Per Cgroup Swap Device Control

Speakers : YoungJun Park LG Electronics, Chris Li Google

東京 2025
LINUX
PLUMBERS CONFERENCE
TOKYO, JAPAN / DEC. 11-13, 2025

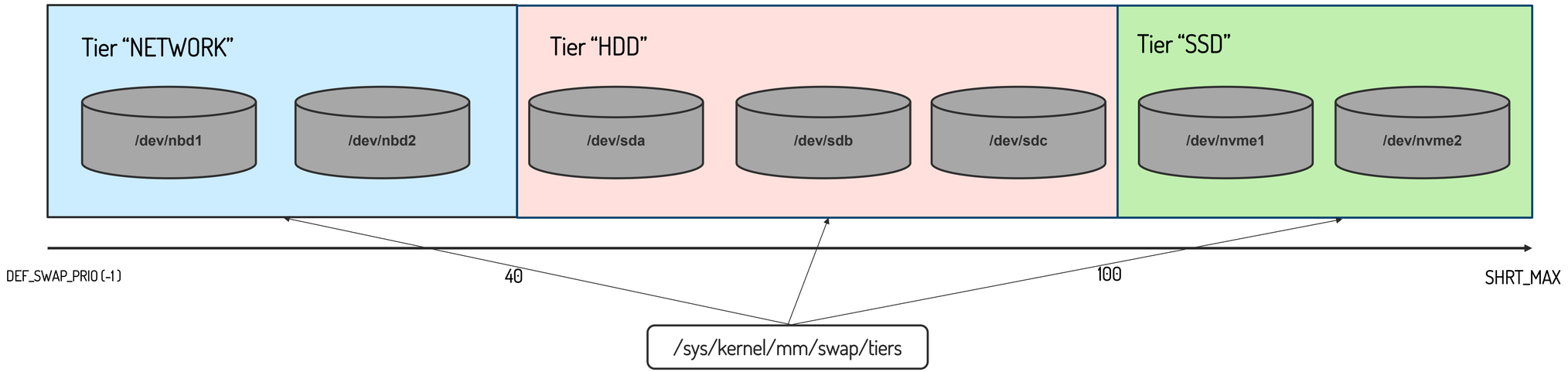
Background

- Motivation
  - Faster swap -> latency-sensitive cgroups. Slower swap -> non-critical cgroups.
  - *Linux allocate swap globally, not per cgroup*

- Initial Approach : Per-cgroup priorities
    - Over-engineering beyond practical needs
      - Our use case need just inclusion & exclusion semantic.
    - LRU inversion across cgroups caused by differing priorities
      - Pages on slow swap devices appear at the tail of LRU list

