PGCluster Round Table

PGCluster円卓会議

At.Mitani

PostgreSQL 10th Anniversary Summit

9 July 2006 Toronto

アジェンダ

Agenda

- エンタープライズ規模で利用可能にしてほしいという要望がある
- レプリケーションがそんな要求をみたすことができるのか？
- PGClusterについて
  - Feature 機能
  - Structu レプリケーションの仕組み
  - demons ion デモ
  - Replica レプリケーションの将来
  - HA fea 高可用性機能
- PGClusterへのその他の要望 uirement for PGCluster
- PGClusterIIのコンセプト PGCluster-II
  - Feature 機能
  - How to tributed servers 分散サーバ間の共有IPCの方法
  - Structu 構造
  - demons ion デモ
  - Expect ffect and problem 期待される効果と問題点
- 質疑応答 ussion

Issues of Enterprise usage

エンタープライズ規模での利用上の問題

Requirement of Enterprise usage

エンタープライズ規模での利用に求められるもの

- 巨大なデータの取り扱い unt of data operation
  - SANやNASを利用した巨大なストレージの使用 SAN / NAS
- 大量のリクエストに対する迅速なレスポンス n a lot of request.
  - 検索（例：ウェブアプリ） o application)
  - 更新（例：ERPシステム） P system)
- 高可用性 availability (HA)
  - 無停止オペレーション -day a week operation
  - 重大な事故からの復旧 very
- 簡単なバックアップとリストア nd restore
  - 差分バックアップ/リストア k up / restore

Does replication help of those requirement ? (1/4)

レプリケーションはこんなことの役に立つのか？（1/4）

- 巨大なデータの操作 ta operation
  - あまり意味がない
    - 操作するデータの量を減らすものではありません nt of operation data
  - その他のソリューション on
    - PostgresForest / pgpool-II

Does replication help of those requirement ? (2/4)

レプリケーションはこんなことの役に立つのか？（2/4)

- 負荷分散 l distribution
  - 効果があるところ
    - 検索の負荷が軽減します earch load
  - あまり効果がないところ
    - 更新の負荷はもっと上がってしまいます！ e heavy!
  - その他のソリューション n
    - クラスタ/グリッドコンピューティン
    - これで鉄板というツールはわかりません ete tools

Does replication help of those requirement ? (3/4)

レプリケーションはこんなことの役に立つのか？（3/4）

- 高可用性 availability
  - 効果あり
    - マスタ-スレーブ構成では、テイクオーバーの際にサービスが停止します vice stop time during take over.
    - 複数マスタ構成では、テイクオーバーの際にもサービス停止は発生しません rvice stop time during take over.
  - その他のソリューション on
    - サーバレベルの高可用性のための商用ツールがあります s for server level HA.

Does replication help of those requirement ? (4/4)

レプリーケーションはこんなことの役に立つのか？（4/4）

- 簡単なバックアップとリストア and restore
  - 効果あり
    - レプリケーションされたデータベースはリアルタイムのバックアップになります
  - その他のソリューション
    - 差分バックアップとリストア up / restore

PGCluster / PGCluster-II

PGClusterとPGCluster-II

History 歴史

- PGReplicate-0.1 released (2002 Mar)
  - クエリベースの単純なレプリケーション query
- PGCluster-1.0 released (2003 July -)
  - トランザクション、コピー、ストアードプロシージャ lure
- PGCluster-1.1 released (2004 Sep -)
  - ロードバランサにPgpoolを利用 lancer
- PGCluster-1.3 released (2004 Dec -)
  - ラージオブジェクト
- PGCluster-1.5 released (2006 Jan -)
  - V3プロトコル、プリペアードクエリ uery
- PGCluster-II started (2006 June -)

