Sheepdog: 仮想化環境のための クラスタストレージシステム

NTTサイバースペース研究所
森田和孝

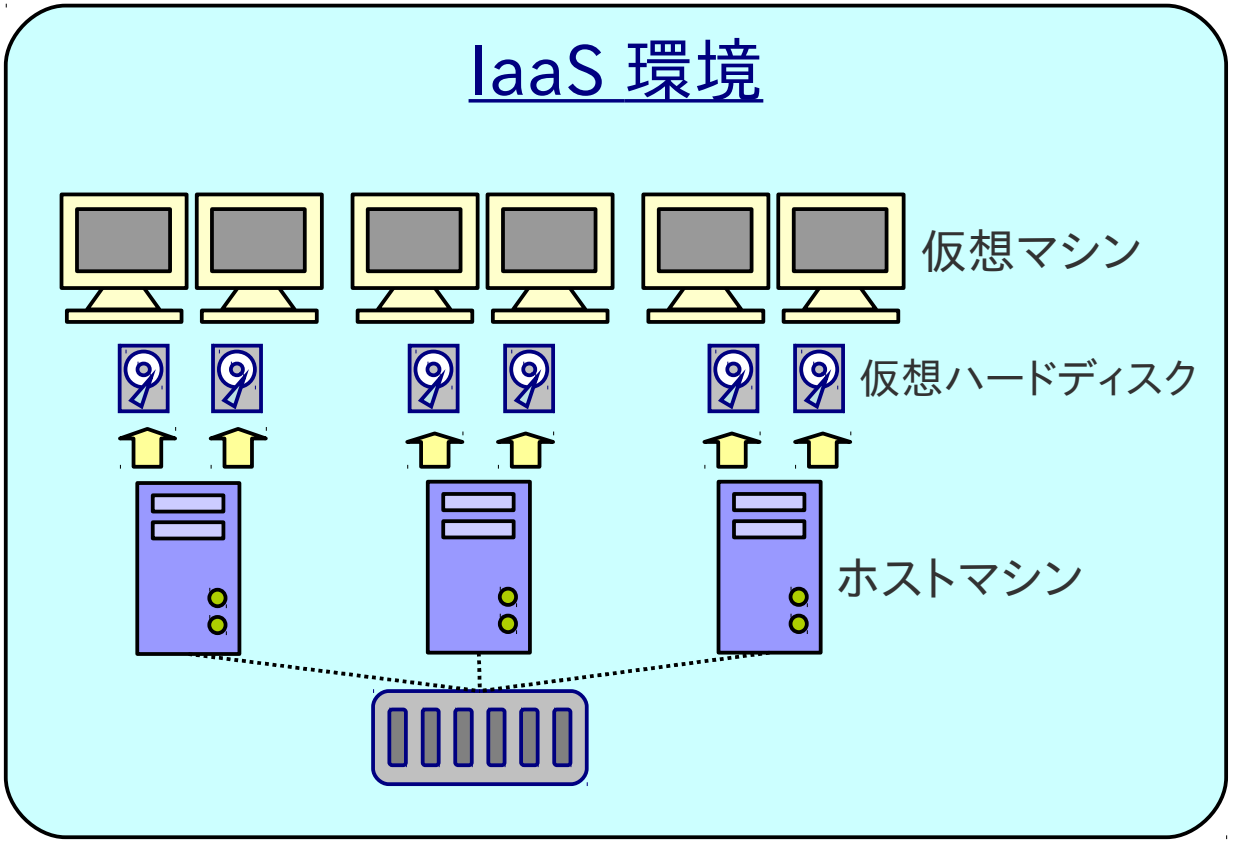
2010/11/24
Internet Week 2010

Sheepdog とは

- 仮想マシン専用のクラスタストレージシステム
  - 仮想マシンに任意のサイズの仮想ディスクを提供
  - Sheepdog のクライアント機能は QEMU 0.13.0 で標準機能として採用されている
  - OpenStack などと組み合わせて動かせるようにしようと検討中

仮想化環境用ストレージ

- IaaS 環境に適した, Amazon EBS のようなストレージシステムが OSS では存在しない

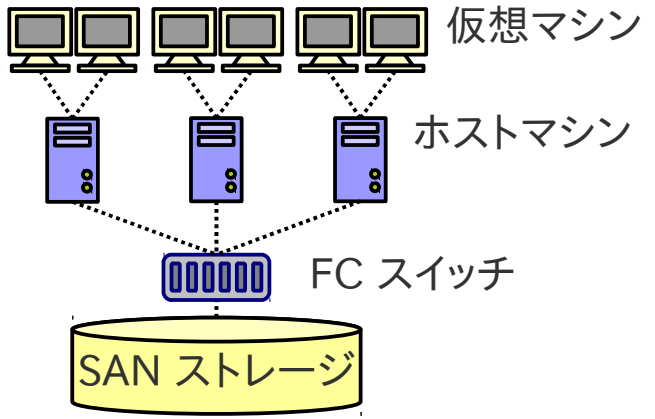


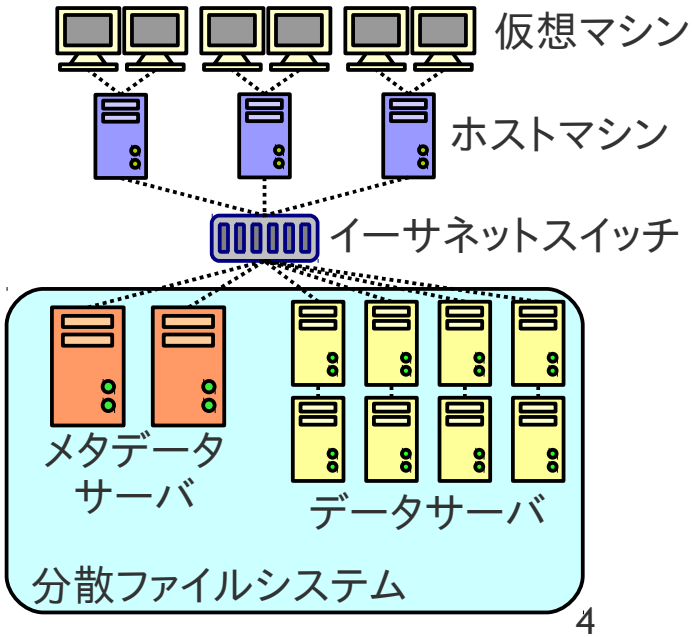
仮想化環境用ストレージの要件

- 拡張性
- 信頼性
- 運用性

既存のストレージ技術

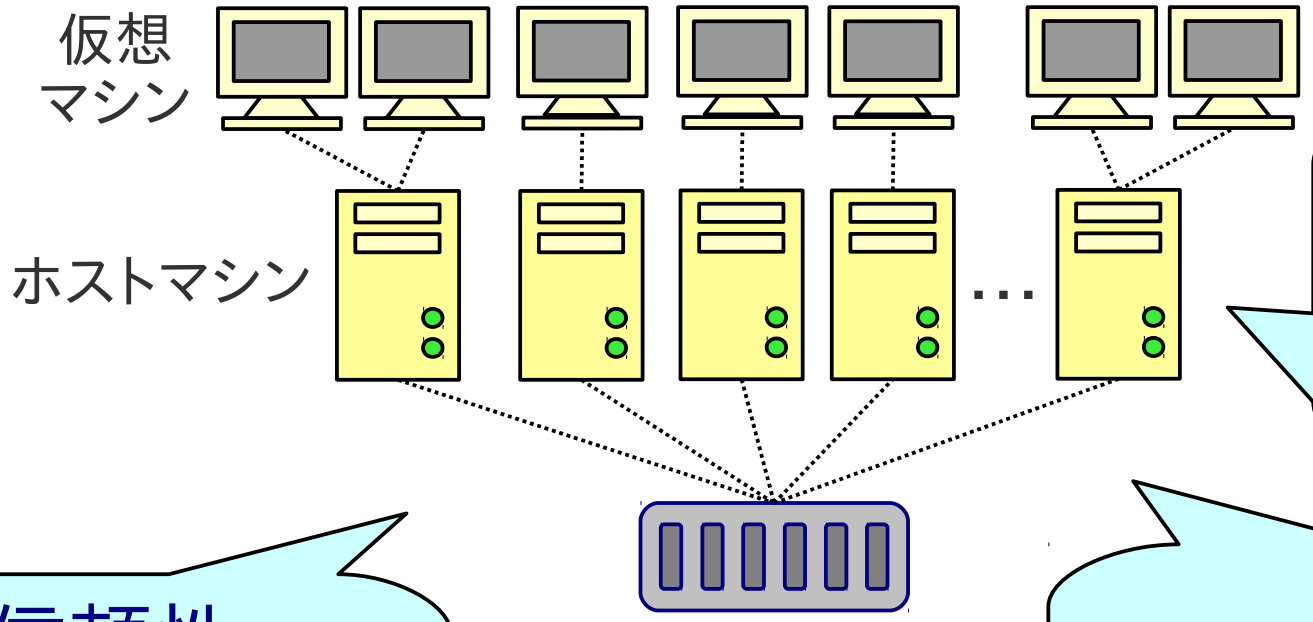
- SAN ストレージ
  - 大規模な SAN ストレージは非常に高価　拡張性×
  - 集中型アーキテクチャなので単一障害点になりうる　信頼性×
- 分散ファイルシステム (Ceph, Luster)
  - 規模が大きくなってくると, クラスタの管理が大変になる　運用性×





