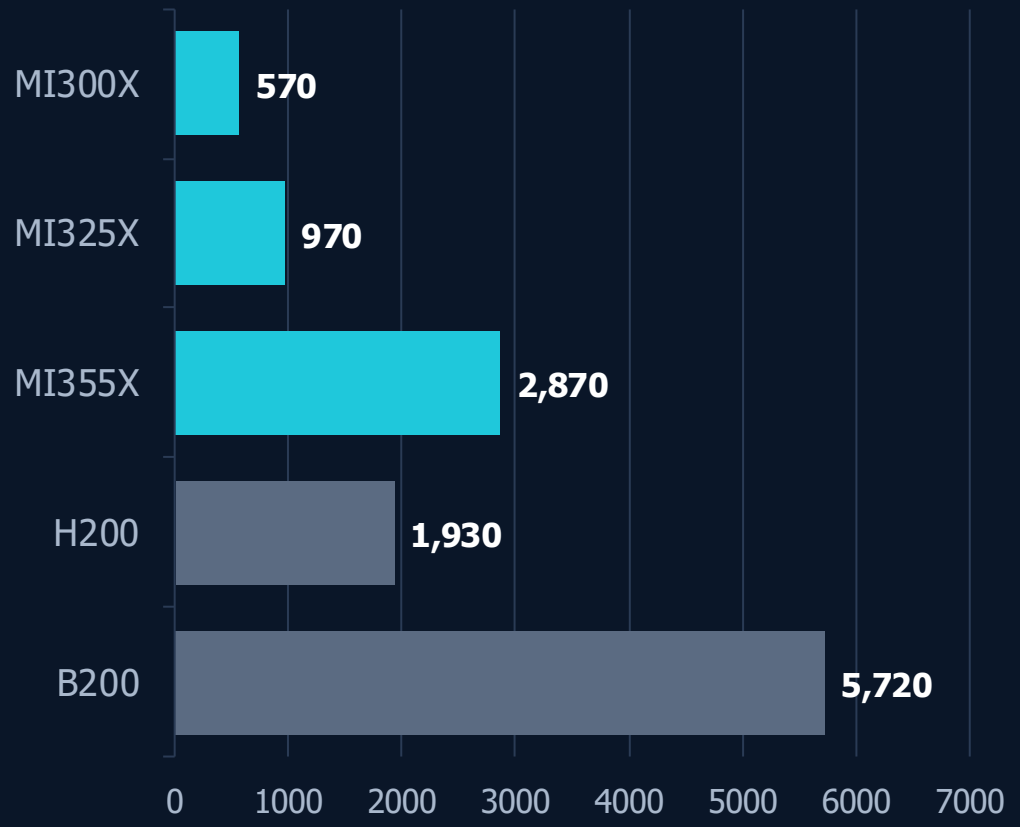
ยังไม่วางขาย ณ วันบรรยาย

ภาพเป็นภาพประกอบเชิงสัญลักษณ์ ไม่ใช่ภาพสินค้าจริง · หน่วยความจำและ bandwidth ควรตรวจกับเว็บไซต์ AMD ก่อนนำเสนอ

ความเร็วและต้นทุน: AMD เทียบ NVIDIA

DeepSeek R1 · ที่ความเร็วราว 33 token/s ต่อผู้ใช้ · ความเร็วจาก SemiAnalysis InferenceX

token/s ต่อ GPU (ยิ่งมากยิ่งดี)



	ต้นทุนเจ้าของเครื่อง		ราคาเช่าคลาวด์จริง	
GPU	$ / ชม.	$ / 1M tok	$ / ชม.	$ / 1M tok
MI300X	0.95	**0.46**	1.99	**0.97**
MI325X	1.10	**0.32**	–	–
MI355X	1.50	**0.15**	–	–
H200	1.22	**0.18**	6.48	**0.93**
B200	1.73	**0.08**	7.82	**0.38**