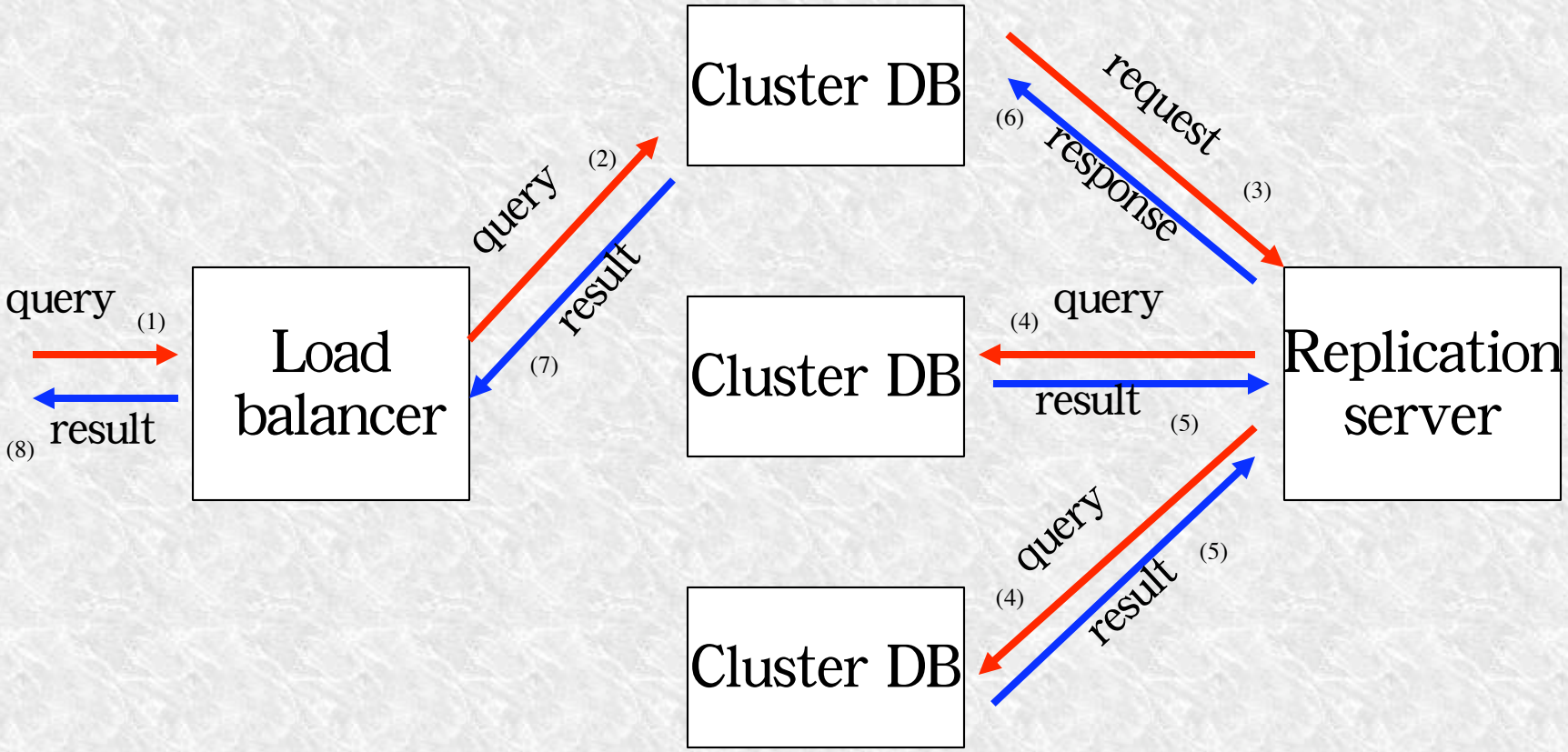
Feature of PGCluster

PGClusterの機能

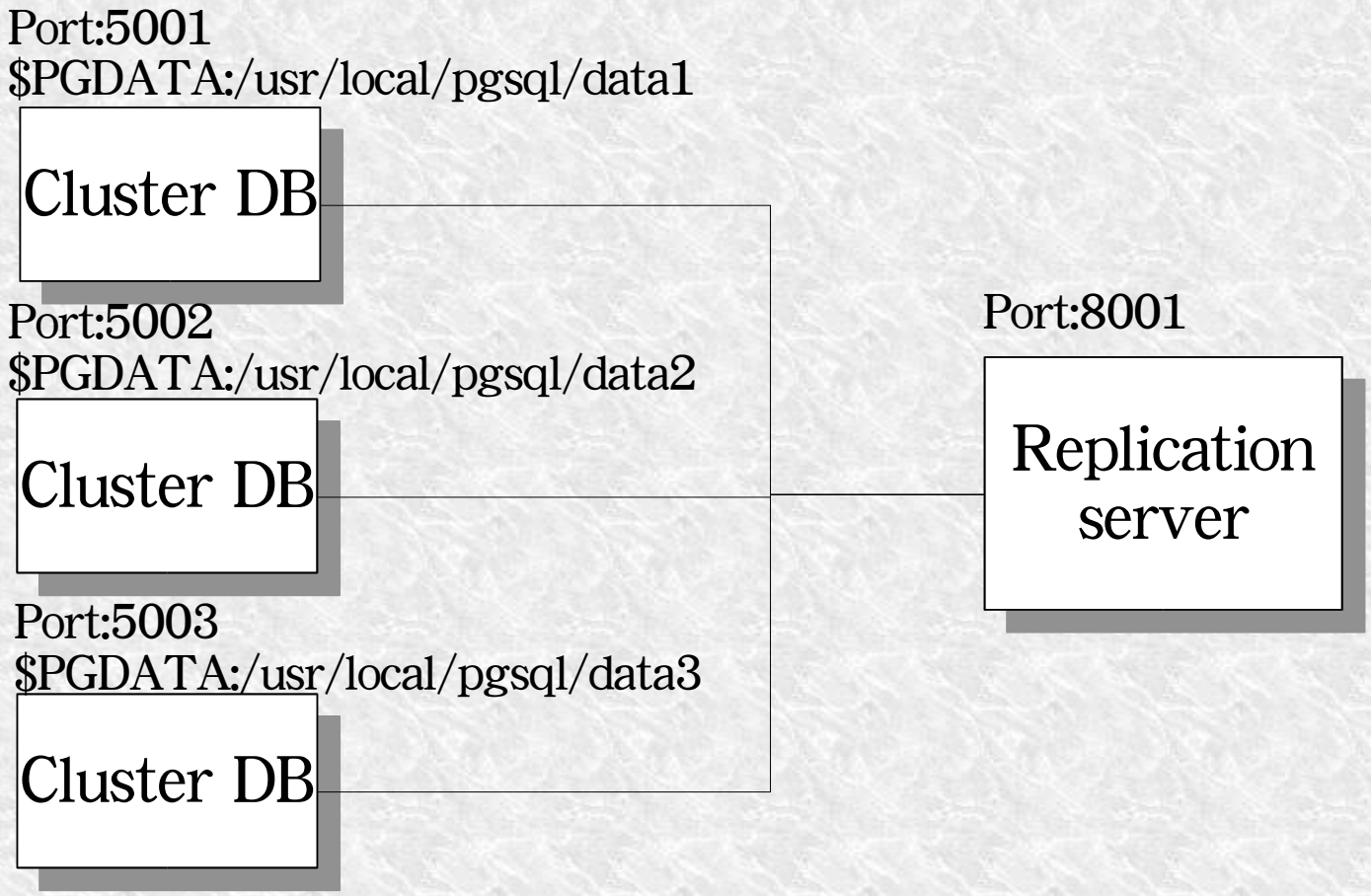
- マルチマスタ＆同期レプリケーションシステム onous replication system
  - 適している用途 or
    - 読み込みの負荷が高いシステム
    - 高可用性の求められるシステム
    - リアルタイムバックアップ
  - 適していない用途 le for
    - 書き込みの負荷が高いシステム tem
    - 巨大なサイズのデータベース e
    - Mobile database 構成変更の多いデータベース