Sheepdog

完全等質なクラスタストレージ



拡張性

- 数百台でも動作可能

信頼性

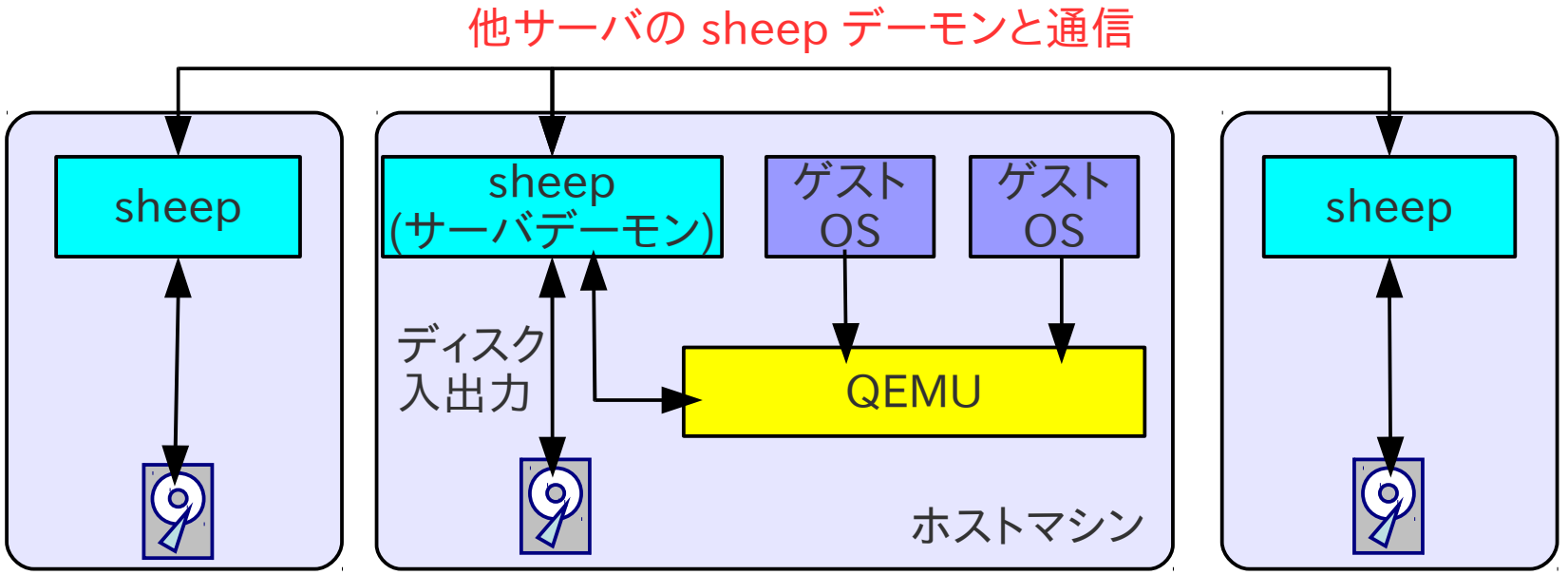
- データは冗長化されて保存されている
- 単一障害点なし

運用性

- 自律動作
- クラスタメンバの動的管理
- 高度なディスク操作機能

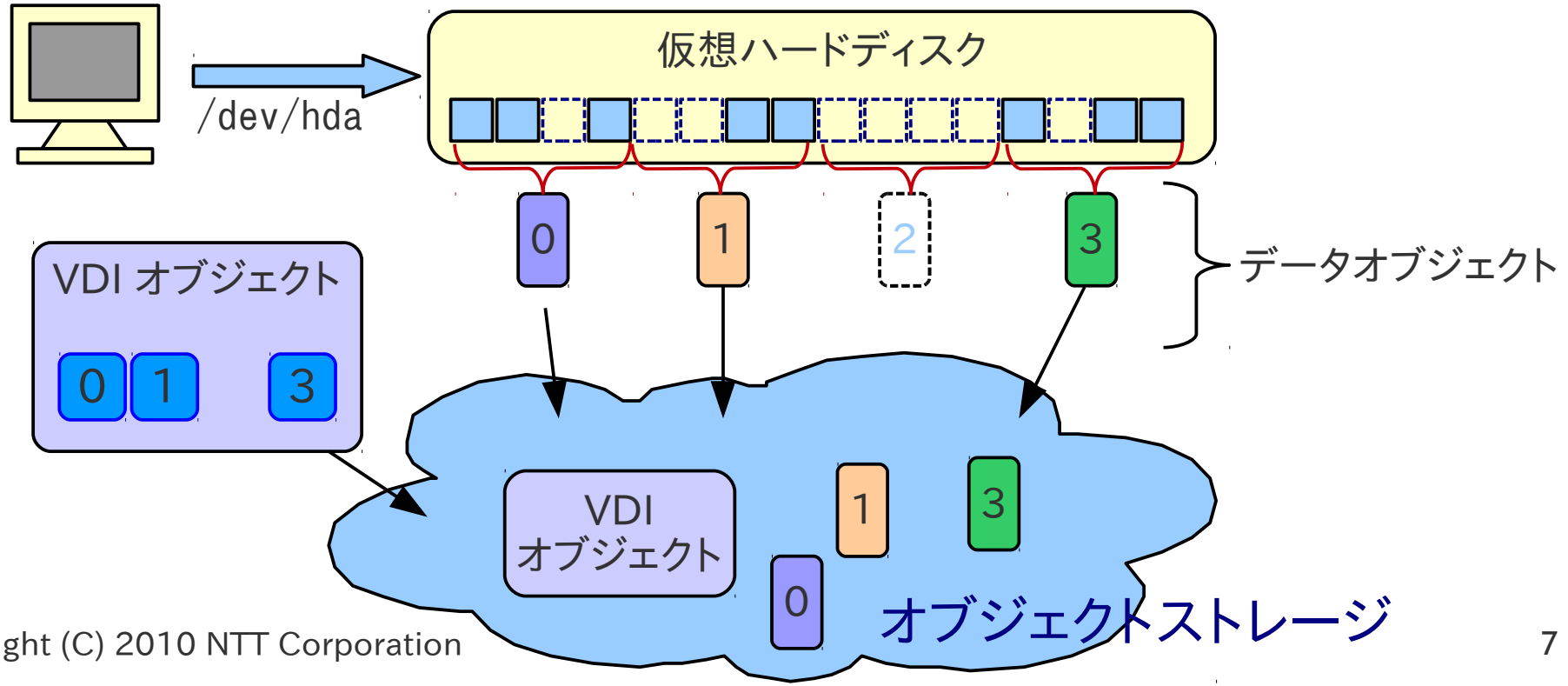
設計: 仮想マシン専用ストレージ

- 制限を加えることで, 非常にシンプルな設計を実現
  - API は 仮想化ソフトウェア QEMU に特化
  - 通常のファイルシステムとしては利用できない
  - 同じ仮想ディスクを複数の仮想マシン間で共有できない