ราคาเช่าที่เห็นบนเว็บ ต.ค. 2026 · H200 จาก Nebius · B200 จาก Verda

ถ้าเช่าคลาวด์ MI300X ต้นทุนต่อ token ใกล้เคียง H200

DeepSeek R1 0528 · 8K/1K · แต่ละชิปใช้ config ที่ดีที่สุด precision อาจต่างกัน · token/s คำนวณจาก $ ต่อ 1M token · ต้นทุนเจ้าของเครื่องจาก SemiAnalysis ก.ค. 2026 · ไม่มีข้อมูล MI250, MI450

Day 0 สำหรับทุกอย่าง

โมเดลใหม่ รันได้ในวันเปิดตัว

DeepSeek V3 / V4 | Qwen 3.x | GPT-OSS | GLM 5.x | Kimi K2 / K3

- โมเดลที่เพิ่มเข้า upstream รันบน ROCm ได้จากโค้ดชุดเดียวกัน
- AMD ทำงานกับ vLLM และเจ้าของโมเดลก่อนวันเปิดตัว
- Quark ปล่อย checkpoint แบบ MXFP4, NVFP4 และ FP8 ในวันเปิดตัว

ชิปใหม่ พร้อมก่อนวางขาย

- MI450 หรือ gfx1250 รองรับใน vLLM upstream แล้ว ผ่าน PR #46516
- สาธิต MI450 ในงาน AMD Advancing AI ด้วย vLLM upstream ไม่ใช่ fork
- ฮาร์ดแวร์รุ่นถัดไปกำลังทดสอบบน vLLM ในระบบจำลอง
- ลูกค้าได้ฮาร์ดแวร์เมื่อไร ซอฟต์แวร์ก็พร้อมใช้แล้ว

3

หัวข้อที่ 3 จาก 7

Kernel: FlyDSL, AITER, ATOM

1 AMD ใน vLLM วันนี้

2 2026 ปีที่ย้ายไป upstream

3 **Kernel: FlyDSL, AITER, ATOM**

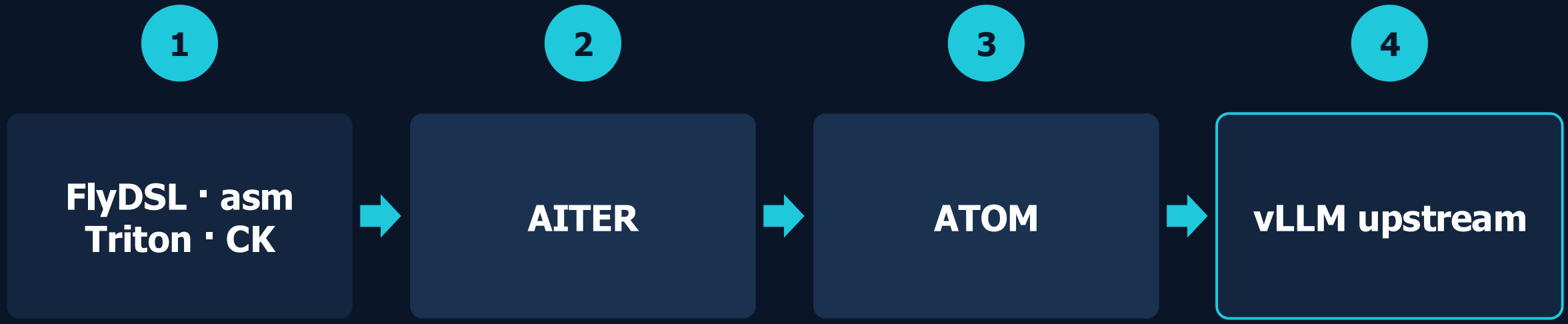
4 เครื่องมือ: Quark และ TraceLens

5 vLLM บน Navi

6 สิ่งที่กำลังจะมา

7 Repo ที่แนะนำ

เส้นทางของ kernel จาก AMD ถึงผู้ใช้



เครื่องมือเขียน kernel หลายแบบ

คลัง kernel กลาง เลือกตัวที่เร็วที่สุดให้อัตโนมัติ

สนามทดสอบความเร็ว ลองของใหม่ได้ทันที

ของที่ได้ผลถูกส่งเข้า vLLM ให้ทุกคนใช้

AITER และ MoRI

ครบทุกงาน AI

- kernel ทุกแบบ เช่น MoE, attention, collectives และ fusion
- รองรับทุก datatype เช่น FP8 และ MXFP4