- New Direction: Swap Tiers
  - Suggested by Chris Li
  - Simpler & Extensible

Concept



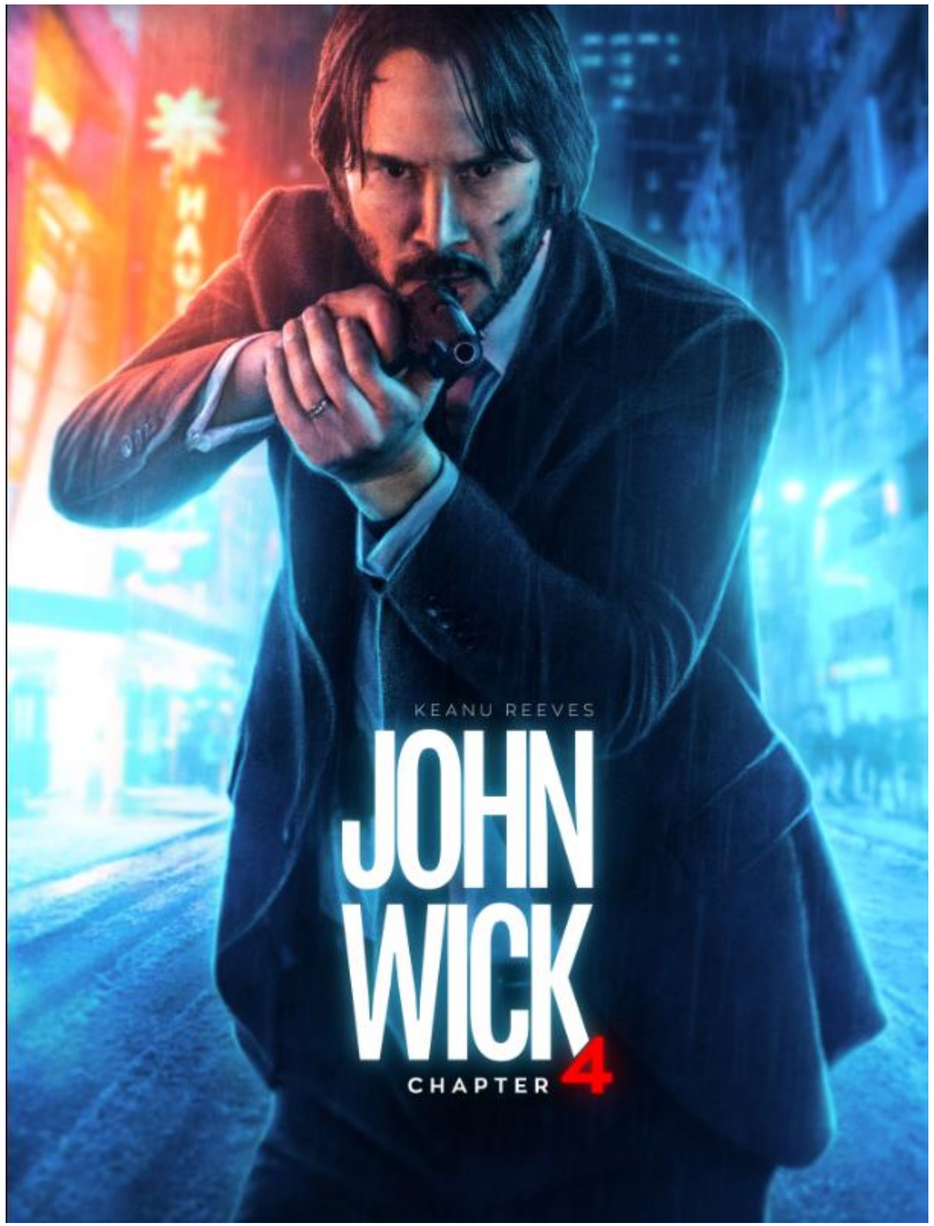
- Tier: user-named group of swap devices sharing the same priority range
- Abstraction layer facilitating swap device selection based on swap speed
  - But, it depends on the user’s freedom to select a priority.
  - Best practice: assign tiers and enable swap in order of device speed and type
- For per cgroup swap control ?
  - Make cgroup select one or more tiers.
  - cgroup only uses swap devices from allowed tiers

Rules

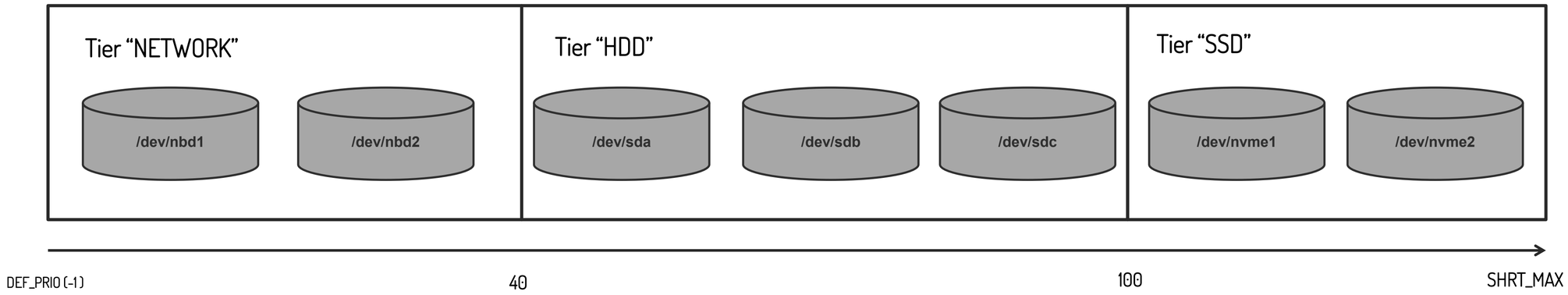
- Default tier operation: inheritance control
  - "*(empty)*": Inherit parent
  - **+**: Ignore parent, Start from all tiers
  - **-**: Ignore parent, Start from zero base