Structure of replication

レプリケーションの構造

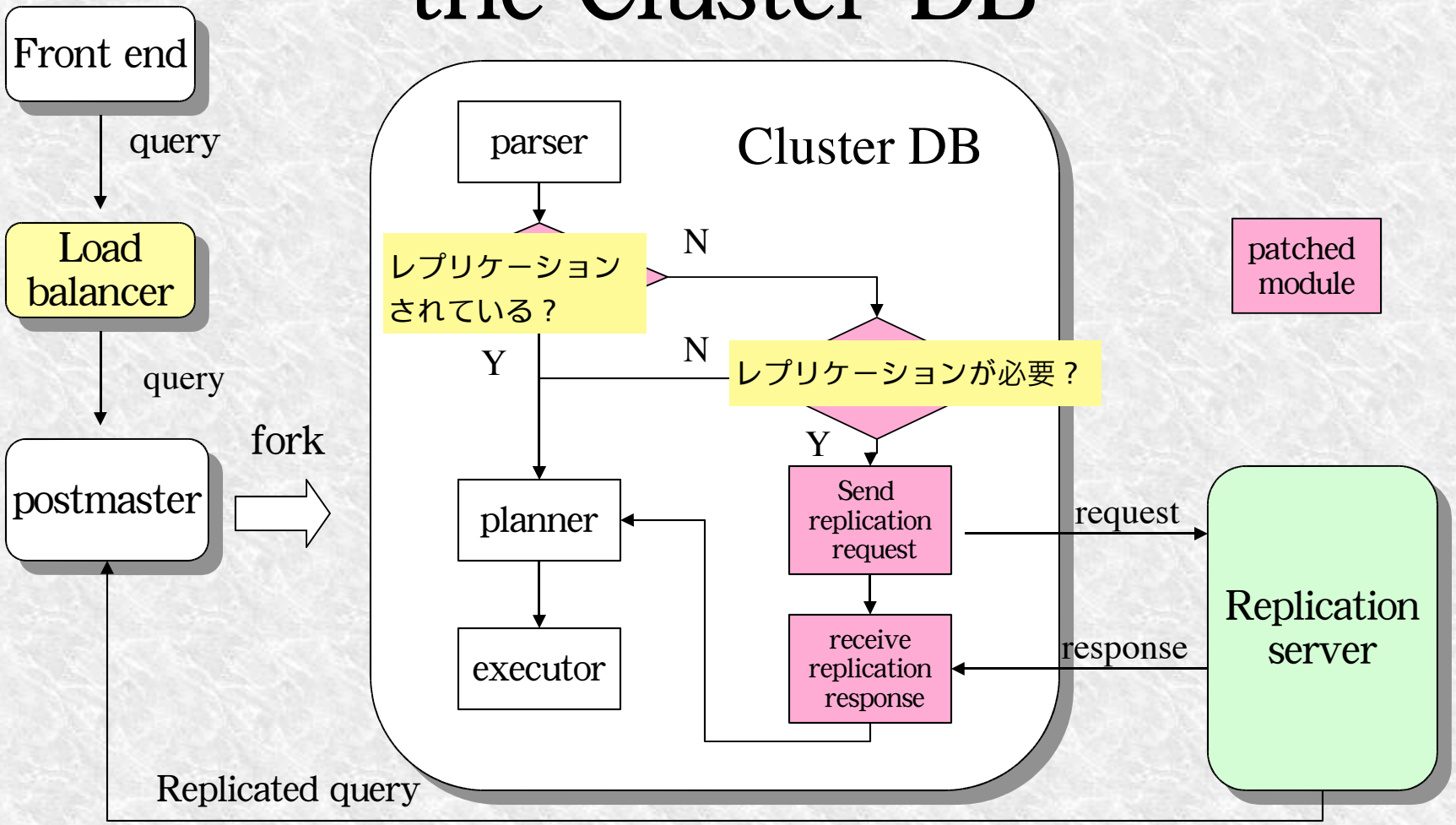


Demonstration of PGCluster



Replication patch of the Cluster DB

クラスタDBでレプリケーションが担当するところ



What can be replicated

レプリケーションされるもの

- トランザクション on
- ストアードプレシージャ edure
- 内部関数の戻り値 alue of internal function
  - now() ,random()
- ラージオブジェクト ct
- プリペアードクエリ query
- Copy (from)
- Create / Drop
  - DB , table, domain, function, group …

High Availability support

高可用性のサポート

- 自動テイクオーバー c take over
  - クラスタDB DB
  - レプリケーションサーバ erver
- 動的なサーバ追加 c server addition
  - クラスタDB DB
  - レプリケーションサーバ erver
- 代替のない破損箇所がないこと ngle point of failure
  - 全てのサーバで複数台構成が可能 et up multiple

