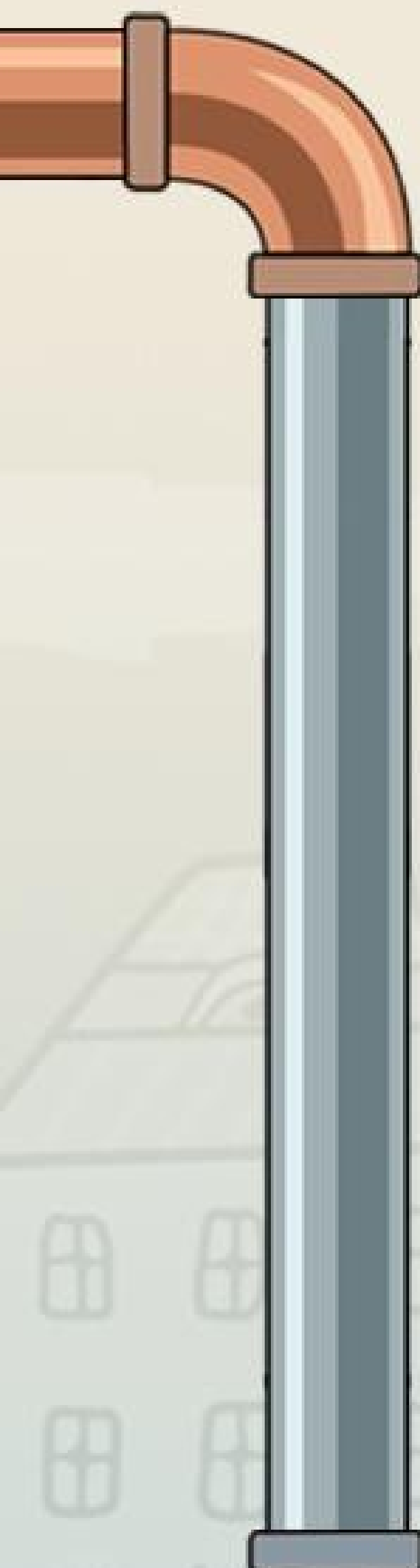




Closing the GPU Locality Feedback Loop

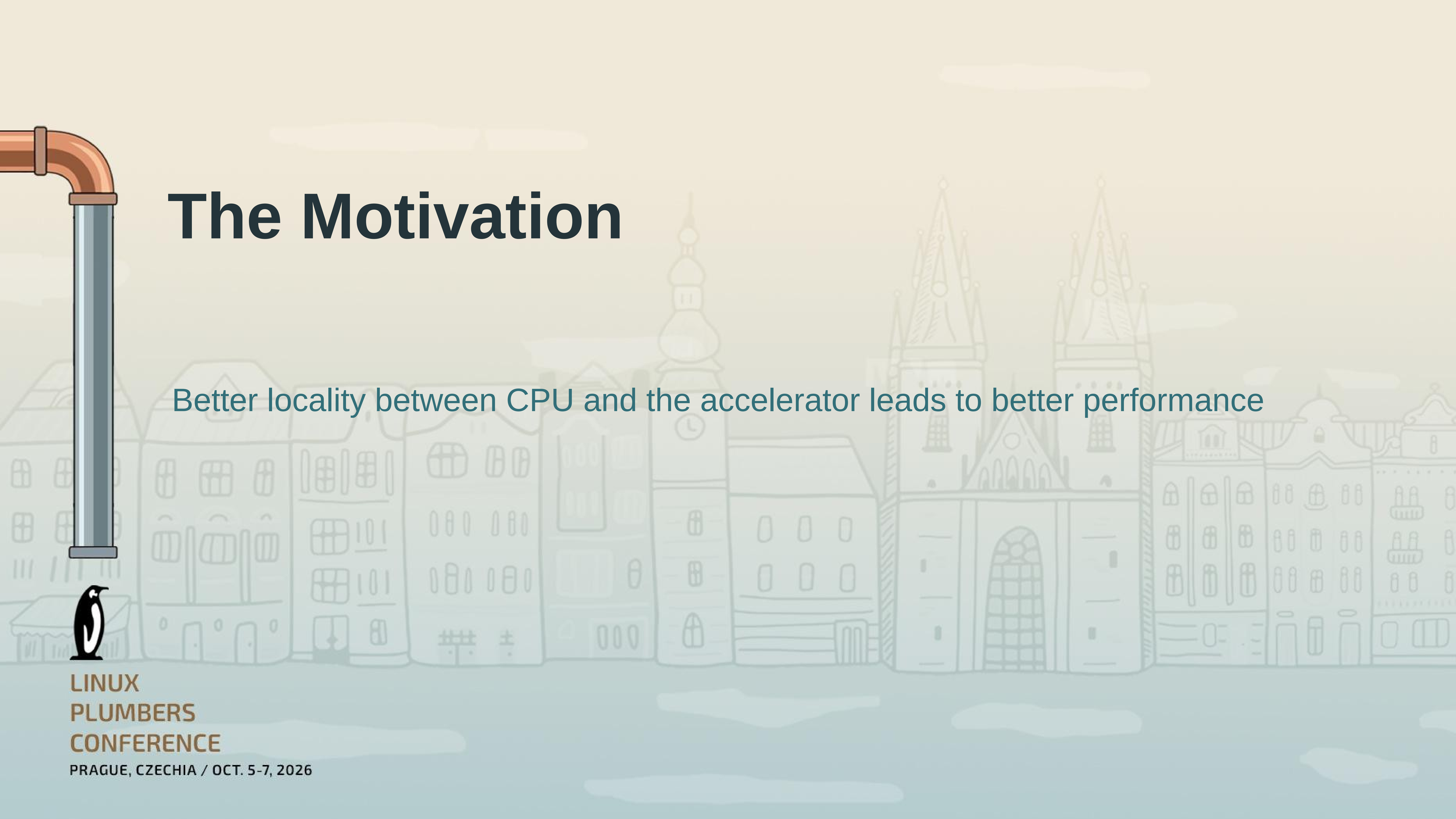
SCX placement + per-task BPF telemetry + application-owned page migration