- Tier operation: per-tier control
  - **+**"*tier*": Select specific tier
  - **-**"*tier*" : Exclude specific tier

And



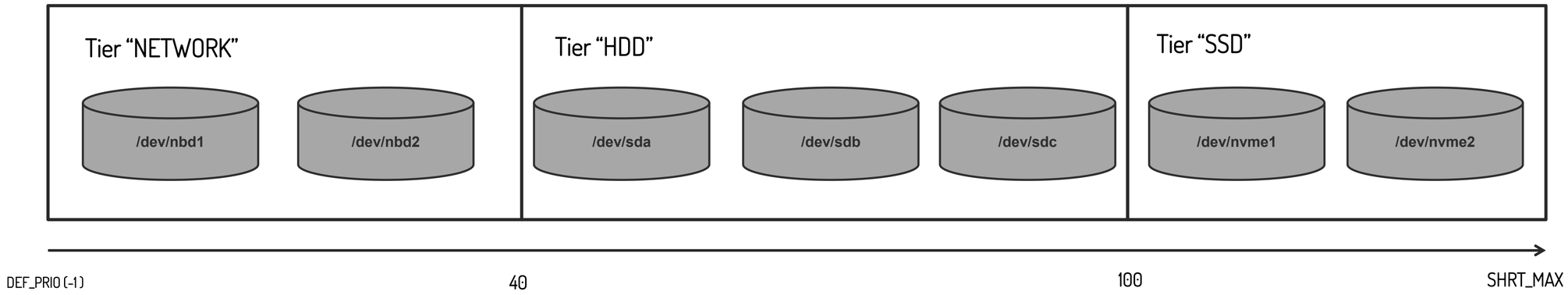
Consequences



evaulate the rule into allowed tiers

Consequences

/

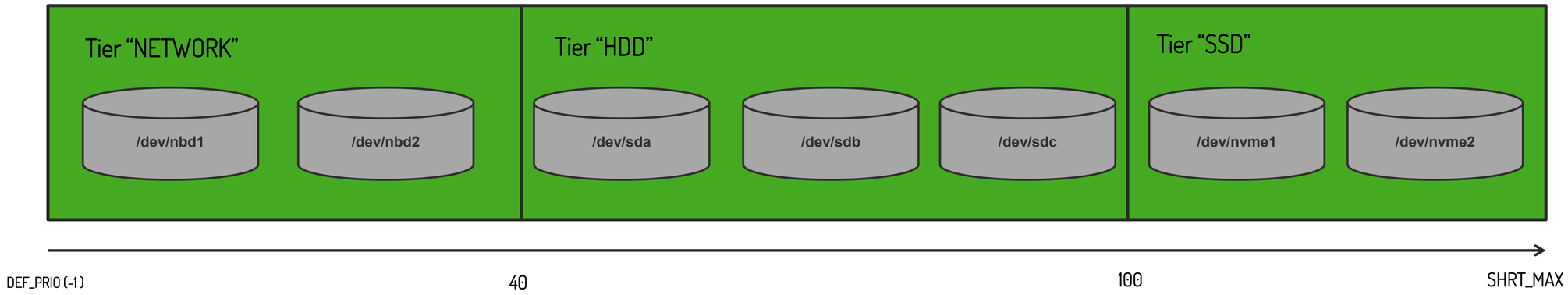


evaulate the rule into allowed tiers

/

Consequences

/

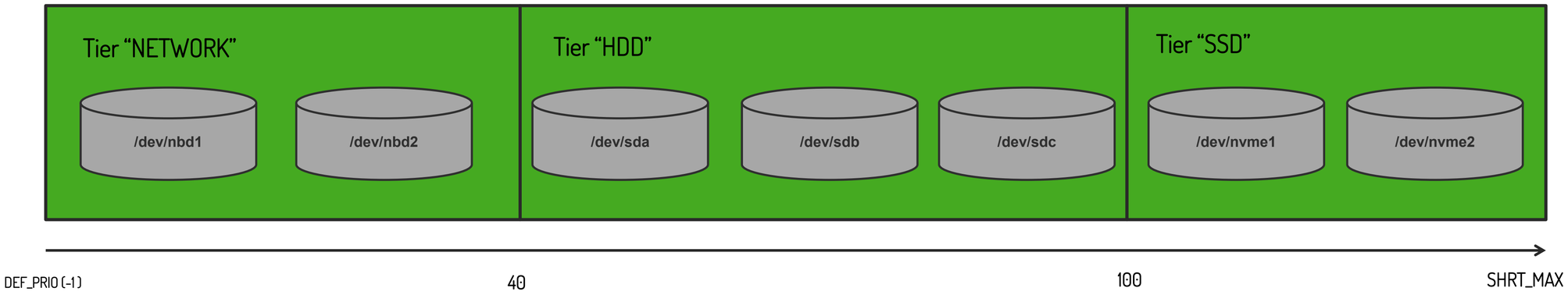


evaulate the rule into allowed tiers

/ +

Consequences

/cg1