Yet another requirement for PGCluster

さらにPGClusterへの要望

- 書き込みの負荷分散 dispersion
- 巨大なサイズのデータベース atabase
- リカバリの時間短縮 recovery

Write load dispersion & Large size of database

書き込みの負荷分散と巨大なサイズのデータベース

- レプリケーションは書き込みの負荷分散の役には立たない for write load distribution
  - 現在のバージョンのスコープにはない e for current version.
- 必要とされる分散データベース ributed database
  - パラレル操作 l operation
    - PostgresForest
    - Pgpool-II
  - 分散サーバ間の共有IPC during distributed servers

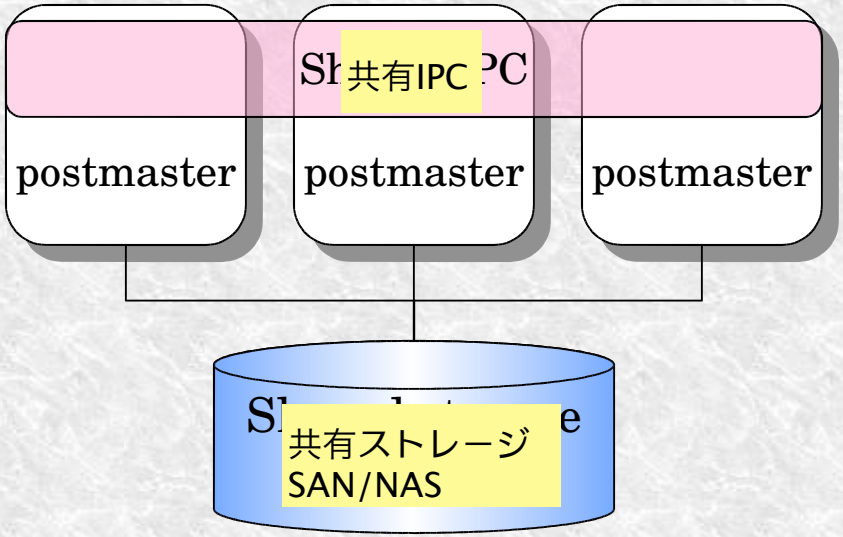
Short ... covery

リカバリの時間短縮

- システム変更の多いユーザからの要求
  - ...ile users
  - The files under $PGDATA are copied using "rsync" command
    - $PGDATA以下のファイルはrsyncを利用してコピーされる
  - 差分のあるファイルのみコピーする ... a difference is copied
    - 差分が小さくても、ファイル全体がコピーされる, a file is all copied.
  - アーカイブログを利用するべき... ...d be used...

Concep PGCluster-IIのコンセプト luster-II

- 共有DBクラスタ DB cluster
  - 共有ストレージ（SAN/NASを利用） sing SAN / NAS)
- 分散postmaster ted postmaster
  - 分散サーバ間の共有IPC during distributed servers