ใช้ได้กับทุก framework

- ใช้ได้ทั้ง training และ inference
- เขียนครั้งเดียว หลาย framework เรียกใช้ร่วมกัน

หลาย backend

- FlyDSL, asm, Triton, CK และ HIP
- เลือก backend ที่ปรับจูนแล้วตามขนาดข้อมูล

AITER คำนวณ · MoRI-IO ส่ง KV cache

- MoRI-IO connector อยู่ใน vLLM upstream พร้อมเอกสาร unit test และอยู่ใน ROCm image
- P/D รันได้บน MI300X ถึง MI355X ผ่าน KV-connector interface มาตรฐานของ vLLM

เทียบกับฝั่ง NVIDIA

จับคู่ตามบทบาทที่ใกล้เคียงกัน ไม่ได้แปลว่าทำงานเหมือนกันทุกอย่าง

	AMD	NVIDIA
ชุดซอฟต์แวร์หลัก	**ROCm**	**CUDA**
เขียน kernel ด้วย C++	**HIP**	**CUDA C++**
เขียน kernel ด้วย Python	**FlyDSL**	**CuTe DSL**
เขียน kernel ได้ทั้งสองค่าย	**Triton**	**Triton**
ชิ้นส่วน kernel สำเร็จรูป	**Composable Kernel**	**CUTLASS**
ภาษาระดับเครื่อง	**AMD GPU assembly**	**PTX และ SASS**
คลัง kernel สำหรับ LLM	**AITER**	**FlashInfer และ cuBLAS**
สื่อสารระหว่าง GPU	**RCCL**	**NCCL**
ส่ง KV cache ข้ามเครื่อง	**MoRI-IO**	**NIXL**
inference engine ของผู้ผลิต	**ATOM**	**TensorRT-LLM**

FlyDSL: เขียน kernel ด้วย Python

DSL ของ AMD ที่มี MLIR compiler อยู่เบื้องหลัง · github.com/ROCm/FlyDSL

1 เขียนอะไร

- tiling, partitioning และการย้ายข้อมูลด้วย Python
- กำหนด register และการเข้าถึงหน่วยความจำได้ชัด
- ใช้จริงแล้วกับโมเดลส่วนใหญ่

2 รันอย่างไร

- ติด @flyc.jit แล้วแปลตอนเรียกครั้งแรก
- ไม่ต้อง build ไฟล์ .so แยกตามรุ่น GPU
- JIT เร็วกว่าเครื่องมือเดิมอย่าง CK มาก

3 ใช้ที่ไหนใน vLLM

- merge แล้ว 24 PR และรอตรวจอีก 28
- kernel MoE และ attention บน gfx942 และ gfx950
- MoE แบบ W4A16 ใน #44400 พร้อมทดสอบความแม่นยำบน MI355

ใช้ ATOM ใน vLLM โดยไม่ต้องแก้ core

1 ติดตั้ง ATOM

ลงไว้ในสภาพแวดล้อมเดียวกับ vLLM

2 vLLM ค้นเจอเอง

ผ่าน Python entry points ทั้ง ATOMPlatform และโมเดลที่ปรับแต่งแล้ว

3 สั่ง vllm serve ตามปกติ

ไม่ต้องเปลี่ยนคำสั่งเลย

vLLM ยังดูแลเหมือนเดิม

- scheduling และ batching
- KV cache และ OpenAI API

ปิดได้เพื่อเทียบแบบ A/B

`ATOM_DISABLE_VLLM_PLUGIN=1`

ทดสอบกับ vLLM v0.17.x แล้ว

4

หัวข้อที่ 4 จาก 7

เครื่องมือ: Quark และ TraceLens

1 AMD ใน vLLM วันนี้

2 2026 ปีที่ย้ายไป upstream

3 Kernel: FlyDSL, AITER, ATOM

4 **เครื่องมือ: Quark และ TraceLens**

5 vLLM บน Navi

6 สิ่งที่กำลังจะมา

7 Repo ที่แนะนำ

Quark: บีบอัดโมเดลสำหรับทุกคน

เครื่องมือ quantization แบบ open source ของ AMD สำหรับ weight, activation และ KV cache

