



RL environment for LLM decode megakernels



Name: Sidhaarth Sredharan Murali
Number: +1 (248) 835 8033
Other details: Ask/Text me

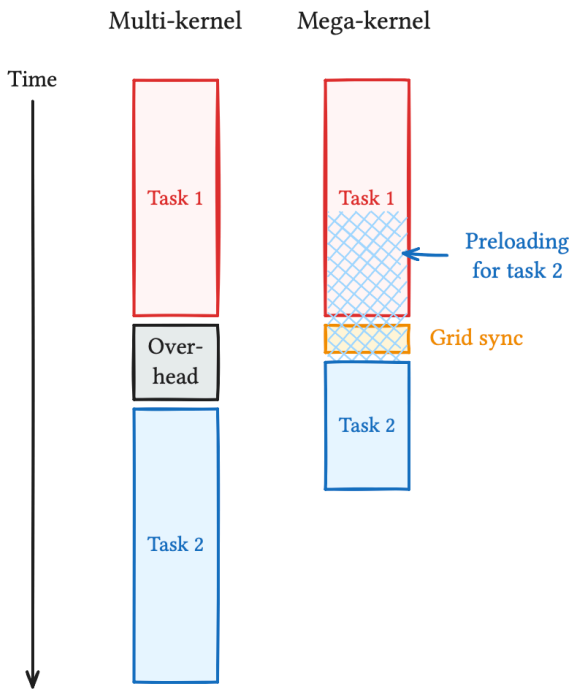
Teaching a frontier model to fuse an entire decode step into one CUDA kernel in a H100

01 Problem

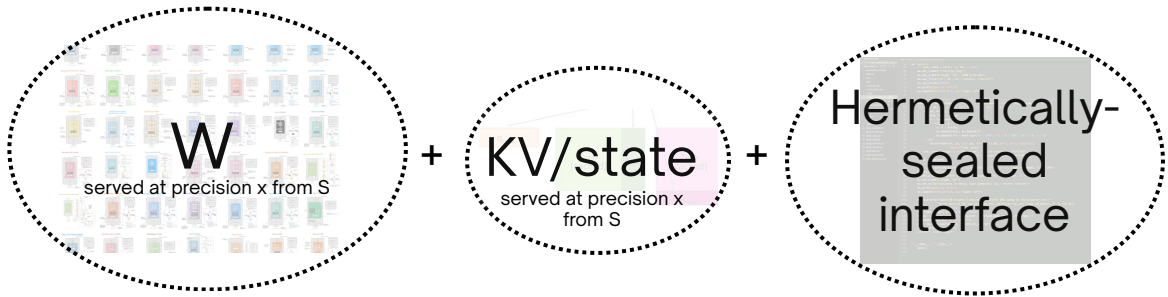
A megakernel fuses a **whole model** into one GPU launch. Making one fast means solving two problems at once:

1. **scheduling** (thousands of ops inside a single launch, kept from stalling by counters instead of global barriers) and
2. **kernel engineering** (tiling, warps, memory staging).

However, a reward that grades both together is noisy and easy to game.



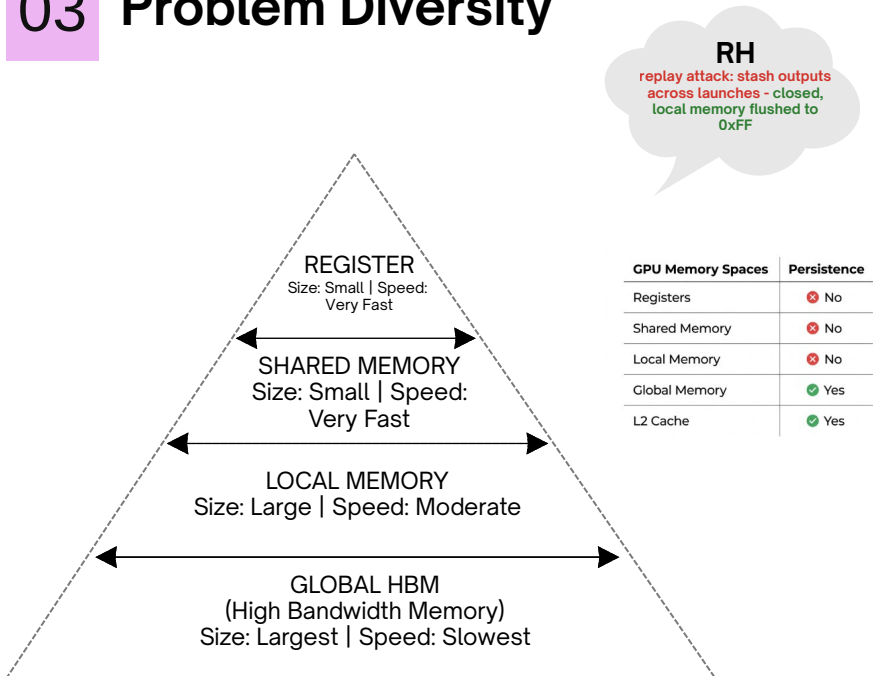
02 What does the student see?



```
extern "C" __global__ void mk_main(  
    const void *hidden_in, const void *ln_w, const void  
    *q_proj_w, ...  
    const void *recurrent_state, const void *conv_state_q,  
    ...  
    void *hidden_out, void *state_new, ... int N, int B,  
    float eps);
```