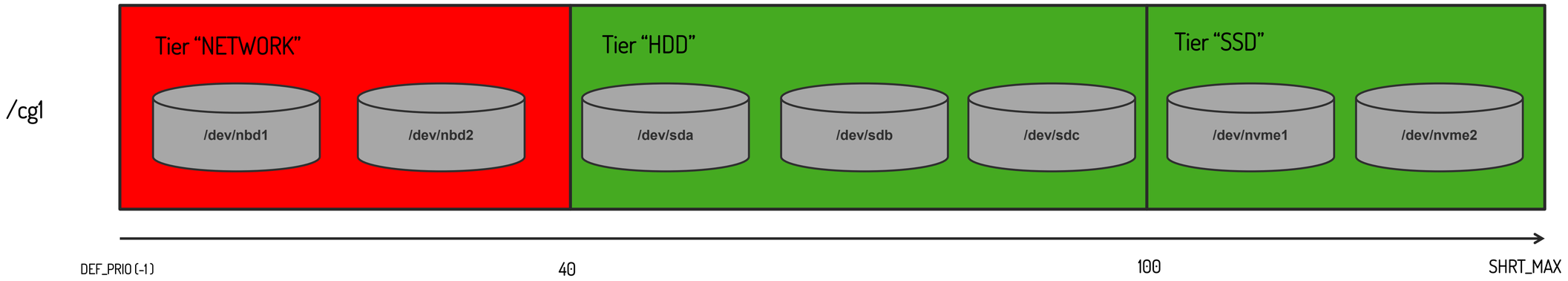


evaulate the rule into allowed tiers

/ +

/cg1

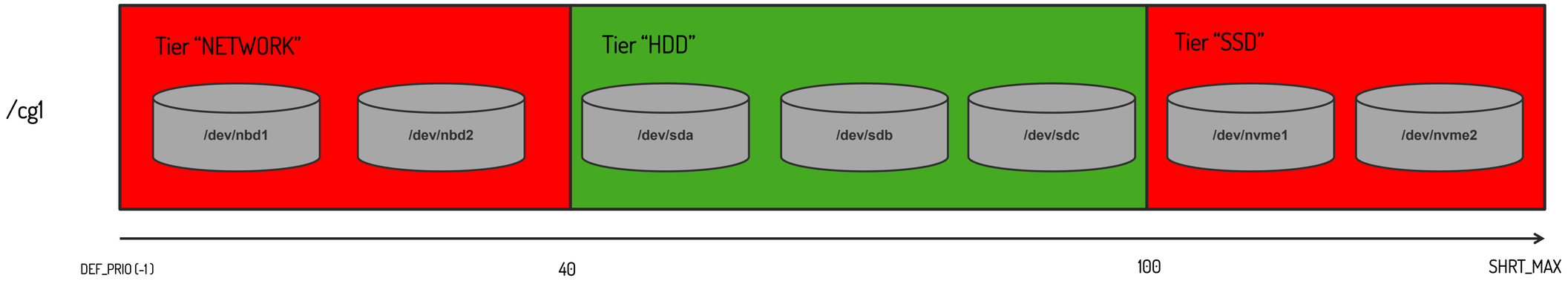
Consequences



evaulate the rule into allowed tiers

/	+
/cg1	-NETWORK

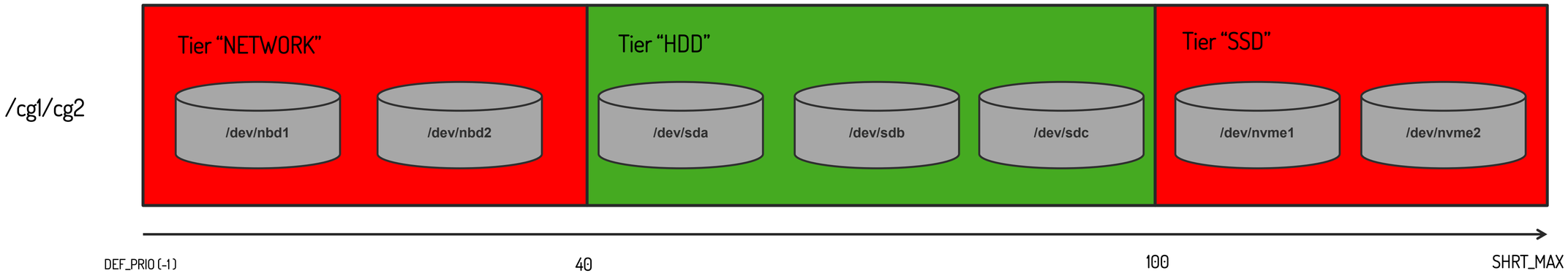
Consequences



evaulate the rule into allowed tiers

/	+	
/cg1	-NETWORK	-SSD

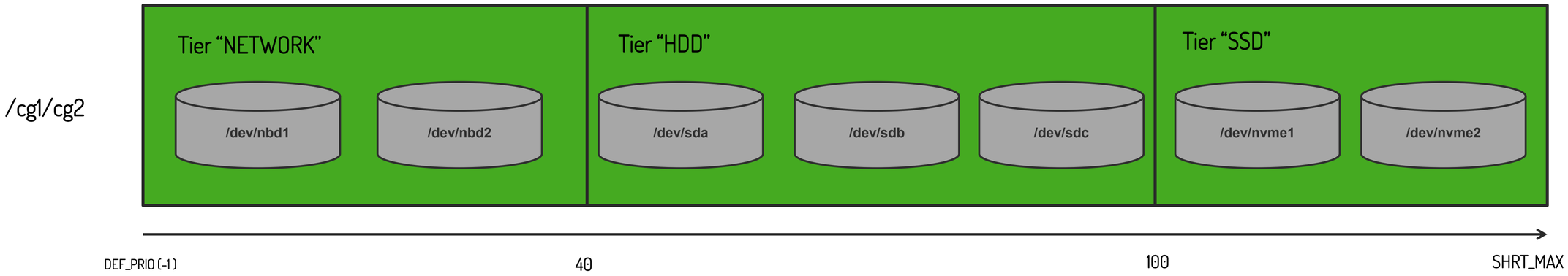
Consequences



evaulate the rule into allowed tiers

/	+	
/cg1	-NETWORK	-SSD
/cg1 /cg2		