ทำอะไรได้

- มี AWQ, GPTQ, rotation และ SmoothQuant ในตัว
- ใช้กับโมเดล PyTorch และ ONNX
- รองรับฮาร์ดแวร์หลายแบบ

หาได้ที่ไหน

- pip install amd-quark
- github.com/amd/quark
- checkpoint บน Hugging Face ใต้บัญชี amd

พร้อมวันเปิดตัว

- PTPC-FP8, INT4-FP8, MXFP4 และ NVFP4
- DeepSeek V3/V4, GPT-OSS, GLM 5.x, Qwen 3.x และ Kimi K2/K3

vLLM อ่านค่า `quantization="quark"` ในไฟล์ตั้งค่าของโมเดล แล้วเลือก backend ให้เอง ไม่ต้องเพิ่ม flag

สิ่งที่ Quark ส่งเข้า vLLM

35+ PR ที่ merge

100+ PR ที่ช่วยตรวจ

ในไตรมาสล่าสุด พร้อมงาน CI และประชุม SIG ทุกสัปดาห์

MX microscaling ใน upstream

- MXFP4, MXFP6 และ MXFP8 อยู่ใน vLLM แล้ว
- kernel abstraction ชุดเดียว เปลี่ยน backend ได้
- ไม่ต้องเพิ่ม flag ในคำสั่ง

NVFP4 บนชิปที่ไม่รองรับในตัว

- emulation kernel ด้วย Triton ทั้ง dense และ MoE
- รันได้บน MI300 และ MI355X
- Quark สร้าง checkpoint แบบ NVFP4 ได้แล้ว