$$S = \{BF16, FP8, INT8\}$$

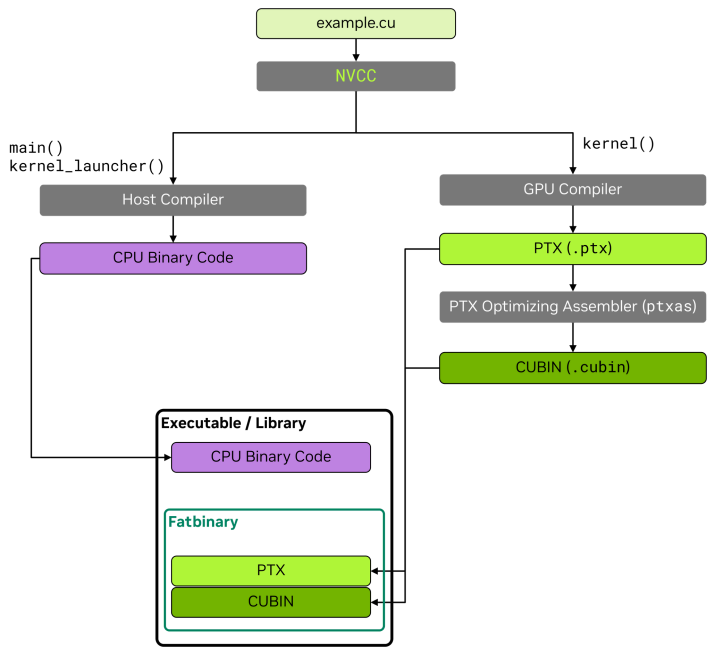
03 Problem Diversity



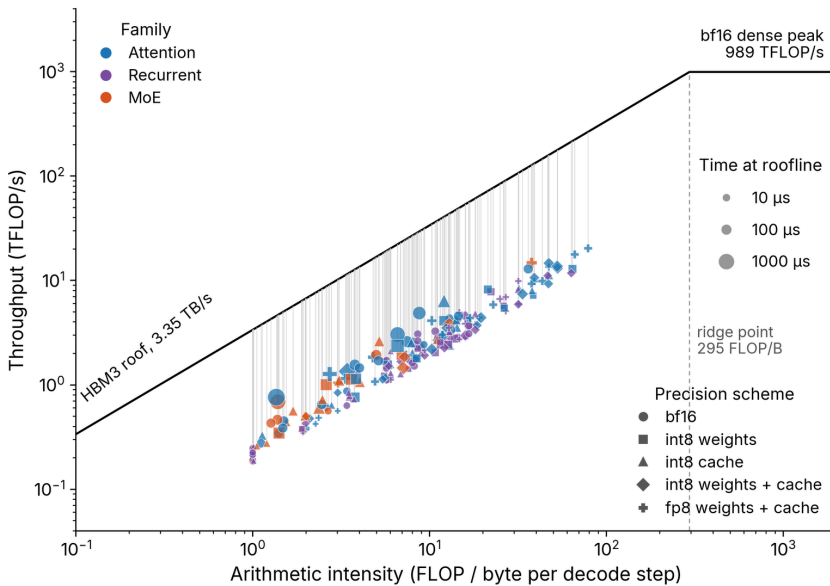
04 Grading

No Host Code. No Host Attack Surface

1. Only the isolated GPU kernel (device code) is extracted and compiled.
2. Since a single-launch kernel cannot access files, spawn processes, or modify scores, host-side attacks are eliminated by construction

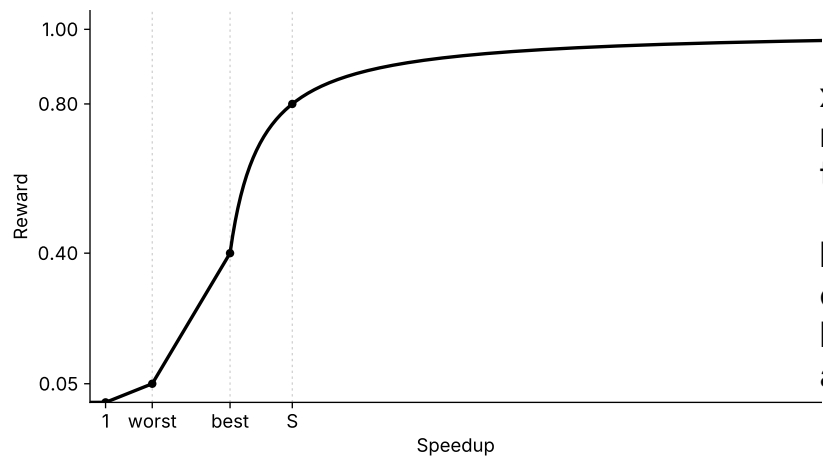


05 Where are the baseline SOTAs at?



torch.compile baselines sit 6–18× below the H100 roofline.

06 Reward Function



x = median of the per-kernel ratios of student kernel time to baseline kernel time.

Median of ratios is preferred over **ratio of medians**, as it better cancels out hardware and sandbox jitter.

$$\begin{aligned} x \leq 1 & \quad 0 \\ 1 < x < \text{worst} & \quad 0.05 \cdot (x-1) / (\text{worst}-1) \\ \text{worst} \leq x < \text{best} & \quad 0.05 + 0.35 \cdot (x-\text{worst}) / (\text{best}-\text{worst}) \\ x \geq \text{best} & \quad 0.4 + 1.2 \cdot (x-\text{best}) / ((S-\text{best}) + 2(x-\text{best})) \end{aligned}$$