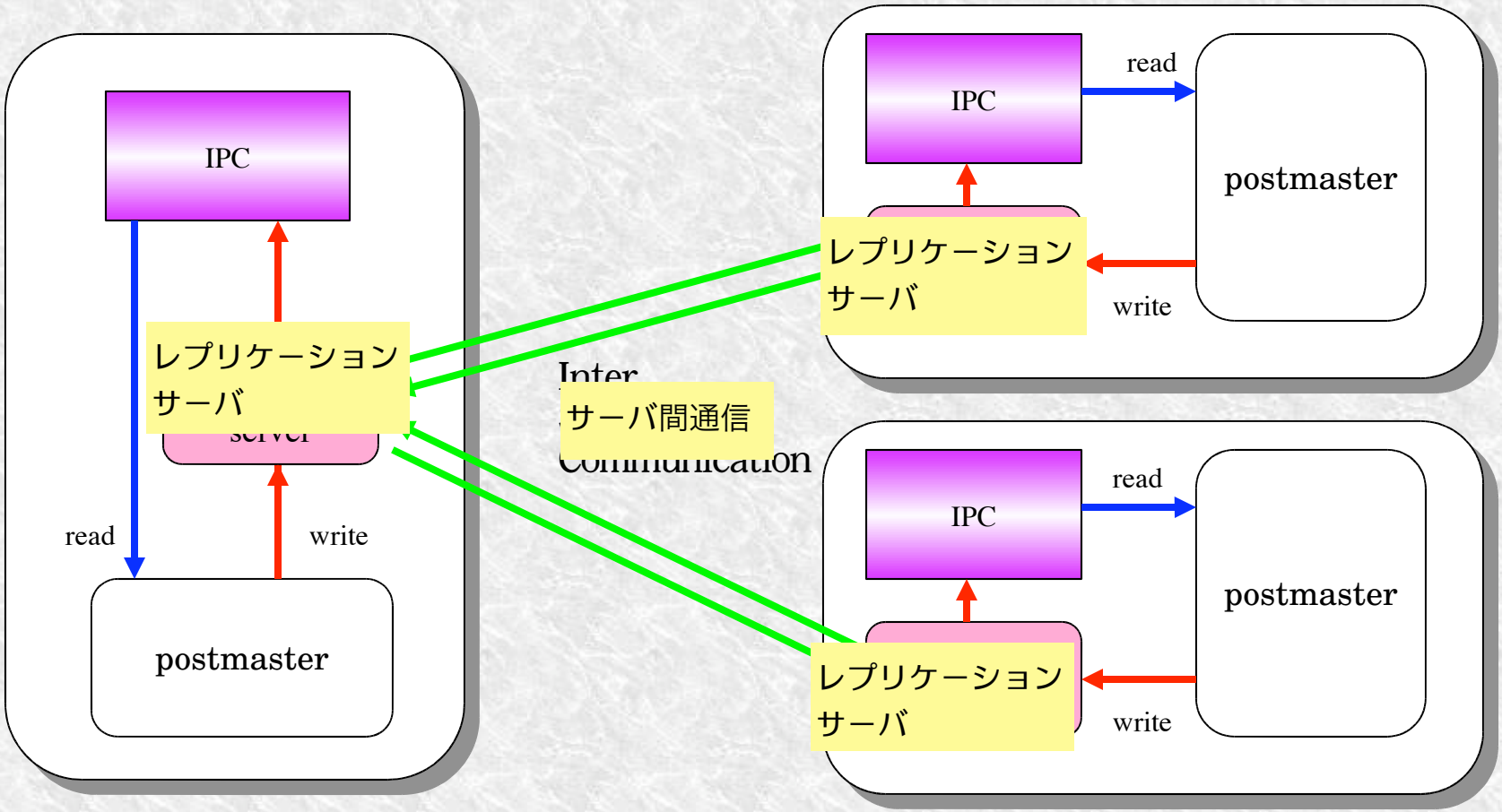


How to share IPC during distributed servers

分散サーバ間でどのようにIPCを共有するか

- 基本的にIPCはプロセス間通信の方法としては最速 ne fastest communication method during process
  - ゆえに、どのような方法で共有してもそれがオーバーヘッドとなる becomes an overhead
- 可能な限りローカルのIPCを利用する al IPC as much as possible
  - 書き込みの同期（レプリケーション） ly (replication)
    - マルチマスタ＆同期レプリケーションが必要 r & synchronous replication
  - ローカルで読み出し y

Server composition of PGCluster-II



Patch point in order to share semaphore

セマフォ共有のためのパッチポイント

Interface function	Source	Target function
PGReserveSemaphores()	storage/ipc/ipci.c	CreateSharedMemoryAndSemaphores()
PGSemaphoreCreate()	storage/lmgr/spin.c	SpinlockSemas()
PGSemaphoreReset()	storage/lmgr/proc.c	ProcCancelWaitForSignal()
		LockWaitCancel()
PGSemaphoreLock()	storage/lmgr/lwlock.c	LWLockAcquire()
	storage/lmgr/proc.c	ProcSleep()
		ProcWaitForSignal()
PGSemaphoreUnlock()	storage/lmgr/lwlock.c	LWLockAcquire()
		LWLockRelease()
	storage/lmgr/proc.c	ProcWakeup()
		CheckDeadLock()
		ProcSendSignal()
	storage/lmgr/spin.c	s_unlock_sema()
PGSemaphoreTryLock()	storage/lmgr/spin.c	tas_sema()