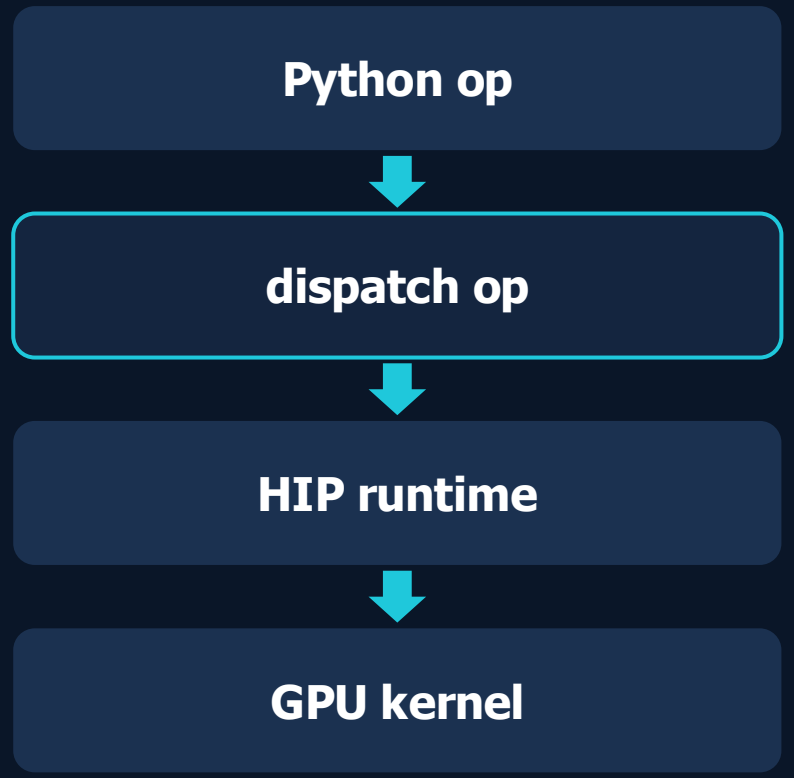
KV cache 4 bit ด้วย TurboQuant

- รองรับ ROCm พร้อม kernel ที่ปรับแต่งแล้ว
- มี 3 ทาง: Triton, HIP และ FlyDSL
- วัดผลกับ MiniMax M2.5 แล้ว Gemma 4 และ GLM-5.2 รอตรวจ

TraceLens: ดูว่าเวลาหมดไปกับอะไร

เครื่องมือวิเคราะห์ trace แบบ open source · github.com/AMD-AGI/TraceLens

call tree 4 ชั้นที่ Trace2Tree สร้างขึ้น



ชื่อ kernel อย่างเดียวไม่พอ

ชื่อเดียวครอบคลุมหลายขนาดข้อมูล และ GEMM เดียวกันมีชื่อต่างกันบน Instinct กับฮาร์ดแวร์อื่น จึงยึด dispatch op เป็นหลักในการเทียบ

สิ่งที่ได้

TFLOP/s และ TB/s ที่ทำได้จริงต่อ operation โยงเวลาถึงระดับ nn.Module มี roofline การเทียบ trace และการ replay ปัญหาแบบย่อส่วน

ใช้กับใคร

อ่าน PyTorch profiler trace ได้ทุกฮาร์ดแวร์ แยกเวลารอกันออกจากเวลาสื่อสารจริงบนหลาย GPU ใช้ผ่าน Python SDK หรือ UI

NIXL: ผู้ผลิตสองรายใน repo เดียว

ก่อน

- AMD ดูแล fork ของ NIXL เองในชื่อ RIXL
- ต้องตามแก้ให้ทันต้นฉบับอยู่ตลอด

ตอนนี้

- NIXL connector รับ PR จากผู้ผลิตรายอื่นแล้ว
- รองรับ AMD บน gfx942 และ gfx950
- AMD ปิด RIXL และใช้ NIXL ร่วมกัน

open source ที่เปิดจริง และน่าจะมีเรื่องแบบนี้ตามมาอีก

5

หัวข้อที่ 5 จาก 7

vLLM บน Navi

1 AMD ใน vLLM วันนี้

2 2026 ปีที่ย้ายไป upstream

3 Kernel: FlyDSL, AITER, ATOM

4 เครื่องมือ: Quark และ TraceLens

5 **vLLM บน Navi**

6 สิ่งที่กำลังจะมา

7 Repo ที่แนะนำ

vLLM บน Navi: ตั้งแต่โน้ตบุ๊กถึงเซิร์ฟเวอร์

โน้ตบุ๊กและ mini PC

- Ryzen AI แบบ APU มี Navi ในตัว
- Ryzen AI Halo มี unified memory ถึง 128 GB

เครื่องตั้งโต๊ะและ workstation

- Radeon 7000 และ 9000 รวมถึง Radeon PRO
- การ์ดใบเดียวคือที่ทดลองที่ถูกที่สุด

เซิร์ฟเวอร์องค์กร

- 1 ถึง 8 GPU ต่อ node
- ต่อหลาย node ได้เมื่อลูกค้าต้องการ

รองรับใน upstream ทันที

- RDNA3 และ RDNA4 ใช้ vLLM ต้นฉบับได้เลย
- build stable และ nightly ครอบคลุม gfx9, gfx11 และ gfx12
- รองรับ vLLM-Omni และโมเดลยอดนิยมที่พอดีหน่วยความจำ
- คนส่วนใหญ่มี Radeon จึงเป็นที่แรกที่ได้รู้จัก vLLM บน AMD