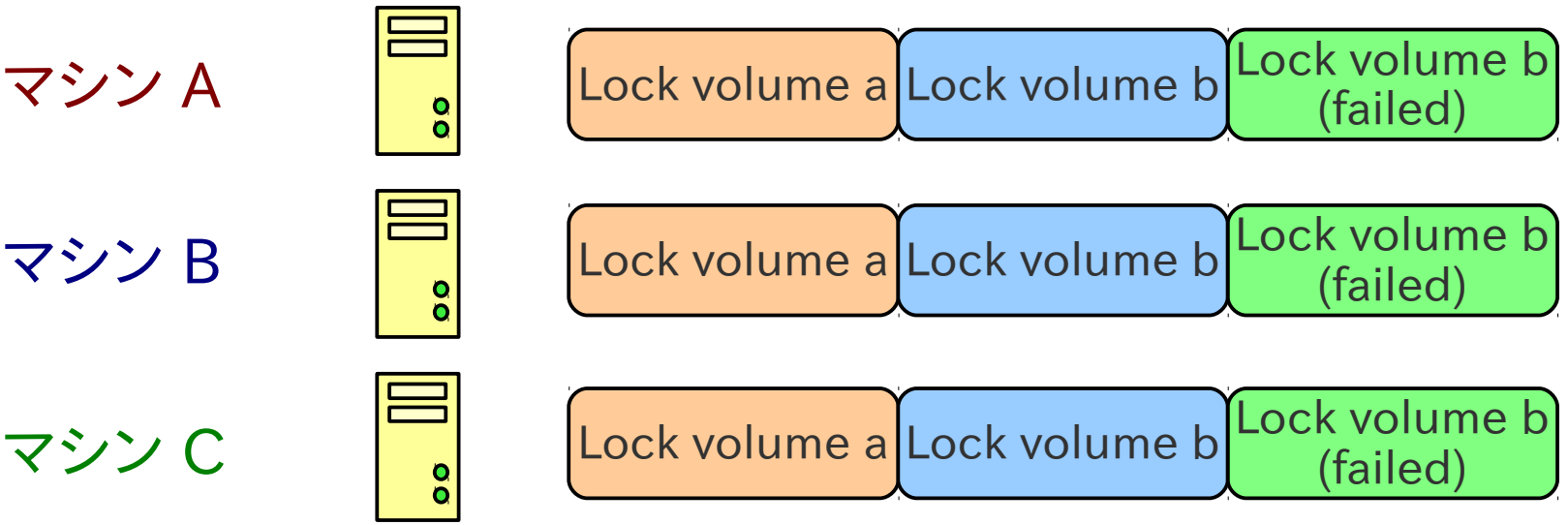
仮想ハードディスクの保存方法

- 仮想マシンにオブジェクトストレージを提供
  - 仮想ハードディスクは 4 MB 単位のデータオブジェクトに分割されて保存される
  - 仮想ハードディスクとデータオブジェクトの対応関係は VDI オブジェクトに保存される

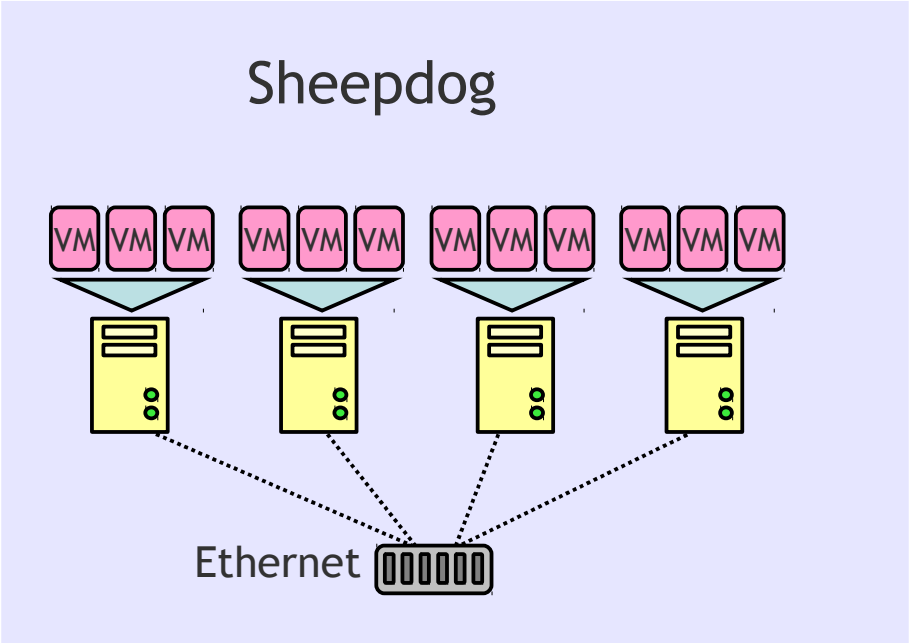


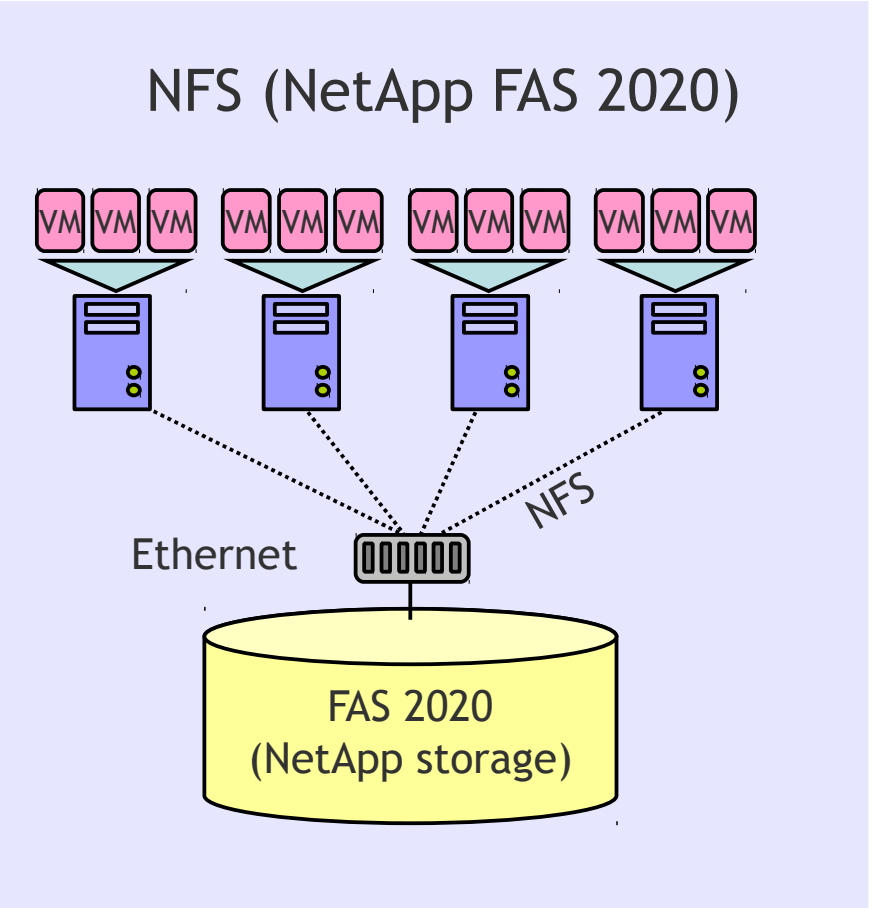
クラスタのマシン管理

- Corosync
  - 高信頼アトミックマルチキャスト, 動的メンバ管理の実装
  - 有名な OSS (Pacemaker, GFS2, etc) に採用されている
- Sheepdog は Corosync の高信頼マルチキャストを用いて, メタデータサーバを不要にした