6 delegate functions

Called 14 functions

Patch point in order to share memory

メモリ共有のためのパッチポイント

BgWriterShmem	postmaster/bgwriter.c	BackgroundWriterMain()
		BgWriterShmemInit()
MultiXactState	access/transam/multixact.c	GetNewMultiXactId()
		MultiXactShmemInit()
		StartupMultiXact()
		MultiXactSetNextMXact()
		MultiXactAdvanceNextMXact()
		TruncateMultiXact()
TwoPhaseState	access/transam/twophase.c	TwoPhaseShmemInit()
		MarkAsPreparing()
		RemoveGXact()
ControlFile	access/transam/xlog.c	XLogWrite()
		WriteControlFile()
		ReadControlFile()
		UpdateControlFile()
		XLOGShmemInit()
		BootStrapXLOG()
		StartupXLOG()
		CreateCheckPoint()
XLogCtl	access/transam/xlog.c	XLogInsert()
		AdvanceXLInsertBuffer()
		XLogWrite()
		XLOGShmemInit()
		StartupXLOG()

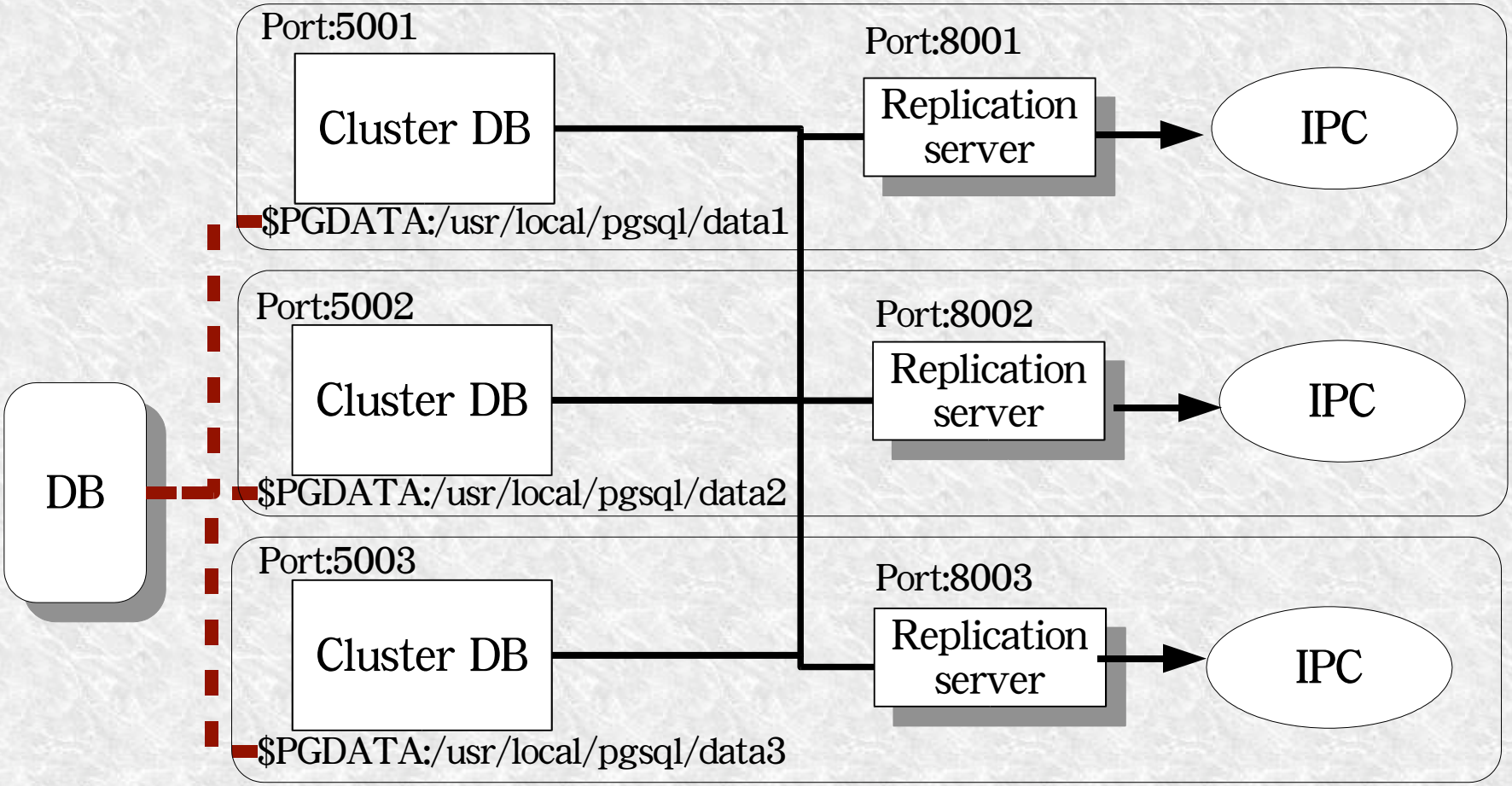
BufferDescriptors	storage/buffer/buf_init.c	InitBufferPool()
	storage/buffer/bufmgr.c	ReadBuffer()
		PinBuffer()
	storage/buffer/freelist.c	StrategyGetBuffer()
		StrategyFreeBuffer()
BufferBlocks	storage/buffer/buf_init.c	
StrategyControl	storage/buffer/freelist.c	StrategyGetBuffer()
		StrategyFreeBuffer()
		StrategyInitialize()
FreeSpaceMap	storage/freespace/freespace.c	InitFreeSpaceMap()
		delete_fsm_rel()
		realloc_fsm_rel()
		link_fsm_rel_usage()
		unlink_fsm_rel_usage()
		link_fsm_rel_storage()
		unlink_fsm_rel_storage()
		compact_fsm_storage()
		push_fsm_rels_after()
PMSignalFlags	storage/ipc/pmsignal.c	PMSignalInit()
		SendPostmasterSignal()
		CheckPostmasterSignal()
procArray	storage/ipc/procarray.c	CreateSharedProcArray()
		ProcArrayAdd()
		ProcArrayRemove()