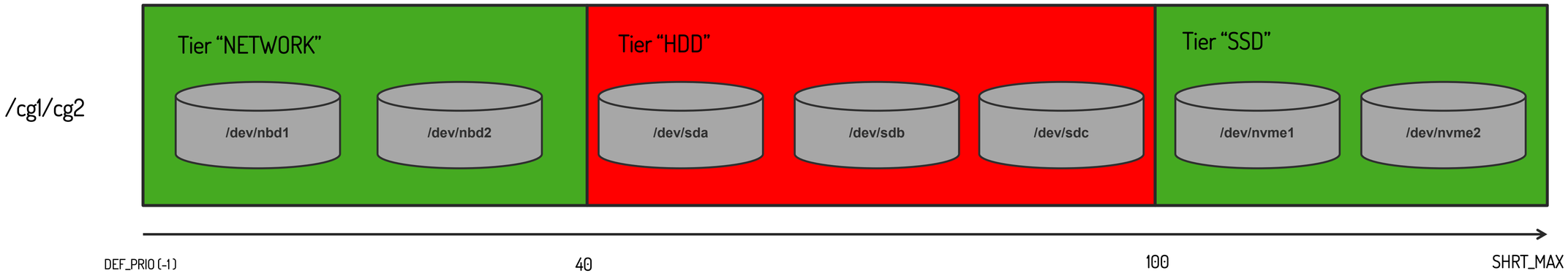
Consequences



evaulate the rule into allowed tiers

/	+
/cg1	-NETWORK -SSD
/cg1 /cg2	+

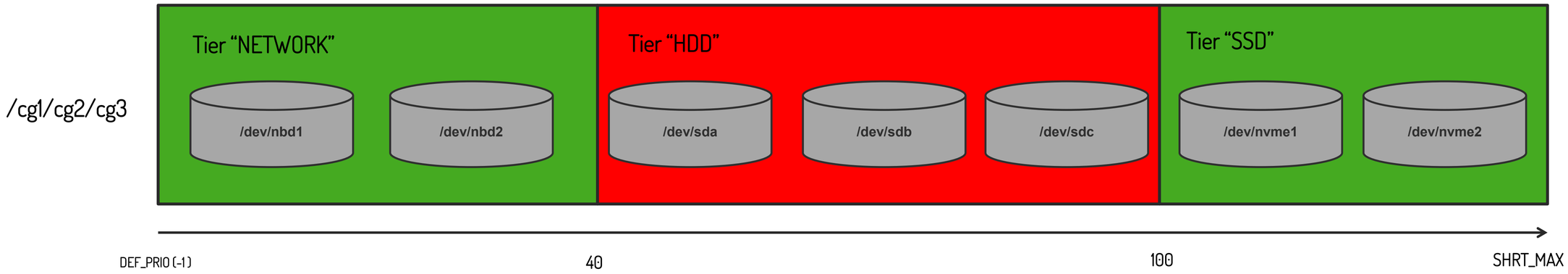
Consequences



evaulate the rule into allowed tiers

/	+	
/cg1	-NETWORK	-SSD
/cg1 /cg2	+	-HDD

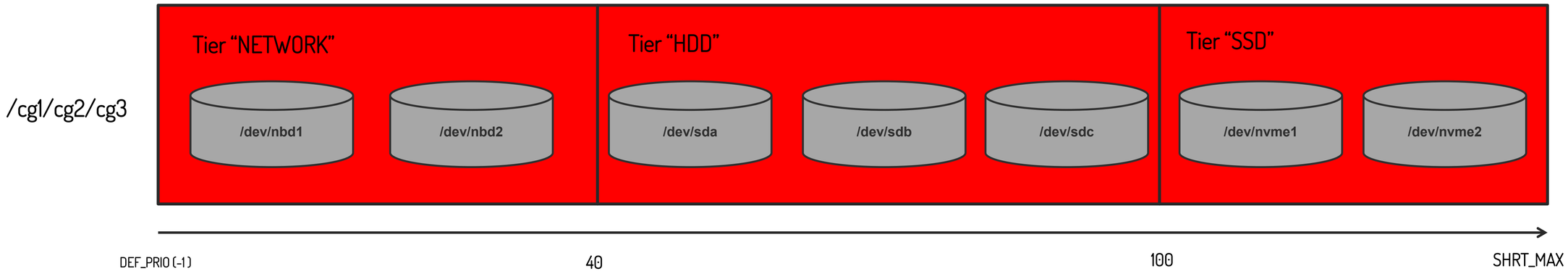
Consequences



evaulate the rule into allowed tiers

/	+	
/cg1	-NETWORK	-SSD
/cg1 /cg2	+	-HDD
/cg1 /cg2/cg3		

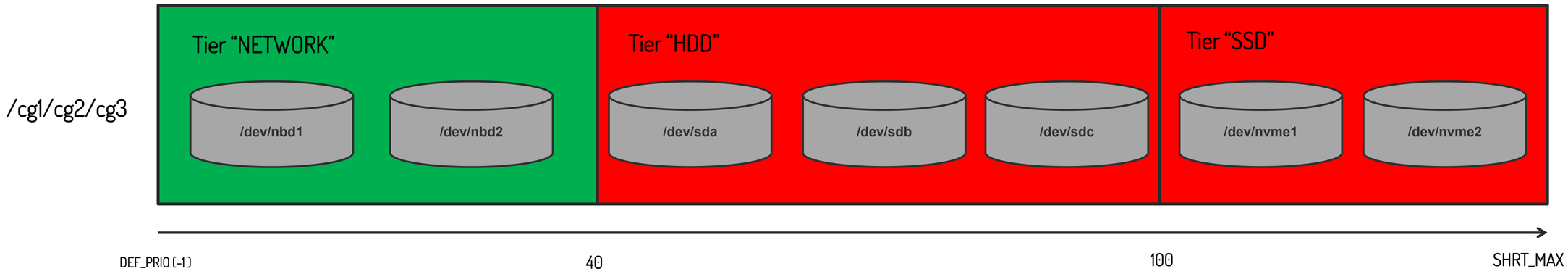