性能

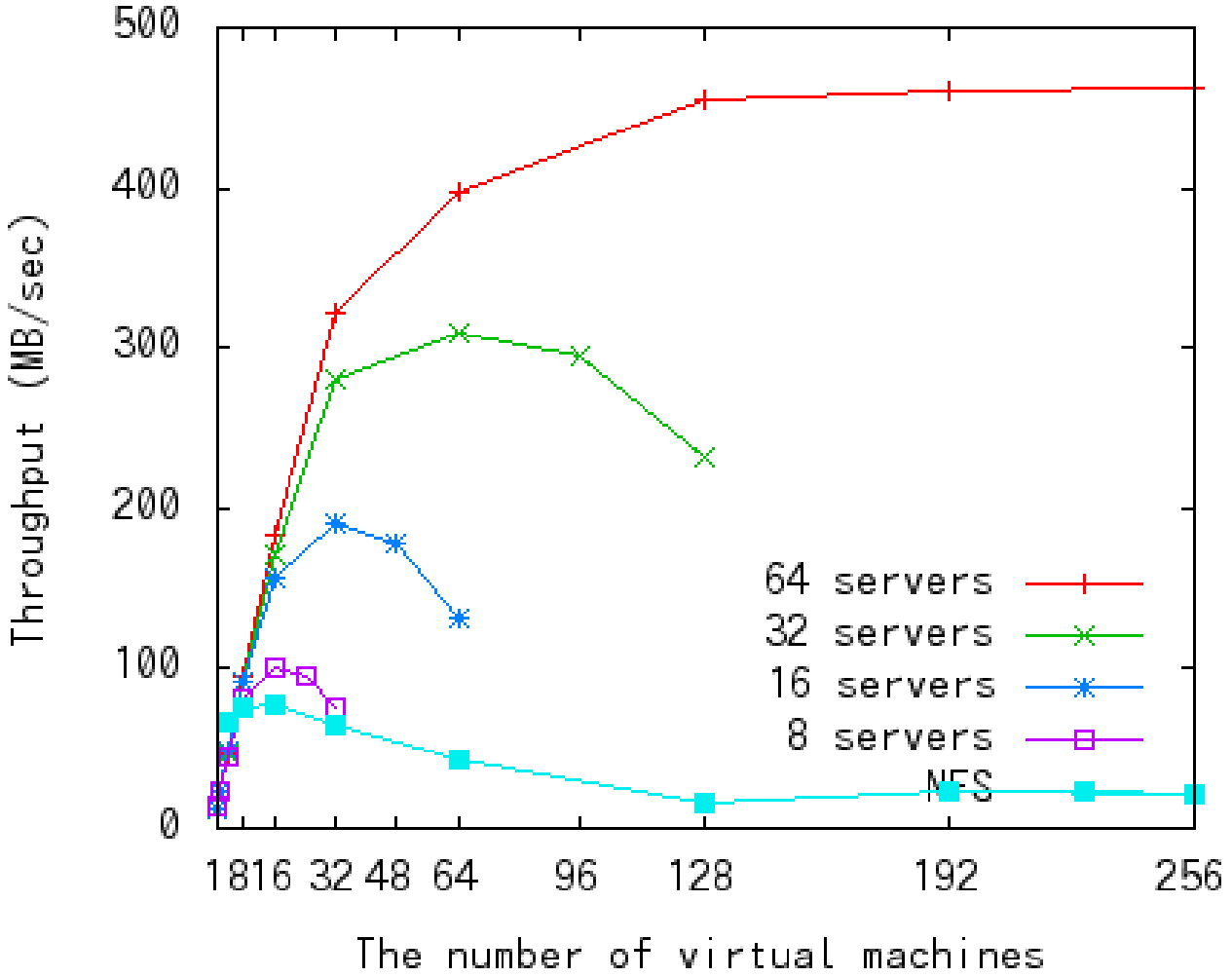




CPU : Core2 Quad 2.4GHz
メモリ : 2 GB
ネットワーク : 1 Gbps
ディスク : SATA 7200 rpm
ホストマシン台数 : 8 ～ 64
仮想マシン台数 : 1 ～ 256
データ冗長度 : 3

性能

```
$ dbench -s -S
```



物理マシンを増やすと
全体のスループットが
上昇する

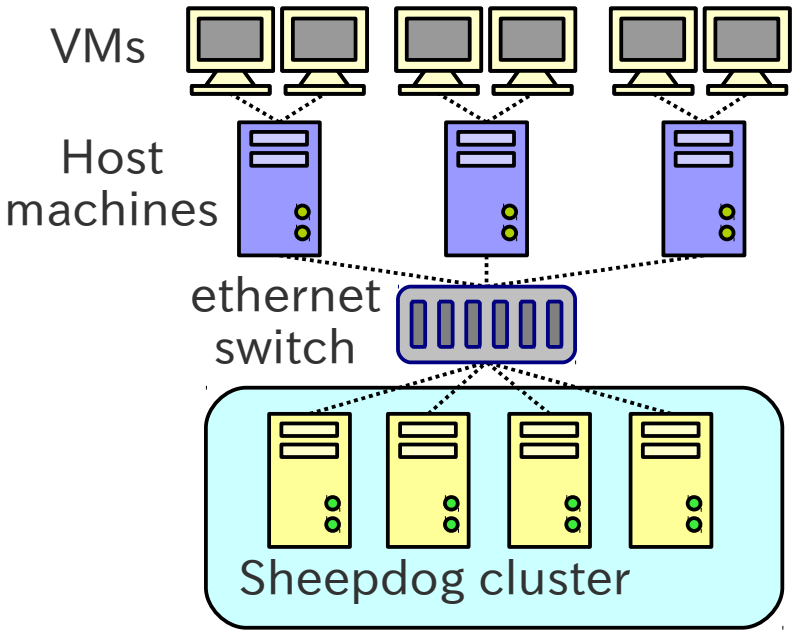
まとめ

- Sheepdog は拡張性, 運用性, 信頼性を考慮した, IaaS 環境用のクラスタストレージシステム
  - Sheepdog のクライアントは QEMU 0.13.0 より標準機能化
  - 開発者, ユーザ募集中
- その他の情報
  - プロジェクトページ
    - http://www.osrg.net/sheepdog/
  - メーリングリスト
    - sheepdog@lists.wpkg.org

