



01 Recovery and elastic use cases

Applications choose membership; Revoke cancels outstanding work while retaining resources.

DP = data parallel; TP/PP = tensor/pipeline parallel. Grids simplify topology.

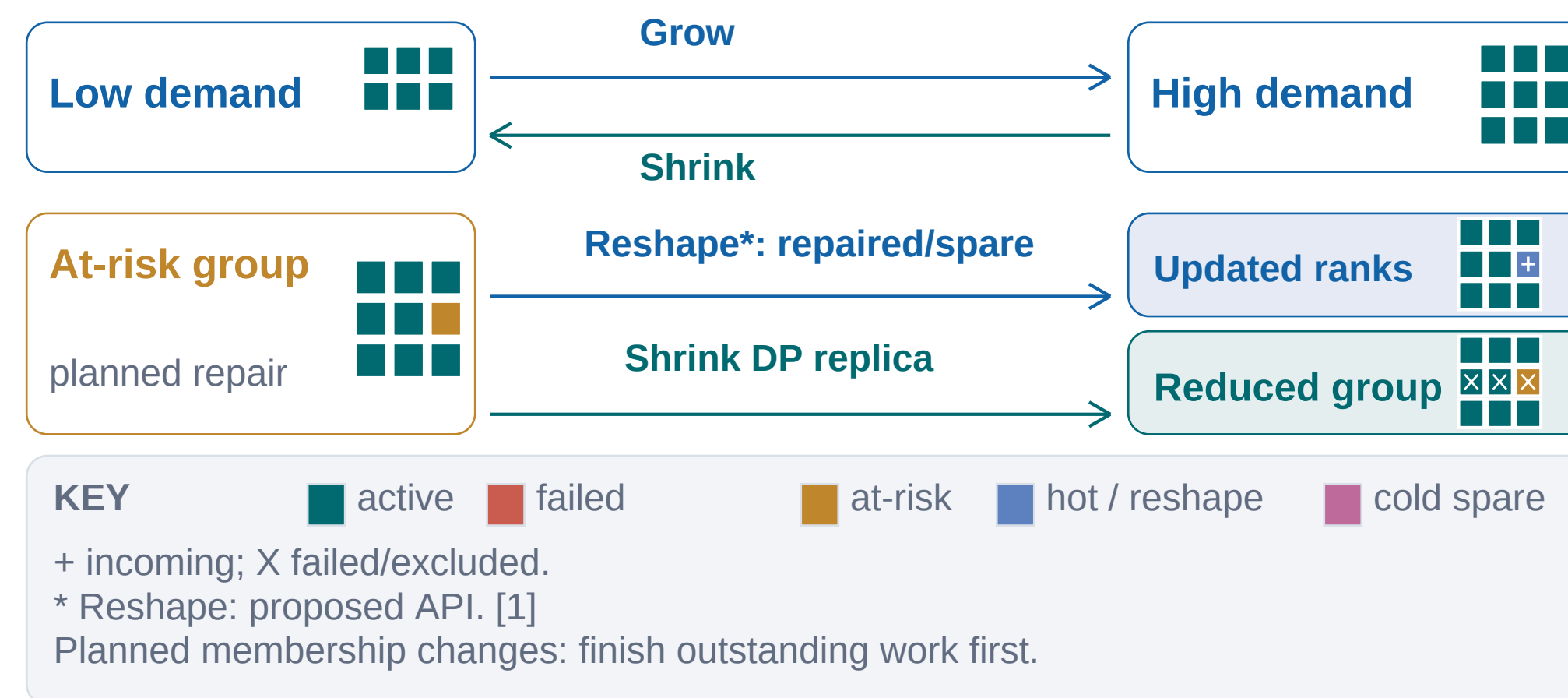


Choose a recovery path



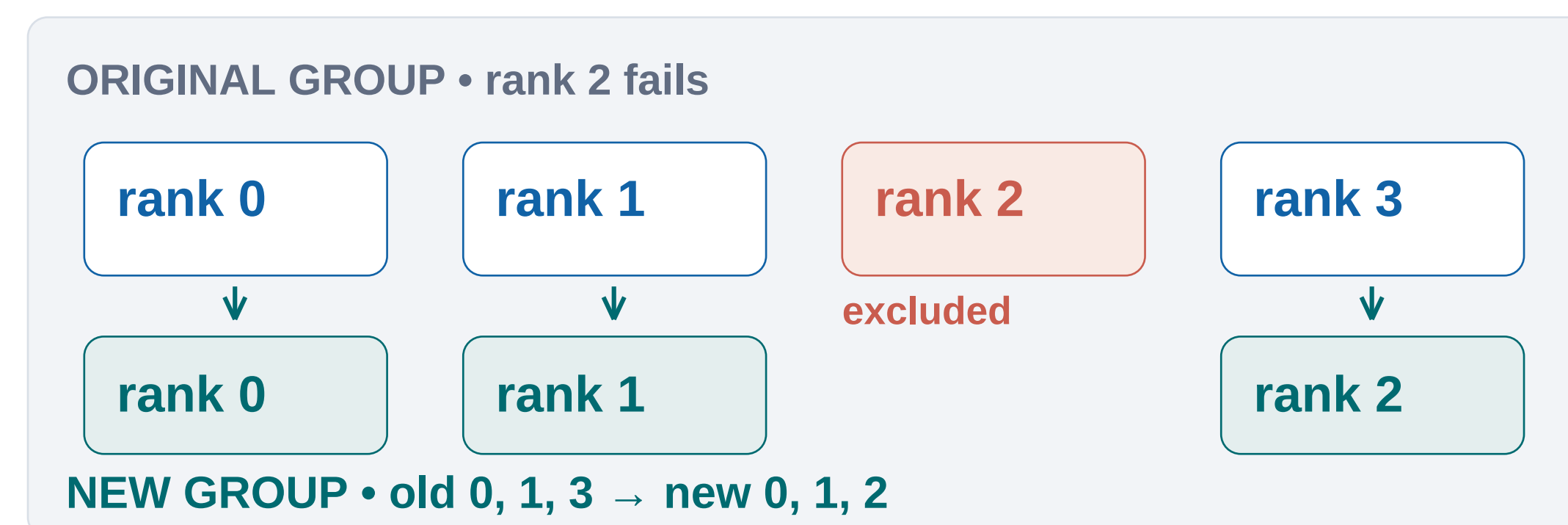
If no spare arrives, continue with 2 DP replicas.

Elastic capacity and planned maintenance



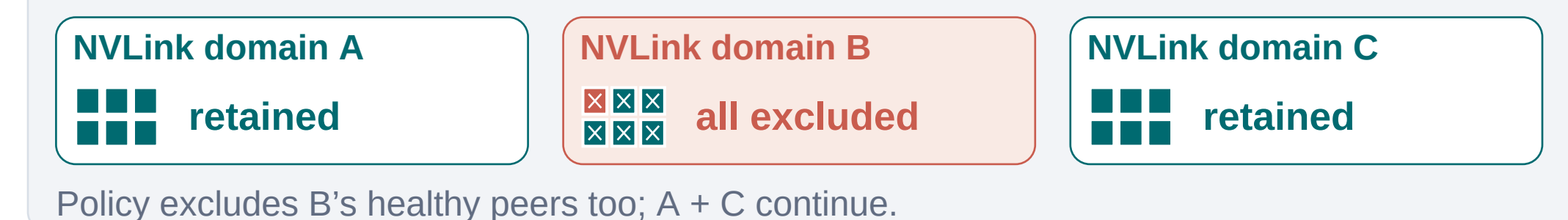
02 Shrink to useful surviving ranks

PyTorch 2.10.0 shrink_group() creates a new group; only retained ranks participate. Applications agree on exclusions and restore state. Shrinking the default group also destroys other process groups. [2]



Whole-domain exclusion

Shrink can exclude a whole replica, node or NVLink domain.



Revoke: cancel before releasing memory

REVOKE • retain resources

Cancel outstanding work; keep memory mapped. Coordinate live peers through the control plane before releasing resources. [1, 4]

ABORT • cancel and free

Teardown while live peers access memory can cause secondary errors. Fatal CUDA errors may require process restart. [4, 5]

Revoke → wait for live-peer cancellation → release

Check local completion: poll ncclCommGetAsyncError until the async state is ncclSuccess. Local completion does not replace live-peer coordination. PyTorch revoke binding: proposed. [1, 4]

Existing features and next directions

NCCL Grow | existing API

Add repaired or spare ranks; capacity restoration needs framework and workload integration. [1]

NCCL Reshape | proposed

Reconfigure membership. In-place NCCL reconfiguration and graph recapture avoidance.