Consequences



evaulate the rule into allowed tiers

/	+
/cg1	-NETWORK -SSD
/cg1 /cg2	+ -HDD
/cg1 /cg2/cg3	-

Consequences



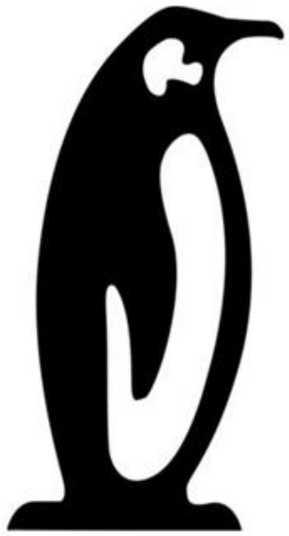
evaulate the rule into allowed tiers

/	+	
/cg1	-NETWORK	-SSD
/cg1 /cg2	+	-HDD
/cg1 /cg2/cg3	-	+NETWORK

Discussion

Discussion

- Cache Strategy: Current per-CPU cache allows tier policy violation (cgroups can access wrong-tier devices). Should we use per-tier per-CPU cache or alternative mechanism?
- Cgroup Hierarchy: Should child memcg be allowed to enable tiers disabled by parent?
- Per-Cgroup Limit: Do we need swap.tier.max in addition to swap.max?
- Tier Mask: Runtime calculation vs. pre-calculation on interface write?
- Migration: Should existing swap pages migrate when tier configuration changes?
- Future Use: Can swap tiers serve as abstraction for extended use cases?



東京 2025
LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025