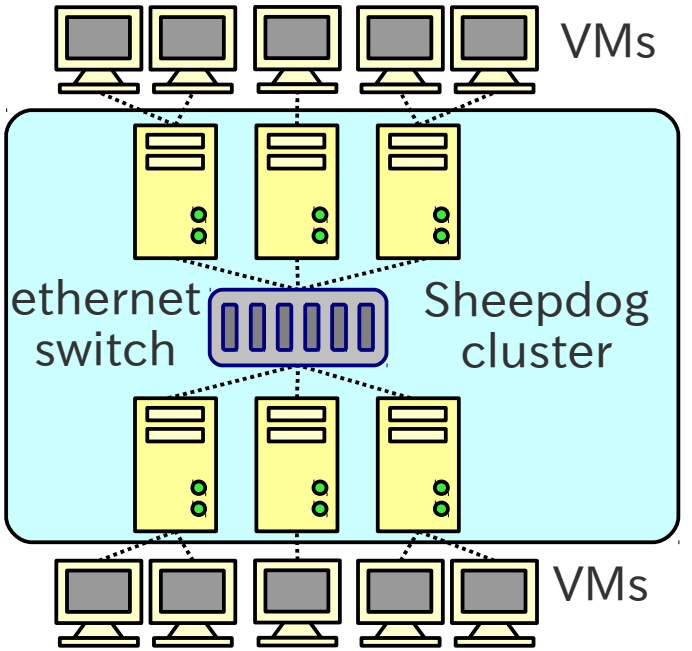
付録

Architecture: fully symmetric

- Zero configuration about cluster members
- Similar to Isilon architecture



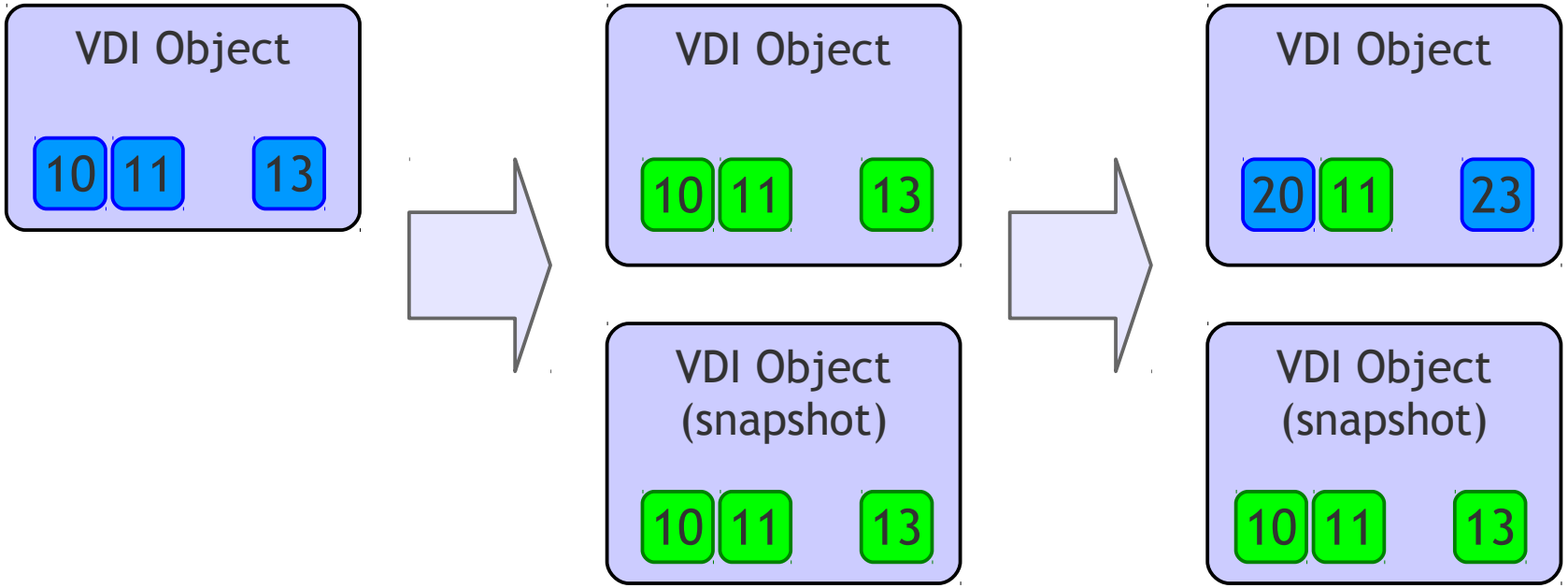
Use sheepdog as a network storage



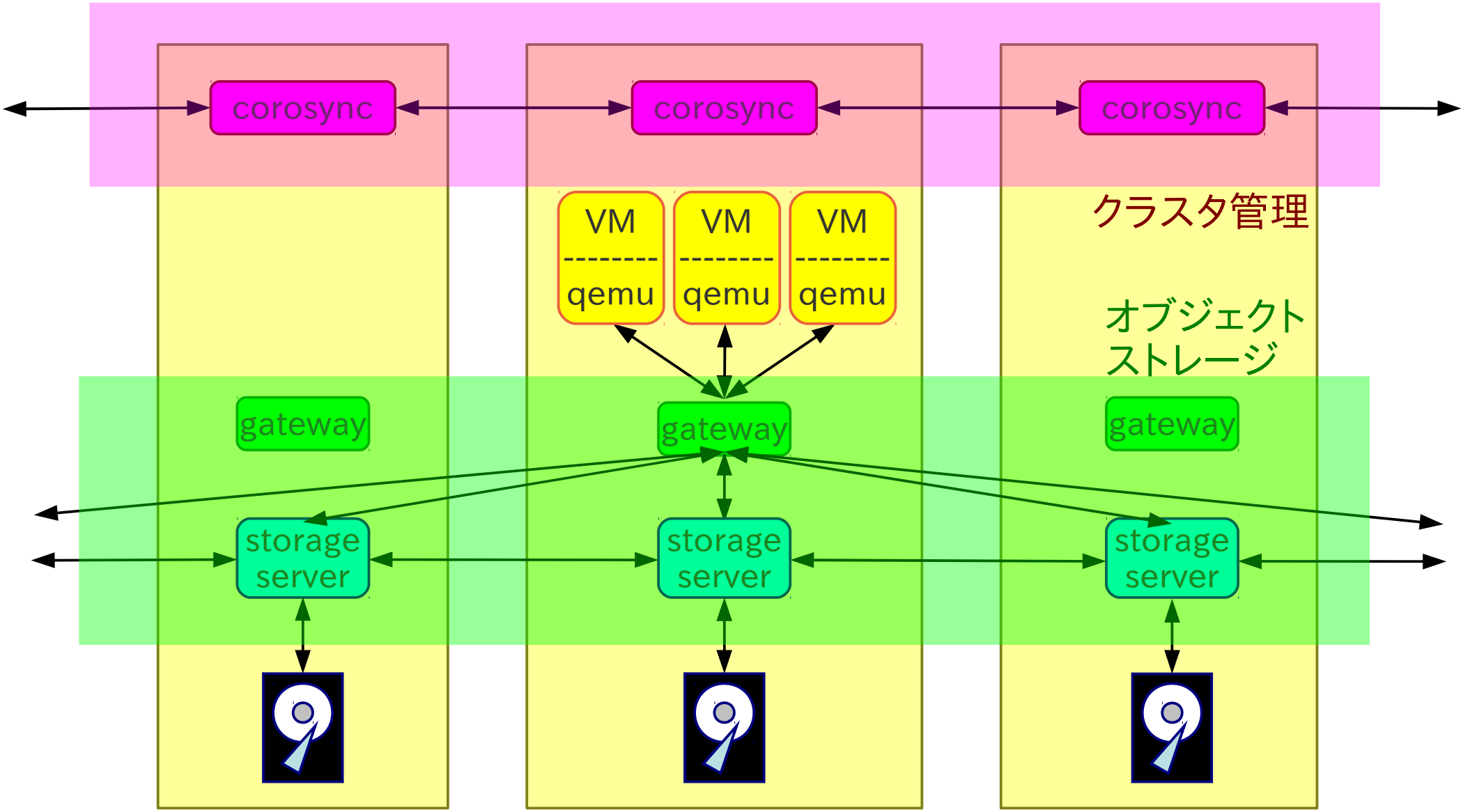
Use sheepdog as a virtual infrastructure

Snapshot

- Copy VDI Object, and make allocated data objects read-only
- Updating read-only objects causes copy-on-write