PyTorch nccl2 | 2.14 release

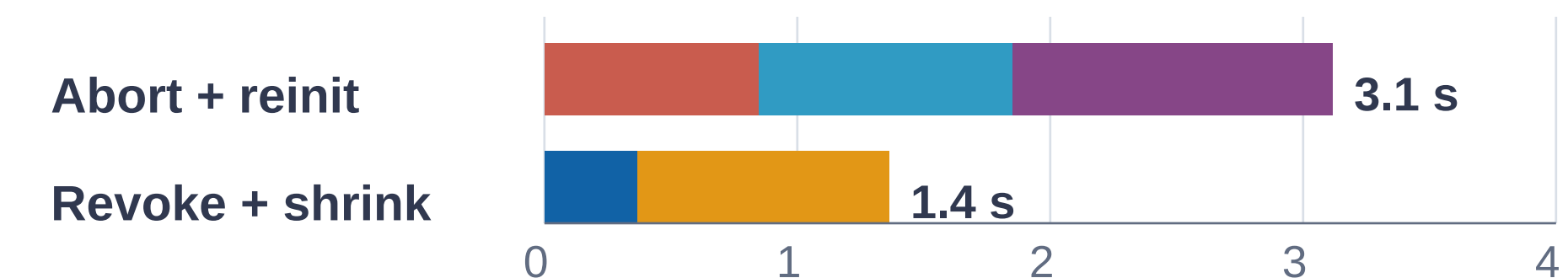
2.14 introduced ProcessGroup reconfiguration. [3] Upstream: opt-in Shrink/Grow, targeting next release. [9] Proposed Reshape integration remains an opportunity.

03 Performance and workload integration

new_group needs a healthy parent. Abort + reinit initializes the full world, then creates subgroups. Revoke cancels outstanding communication in the affected group. Shrink then forms a new group from the remaining workers, avoiding full-world reinitialization.

MEASURED • 8-GPU training

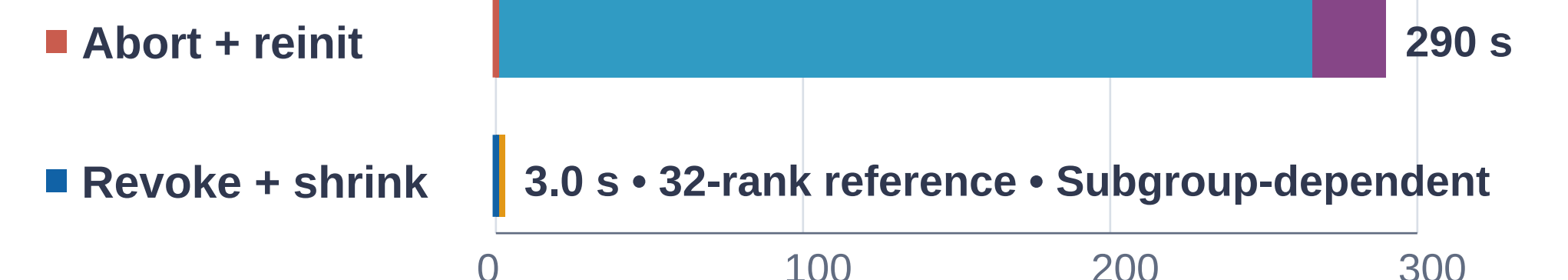
DGX H100: 8 DP ranks, one GPU per rank.



Communicator setup only; excludes state recovery and resume. Timing varies by setup.

ESTIMATED • 100K-scale training

Estimate follows Meta's training setup and 96K reference data [8]. This recovery approach is not universal; applicability and Shrink scope depend on the training setup.

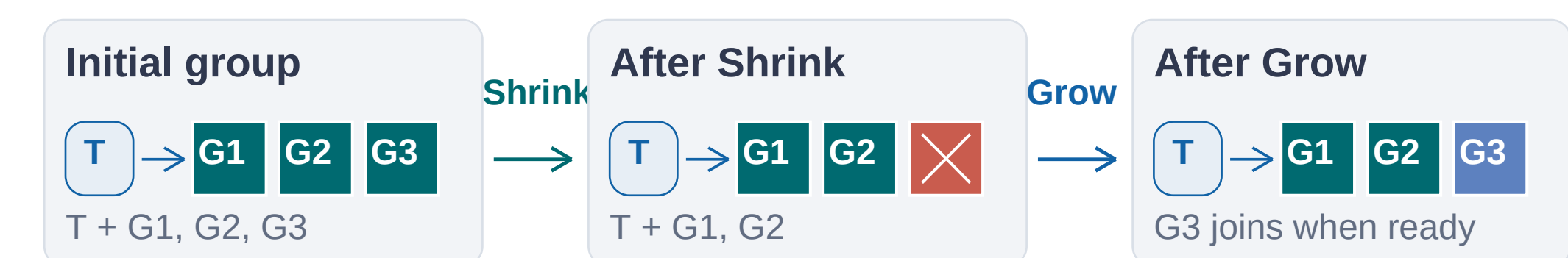


Seconds; lower is better. Each chart uses its own scale.

KEY abort init_process_group new_group revoke shrink_group

Reinforcement learning: Shrink and Grow

T = training participants; G = generation instance. Boxes show weight-update membership.



Refit: training weights → generators. Shrink excludes G3; Grow adds a ready instance. Reduced capacity; application handles routing and rollout consistency. [1, 6]

NCCL Shrink optimization opportunities

Cumulative estimates

Measured NeMo RL baseline; reuse improvements estimated.

