

Supporting workgroups where there aren't any

The creation of `nir_lower_workgroup_size`

Timur Kristóf

2026



VALVE

Linux Open-Source
Graphics Drivers Group

Table of Contents

1. What's a workgroup? Why does its size matter?
2. What can we do about it?
3. Results



What's a workgroup?
Why does its size matter?

What's a workgroup?

VALVE

What's a subgroup?

VALVE

- SIMD group of shader invocations
- aka. "wave" or "warp"
- 1 invocation = 1 SIMD lane
- May cooperate through subgroup operations

What's a workgroup?

VALVE

- Multiple subgroups dispatched together
- May cooperate through shared memory (ie. LDS)
- Workgroup size specified in API limits

What's the problem?

VALVE

- Not all HW supports workgroups
- Not all HW supports large enough workgroups
- Not all apps respect API limits

Who needs this?

VALVE

- NVK: mesh shaders
- RADV: compute shaders on old HW
- V3DV: apps disrespect limits



What can we do about it?

Changing the workgroup size

VALVE

Easy.

Just repeat the same work in all subgroups.

Let's call them "logical subgroups".

Copy-paste shader code and rewrite invocation id, subgroup id, etc.

Changing the workgroup size

Silver bullets:

- `nir_cf_extract`
- `nir_cf_list_clone`
- `nir_cf_reinsert`

Changing the workgroup size

VALVE

Easy.

Just repeat the same work in all subgroups.

Let's call them "logical subgroups".

Copy-paste shader code and rewrite invocation id, subgroup id, etc.

Wait a sec... it can't be that simple, can it?

What if the shader has barriers?

What's wrong with barriers?

VALVE

- Barriers wait for work to be done in all subgroups
- We can't just copy-paste all the shader
- Is the plan ruined?

Dealing with barriers

VALVE

Each CF subtree that contains a barrier needs to be augmented to handle logical subgroups.

The "copy-paste" happens within each CF node, split by barriers.

Helper variables

Make sure everything executes under the correct conditions.

- Predicate
- Participation in current loop
- Participation in current loop iteration

CF nodes - block

- Repeat and predicate all instructions before barrier
- Process barrier
- Repeat and predicate all instructions before next barrier
- Rinse and repeat

CF nodes - if

- Determine which logical subgroups take "then" branch
- Augment the "then" CF subtree
- Move "else" branch into separate "if"
- Determine which logical subgroups take "else" branch
- Augment the "else" CF subtree

CF nodes - loop

- Keep track of which logical subgroups participate in the loop and the current iteration
- Augment the CF subtree
- Special lowering needed for "break" and "continue"

NIR pain points

CF lowering passes are hard.

- Go out of SSA and repair at the end
- Tricky to iterate over CF tree while changing it
- Automatic split/stitch of blocks (insert NOP)
- Remap table is clumsy (lower some instrs later)
- Excluding some CF is also clumsy
(extract-reinsert)



Eventually, it started
working!

Couple of months of review later...

NIR now supports lowering a shader to use smaller workgroups.

Limitations of the new pass

VALVE

- No independent forward progress
(logical subgroups aren't really independent)
- No support variable workgroup size
(had to drop it during review)

Who uses this pass?

VALVE

- NVK: mesh shaders unblocked
- RADV: mitigated HW bug on old GPUs
- RADV: uncovered ray tracing bugs
- V3DV: can run apps that use larger workgroups

Thanks

Questions, suggestions, discussion?

Supporting workgroups where there aren't any
Timur Kristóf

Venemo @ #dri-devel, #radeon, ...
<https://timur.hu>



Supporting workgroups where there aren't any

The creation of `nir_lower_workgroup_size`

Timur Kristóf

2026

VALVE

Linux Open-Source
Graphics Drivers Group