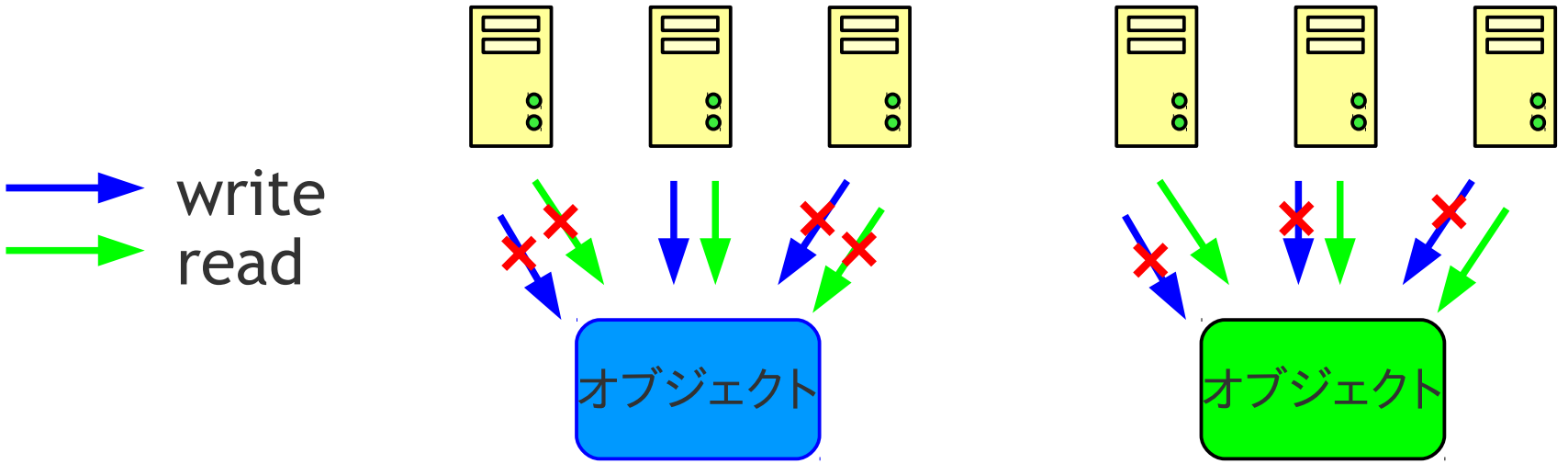


Sheepdog の構成要素



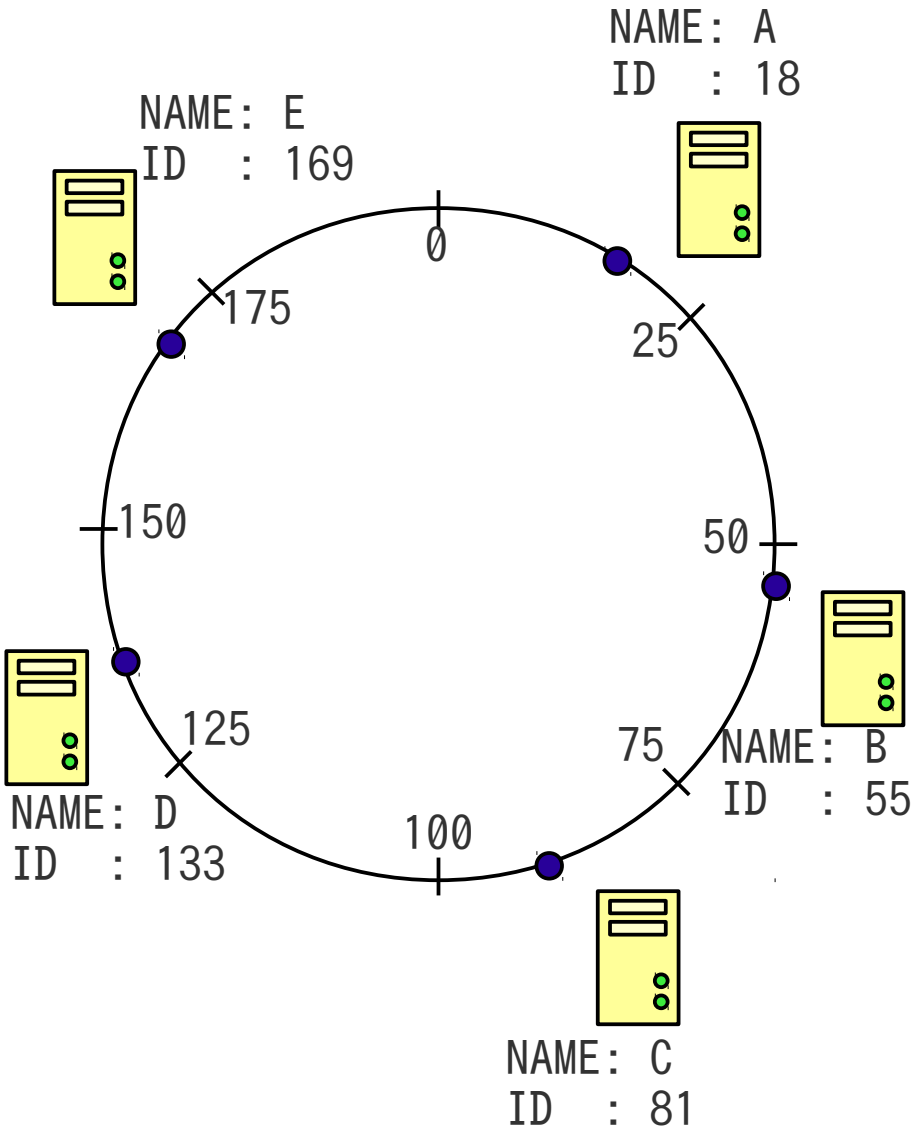
オブジェクトストレージ

- 可変長のデータ(オブジェクト)を一意の識別子を指定して保存できる
- クライアントはオブジェクトがどこに保存されるかを気にしなくてよい
- Sheepdog に存在するオブジェクトは二種類
  - Writable オブジェクト
  - Read-only オブジェクト



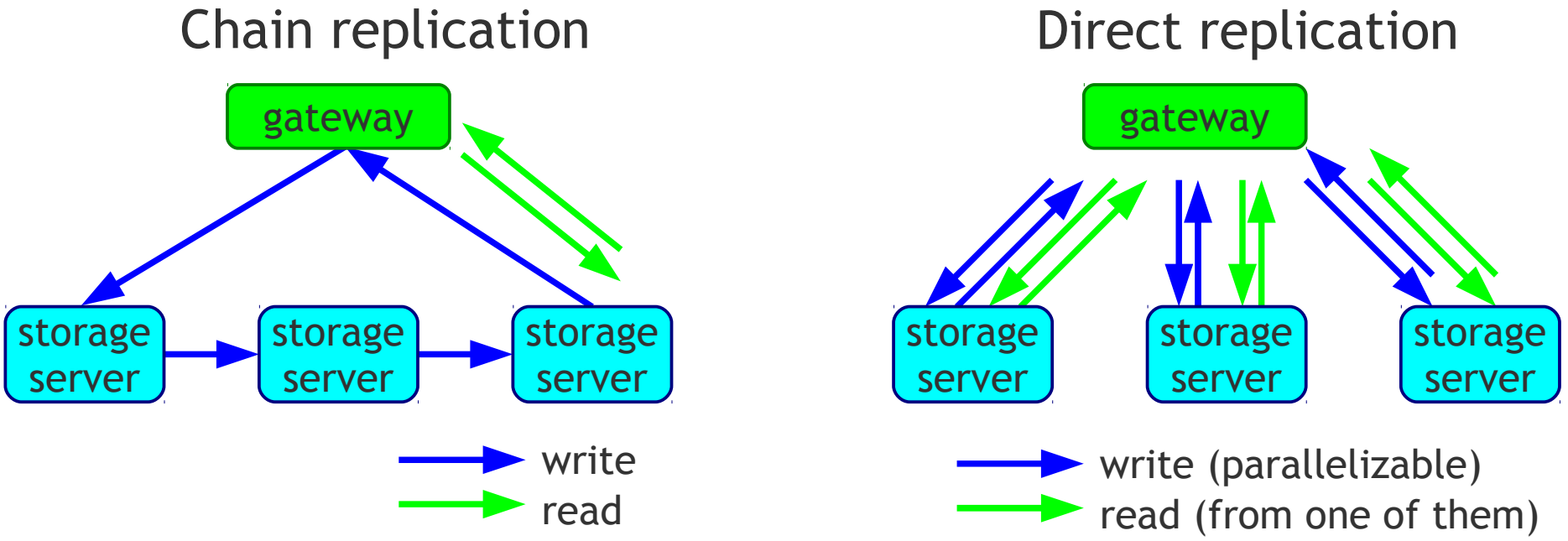
オブジェクトストレージの実装

- コンシステントハッシュ法を利用
  - マシンの参加や離脱によって, データ配置が大きく変わらないことが特徴
  - Sheepdog のサーバ, オブジェクトはコンシステントハッシュリングの上に配置される



