

LPC 2026 - sched_ext MC

Boro: Do We Need More AI-Assisted Kernel Tooling?



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

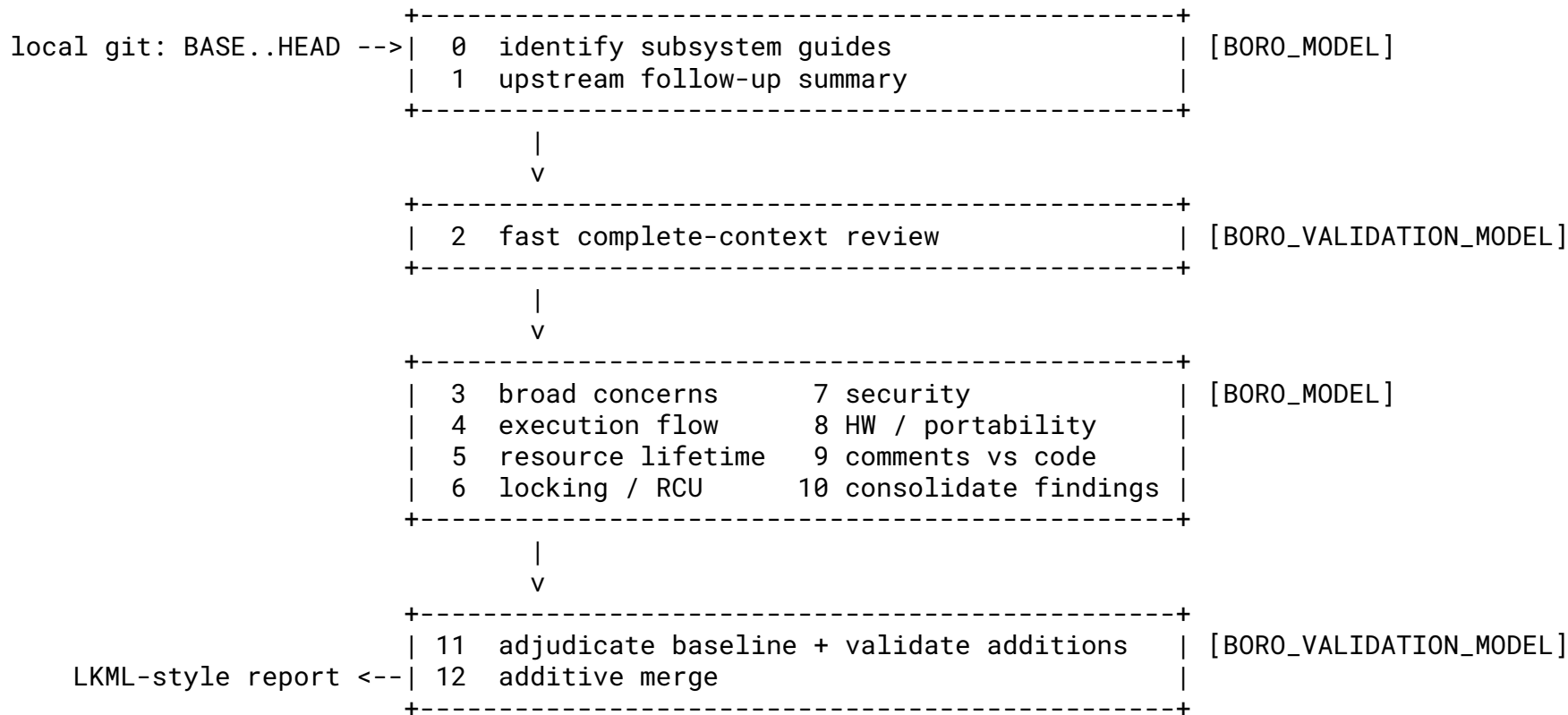
Andrea Righi <arighi@nvidia.com>

Boro

- Git repo: <https://github.com/NVIDIA/boro>
- Downstream/distro kernel dev features
 - Validate patches backported to older kernels (detect upstream follow-ups)
 - Build/boot/test plan (virtme-ng integration)
 - Detect out-of-tree module breakage (TODO)

```
$ boro review BASE..HEAD  
$ boro build  BASE..HEAD  
$ boro test   BASE..HEAD
```

Multi-stage review pipeline



Proposal: fold the useful pieces into Sashiko

- Downstream/backport model
 - Upstream follow-up tracking
- Multi-model pipeline
 - Cheap/local discovery + strong validation
 - Run stages with different models + validate disagreements
- virtme-ng integration
 - Build, boot, generate & run a target test

Questions for the room

- Which Boro features would be useful enough to add to Sashiko?
- Should upstream patch submissions be expected to include local AI-assisted review / test report results?

Thank you

Andrea Righi <arighi@nvidia.com>

Review / validation split in Sashiko (WIP)

```
[ai]
```

```
model = "gemma-4-31b-it" # 31B parameters
```

```
[ai.validation]
```

```
model = "nemotron-3-super-v3" # 120B parameters
```

Metric	Nemotron Super	Gemma4 + Super
Detected	23	19
Partially detected	2	5
Missed	72	73
Not reviewed	2	2
Avg review time	6.08 min	4.50 min
Expensive-model input	49,077,375	8,548,763
Expensive-model output	10,804,621	1,981,461
Cheap-model input	0	22,174,456
Cheap-model output	0	778,697
Expensive-token savings	-	82.4%
Avg review time savings	-	25.9%