6

หัวข้อที่ 6 จาก 7

สิ่งที่กำลังจะมา

1 AMD ใน vLLM วันนี้

2 2026 ปีที่ย้ายไป upstream

3 Kernel: FlyDSL, AITER, ATOM

4 เครื่องมือ: Quark และ TraceLens

5 vLLM บน Navi

6 **สิ่งที่กำลังจะมา**

7 Repo ที่แนะนำ

ถัดไป: ติดตั้ง ROCm ผ่าน pip

เดิม: ติดตั้ง ROCm เอง หรือเริ่มจาก container

ต่อไป: `pip install vllm` **แล้วได้ ROCm, PyTorch, Triton และ AITER ครบ**

ไม่ต้องใช้ root

- runtime ลงใน site-packages
- ไม่ต้องใช้ apt หรือ container

ล็อกเวอร์ชันได้

- rocm-sdk อยู่ใน requirements
- หลายเวอร์ชันใน virtualenv แยกกัน
- lockfile บันทึกทั้ง stack

ลองและแก้ได้เร็วขึ้น

- TheRock ออกเวอร์ชันใหม่ทุก 6 สัปดาห์
- CI ทดสอบเวอร์ชันใหม่ได้วันที่ออก
- ไล่หา regression ได้ง่าย

ก้าวแรก: vLLM PR #49925 นำ TheRock เข้า container บน Python 3.12 และ Ubuntu 22.04 · งานที่เหลือติดตามใน Issue #48826

Mooncake และ batch invariance

ส่งแล้ว

- HIP transport สำหรับ GPU AMD ตั้งแต่ ธ.ค. 2025
- AMD เป็น code owner ของ hip_transport

merge แล้ว

- wheel และ CI ของ ROCm ใน Mooncake #3184 เมื่อ 3 ส.ค.
- ROCm เป็นการทดสอบบังคับของ Mooncake

ยังเปิด

- ปล่อย ROCm wheel
- เพิ่ม Mooncake ใน ROCm image ของ vLLM ใน PR #52650

วันนี้ P/D บน AMD ใช้ MoRI-IO และ Mooncake จะเป็นอีกทางเลือก

Batch invariance

- ผลลัพธ์เหมือนเดิมทุกครั้ง ไม่ว่าคำขอจะถูกรวม batch แบบไหน
- ผู้ใช้ขอกันมานาน
- PR ของ Matt Wong ใกล้ merge
- จะขยายไปโมเดลอื่นต่อเนื่อง

7

หัวข้อที่ 7 จาก 7

Repo ที่แนะนำ

1 AMD ใน vLLM วันนี้

2 2026 ปีที่ย้ายไป upstream

3 Kernel: FlyDSL, AITER, ATOM

4 เครื่องมือ: Quark และ TraceLens

5 vLLM บน Navi

6 สิ่งที่กำลังจะมา

7 **Repo ที่แนะนำ**

Repo ที่แนะนำ

จุดเริ่มต้นสำหรับคนที่อยากลองใช้หรือร่วมพัฒนา vLLM บน AMD

vllm-project/vllm

vLLM ต้นฉบับ backend ของ ROCm อยู่ในนี้

ROCm/aiter

คลัง kernel ความเร็วสูงของ AMD

ROCm/ATOM

สนามทดสอบความเร็ว ใช้เป็น plugin ของ vLLM

ROCm/FlyDSL

เขียน kernel สำหรับ GPU AMD ด้วย Python

amd/quark

บีบอัดโมเดลด้วย quantization

AMD-AGI/TraceLens

วิเคราะห์ trace หาจุดที่ช้า

ROCm/TheRock

build ROCm และติดตั้งผ่าน pip

rocm.blogs.amd.com

บทความเจาะลึกของทีม ROCm

repo ทั้งหมดอยู่บน github.com ยกเว้นบล็อกที่เป็นเว็บไซต์

คุยกันบน Slack ของ vLLM

#sig-amd

slack.vllm.ai

ห้องที่นักพัฒนาฝั่ง AMD ของ vLLM ใช้คุยกัน ถามปัญหาหรือเสนอไอเดียได้

ขอบคุณครับ

ถาม-ตอบ