Replication

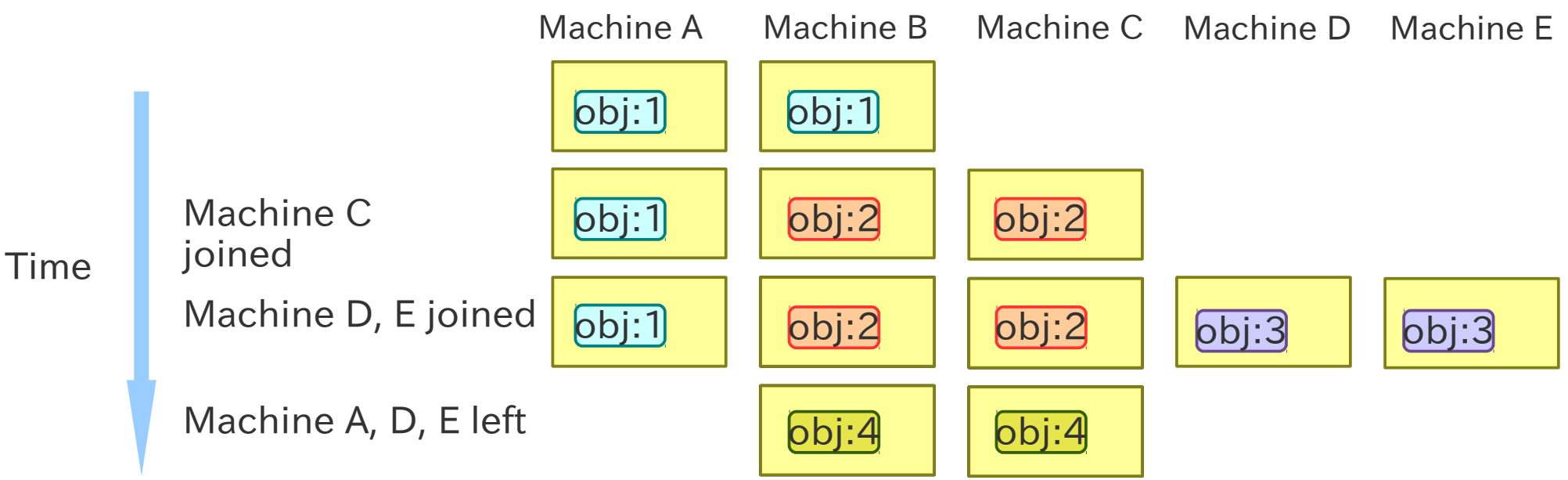
- Many distributed storage systems use chain replication to maintain I/O ordering
- Sheepdog can use direct replication because write collision cannot happen



Node membership history

- All nodes store the history of node membership
- Objects are stored with the version of node membership (epoch)

epoch	Node membership
1	A, B
2	A, B, C
3	A, B, C, D, E
4	B, C

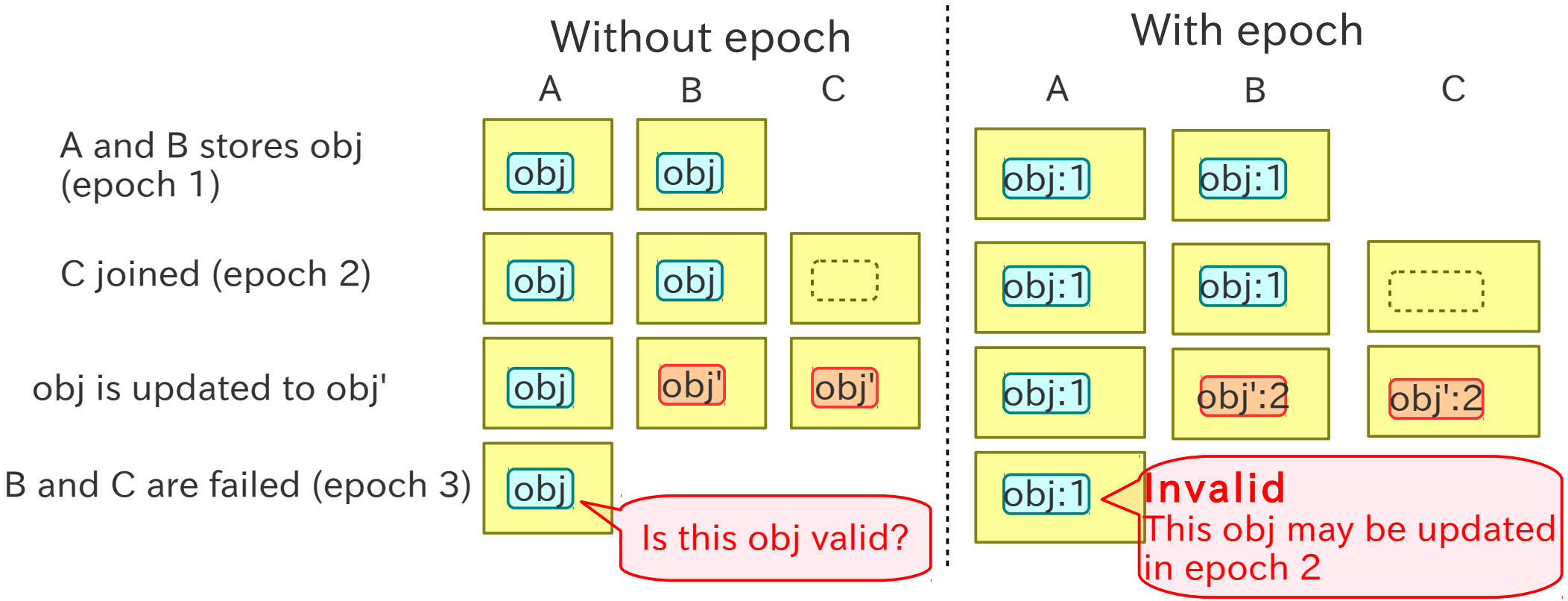


クラスタのマシン管理

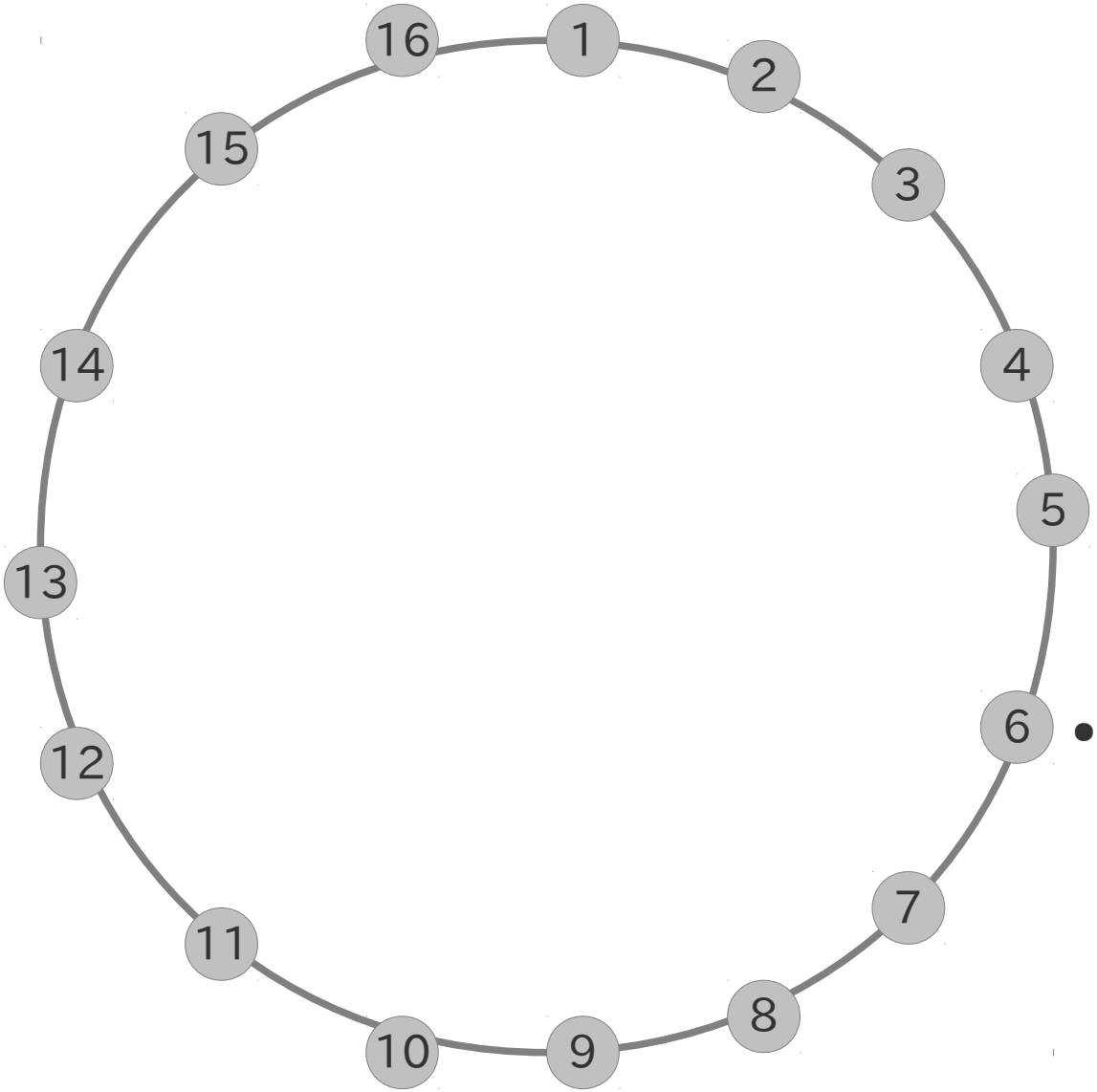
- Totem リングプロトコル
  - 動的なメンバの管理を実現
  - 全順序かつ高信頼なマルチキャストを実現
  - 仮想同期を実現

