大規模システムにおけるユーザ認証
- OpenLDAP を用いた認証サーバ構築時の注意点 -

2004 年 12月2日

VA Linux Systems Japan, K.K.
樽石 将人

目次

VA LINUX SYSTEMS JAPAN

- 認証サーバと OpenLDAP
- 大規模システムにおける OpenLDAP
- UltraPossum 概要/導入
- 大規模システムでの注意点

VA LINUX SYSTEMS JAPAN

1. 認証サーバと OpenLDAP

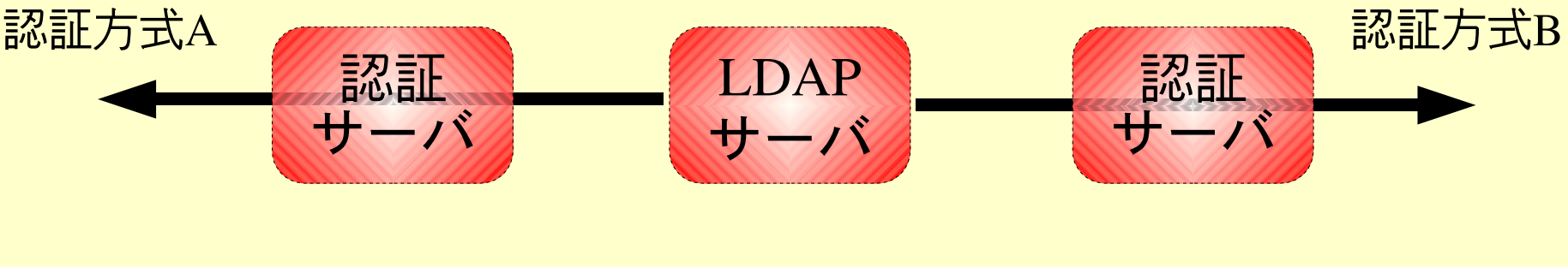
1.1 認証

VA LINUX SYSTEMS JAPAN

- 多数の認証方式 (本人確認)
  - ユーザ/パスワード
  - PKI/Kerberos
  - 顔/指紋
- アクセスコントロール (権限確認)
  - RBAC (Role-Based Access Control)
  - POSIX アクセス権限

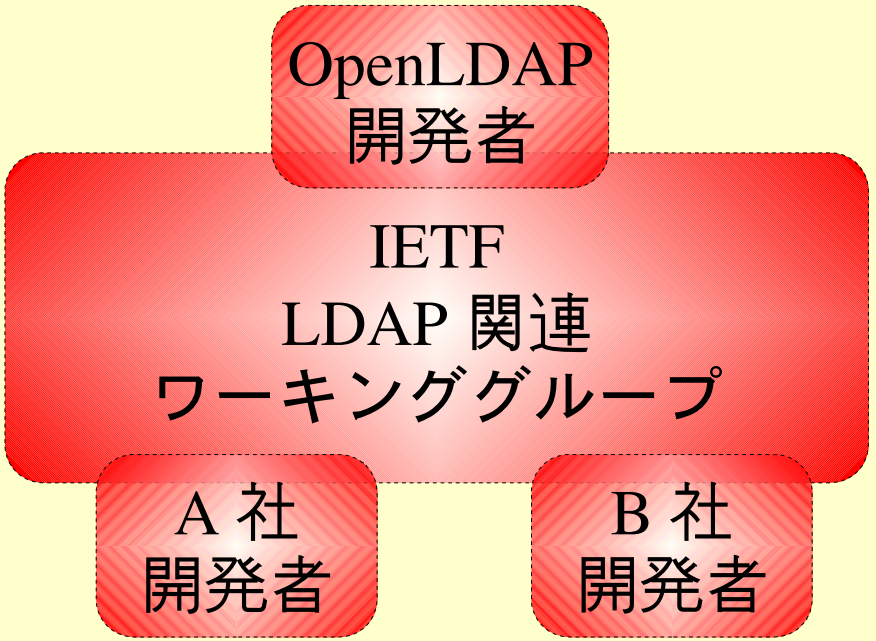
1.2 認証情報の保存先

- 認証には認証情報が必要
  - 一元管理する事でコストの削減
- LDAP
  - 一元管理のための標準プロトコル
  - 1997 年に LDAPv3 が RFC で提案
  - 現在も活発にプロトコルの拡張を継続