Balbir Singh, Lee Trager, NVIDIA | September 2026



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



The Motivation

Better locality between CPU and the accelerator leads to better performance



**LINUX
PLUMBERS
CONFERENCE**

PRAGUE, CZECHIA / OCT. 5-7, 2026

SCX_cosmos

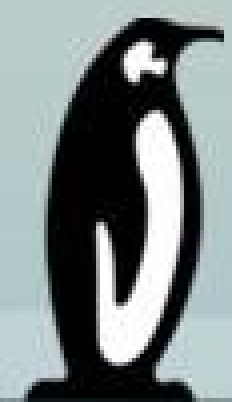
Benefits

- Uses NVML observations to place GPU workloads on CPUs near GPUs they use

Limitations

- Depends on NVIDIA/NVML-specific userspace polling
- No visibility into the workload's host-memory placement

Previous discussion at OSPM 2026 Ref:
<https://lwn.net/Articles/1077759/>



**LINUX
PLUMBERS CONFERENCE** PRAGUE, CZECHIA / OCT. 5-7, 2026



The Feedback Loop

Six observable stages; one bounded action

01 Accelerator call

device 0

02 BPF hint

node 0

03 SCX

CPU node 0 Or numa
balancing

04 Telemetry

remote access (optional)

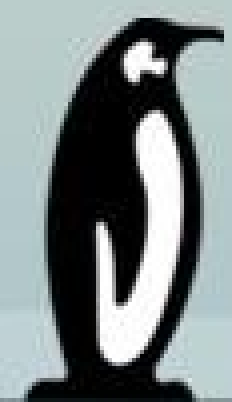
05 Workload

move_pages (optional)

06 Feedback

bandwidth + event

BPF recommends; userspace acts or kernel space acts. Address lifetime ownership stays with the workload.



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



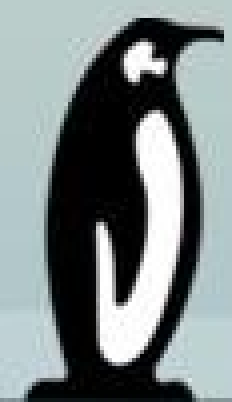
Kernel Improvements

BPF-driven NUMA Balancing

- Enable NUMA access tracking for sched_ext tasks
- Allow sched_ext to express a preferred memory node and influence the existing page-migration decision

Device Process Association

- Allow devices to associate with a process(es)
- Data provided to sched_ext to allow schedulers to make intelligent placement decisions



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Target Topology and Experimental Setup

GPU 0 is local to CPU node 0; memory begins on CPU node 1

CPU node 0 · target

GPU 0 / GPU 1

SCX target when the task feeds device 0.

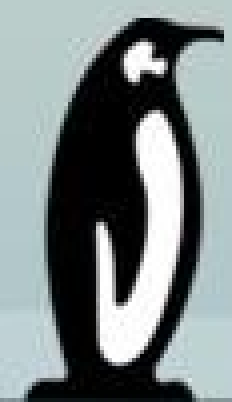


CPU node 1 · initial memory

GPU 2 / GPU 3

512 MiB allocated and faulted here.

sched_ext moves the task; page migration is a separate, evidence-driven decision.



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



The Result in One Slide

GPU topology becomes a measured placement decision

+67.0%

GPU host-memory bandwidth

−99.96%

REMOTE_ACCESS event

Three bounded trials on a two-node, four-GPU host; every page verified after `move_pages()`.



LINUX
PLUMBERS CONFERENCE

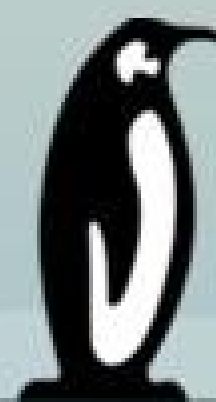
PRAGUE, CZECHIA / OCT. 5-7, 2026



Technology: Deliberately Small Pieces

Separate owners; PMU access and security policy need review

Component	Implementation	Signal / API	Responsibility
GPU program producer	C++ LD_PRELOAD	Set Device, Launch program on accelerator	Publishes one NUMA node per thread
Scheduler	SCX BPF + Rust	Task-storage hint	Prefers an allowed idle CPU on the GPU node
Telemetry	BPF + Rust	6 pinned core PMU events*	Per-task/TGID totals and recent deltas
Decision	CUDA C++ workload	REMOTE_ACCESS / instructions	Complete interval and calibrated threshold
Action	Linux syscall	move_pages(MPOL_MF_MOVE)	Ordinary, unregistered pages only
Outcome	CPU loop + CUDA events	Relative bandwidth gain	Event collapses; CPU and GPU paths speed up



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



What the Result Means

A useful division of responsibility

Scheduler / SCX

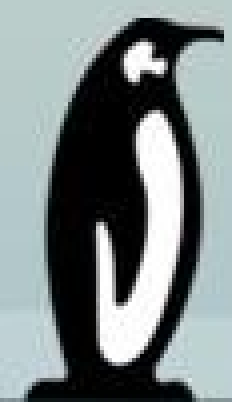
Reads the per-task GPU hint at scheduling callbacks. Chooses the GPU-local CPU node without knowing virtual addresses.

Application

Knows allocation type, page range, reuse, and whether migration is legal. Owns the address-aware action.

Evidence

Bandwidth verifies the benefit. Incomplete coverage or a low normalized signal means no move.



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