shmInvalBuffer	storage/ipc/sinvaladt.c	SIBufferInit()
newLockMethod	storage/lmgr/lock.c	LockMethodTableInit()
ProcGlobal	storage/lmgr/proc.c	InitProcGlobal()
		ProcKill()
		DummyProcKill()
DummyProcs	storage/lmgr/proc.c	InitProcGlobal()

15 pointers

Called from 50 functions

Demonstration of PGCluster-II



Expected effect and problem

期待される効果と問題点

- 効果 fect
  - 書き込みの負荷分散 ce improvement of write load
- 問題点 blem
  - 共有IPCのオーバーヘッド the shared IPC
  - ハードウェアコストの上昇 rdware cost
    - ストレージ（NAS/SAN） / SAN)
    - ネットワーク（iSCSI/FC） / FC)

Discussion

質疑応答

Discus質疑応答（エンタープライズ規模での利用）se usage)

- いかにしてパフォーマンスを改善するか he performance
  - データサイズが巨大な場合 e of datas
  - アクセスが多い場合（serach/update/insert） ch/update/insert)
- いかにして可用性を確保するか the availability
- いかにして短時間で巨大なデータをバックアップすることができるか ta in a short time
- どうやってこれら全てを実現するか all of them

Discuss (PGCluster)

質疑応答（PGCluster）

- サードパーティとして貢献するには te as 3rd party
- 共有/非共有 l / Non shared
  - どちらがより重要なのか e important
- PGClusterに必要なのはなにか ed for PGCluster

Thank you 謝辞

- When you have a
  question ‖ issue ‖ requirement
  for PGCluster, would you please send email.
  - mitani@sraw.co.jp
  - pgcluster-general@pgfoundry.org
- You can download all version of PGCluster from following site,
  - http://pgfoundry.org/projects/pgcluster/