Strong consistency

- Avoid reading old objects
- If requested object is not valid, system must return I/O error

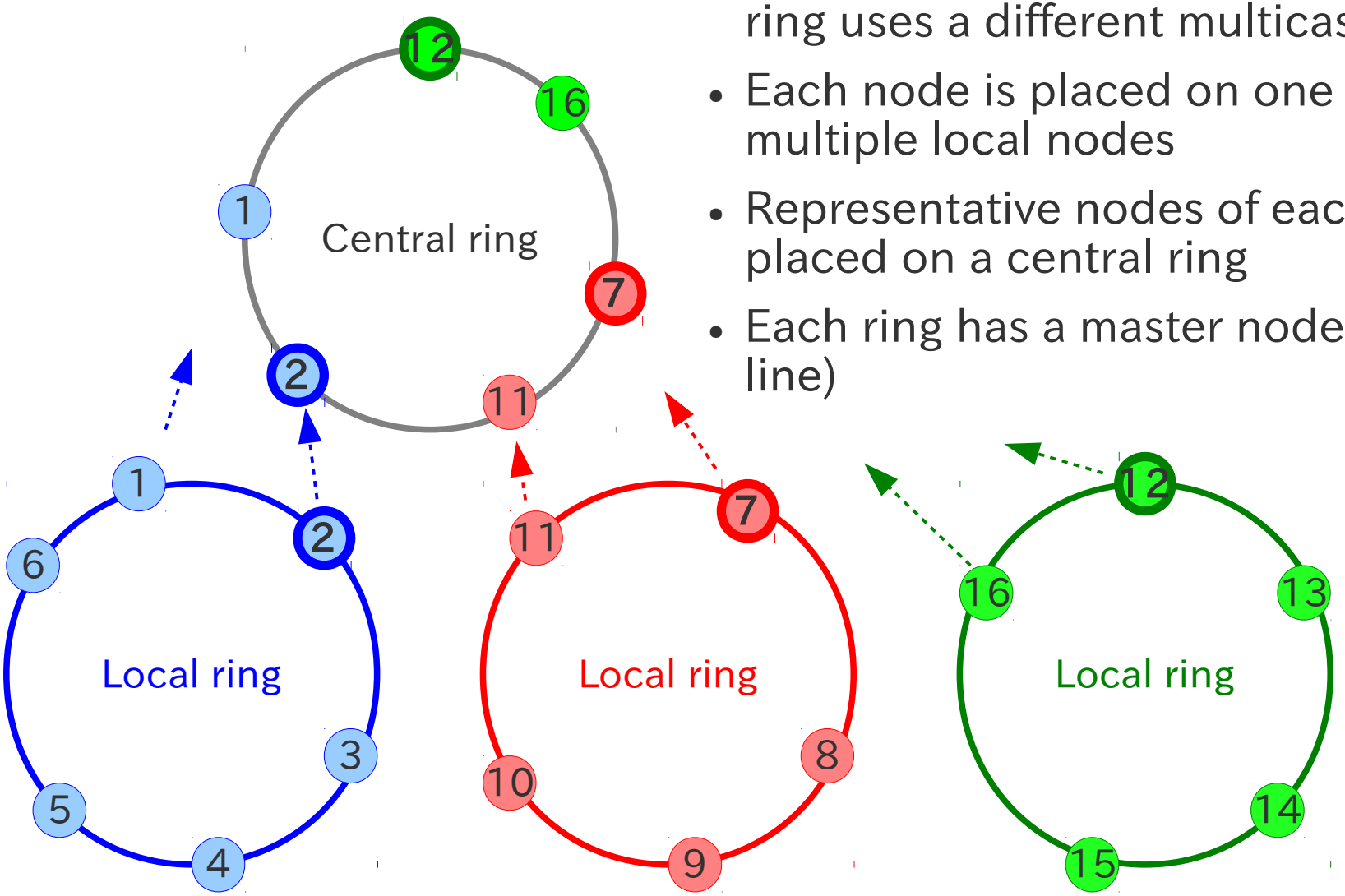


Corosync problem



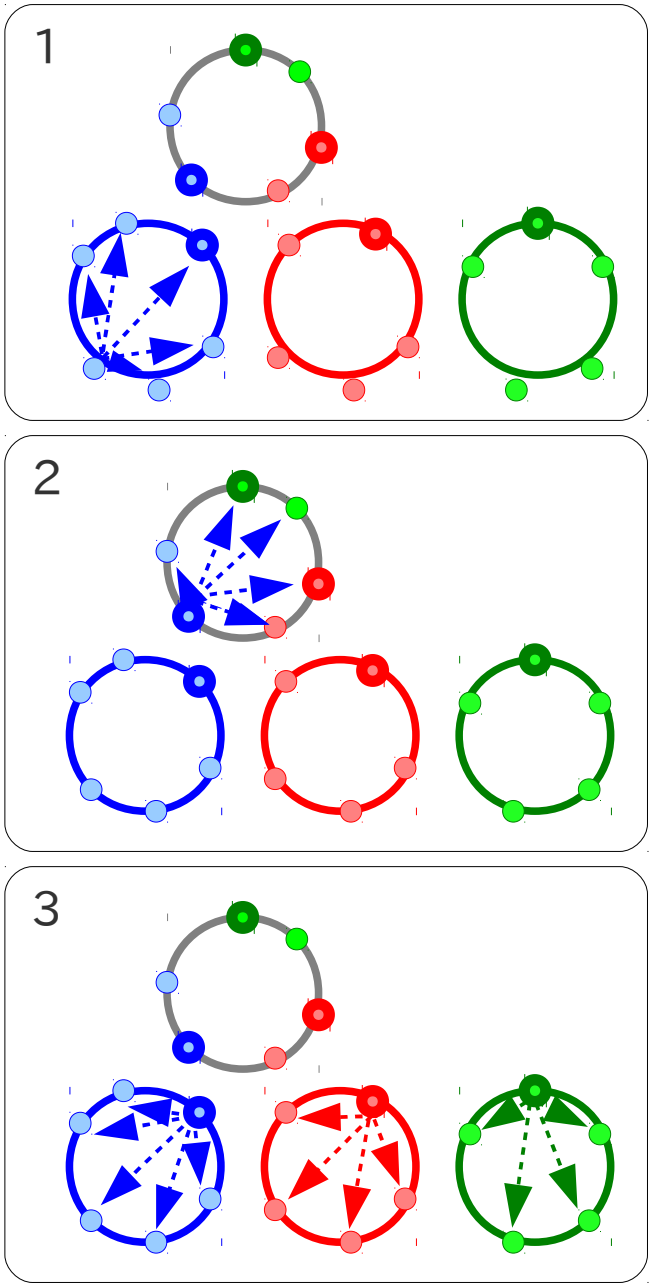
- Totem single-ring protocol with a large number of nodes doesn't work well

Approach: use multiple rings



- Consider several local rings (blue, red, and green) and one central ring (gray). Each ring uses a different multicast port.
- Each node is placed on one of the multiple local nodes
- Representative nodes of each ring are also placed on a central ring
- Each ring has a master node (with a bold line)

Total order multicast



1. Send multicast message in the local ring
2. If master node receives the message, the node resends the multicast message in the central ring
3. If master nodes receive the message in the central ring, each master node resends the multicast message in the local ring

Message ordering is coordinated by multicast in the central ring