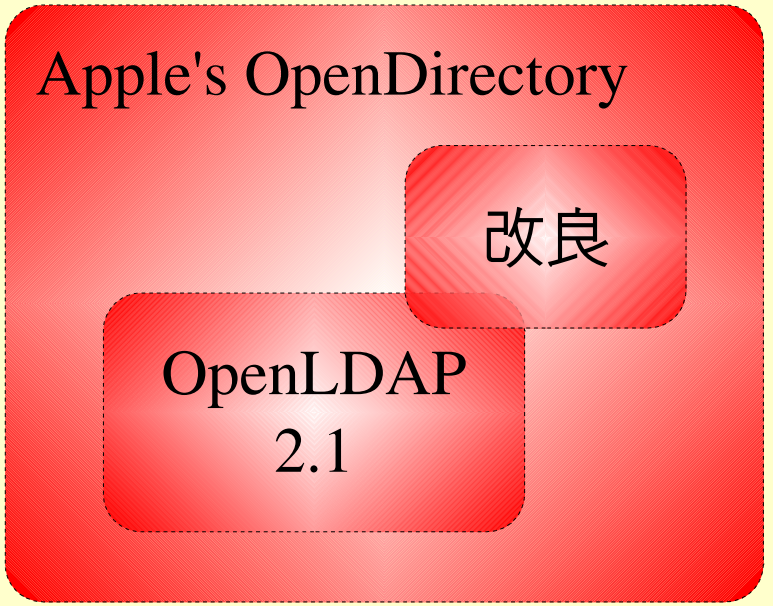
1.3 OpenLDAP

- オープンソースの LDAP サーバ
  - 標準との密接な関わり
    - IETF LDAP 関連ワーキンググループの議長



1.4 Mac OS X での採用

- コアコンポーネントに OpenLDAP を採用
  - http://developer.apple.com/darwin/projects/opendirectory/



1.5 まとめ

VA LINUX SYSTEMS JAPAN

- LDAP で認証/権限情報の一元管理
  - 標準化されたインターネットプロトコル
- OpenLDAP は標準的な LDAP サーバである
  - OpenLDAP のリーダが IETF LDAP ワーキンググループの議長
  - 製品レベルでの採用

2．大規模システムにおける OpenLDAP

2.1 大規模システムに必要とされる主な要件

VA LINUX SYSTEMS JAPAN

- 高性能
  - 大量のアクセス
- 高可用性
  - 停止時の損失が膨大

2.2 高性能

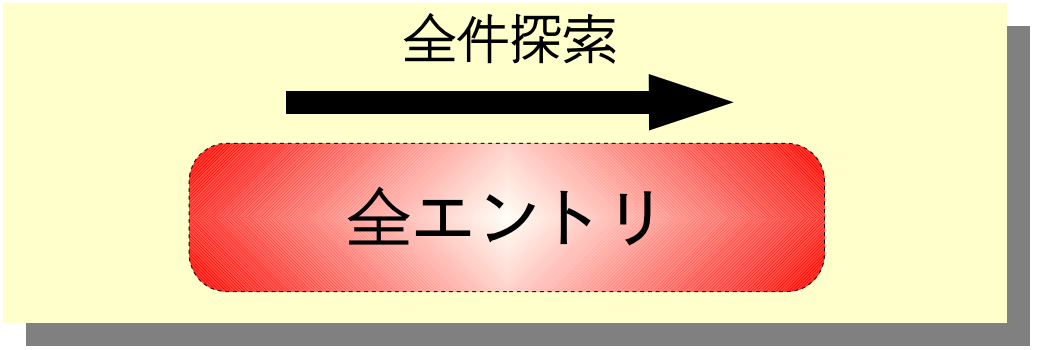
VA LINUX SYSTEMS JAPAN

- 処理性能をあげる
  - ハードウェアスペックをあげる
- 処理単価を減らす
  - 計算量を減らす(チューニング)
- 負荷を分散する(複数のサーバを用意)

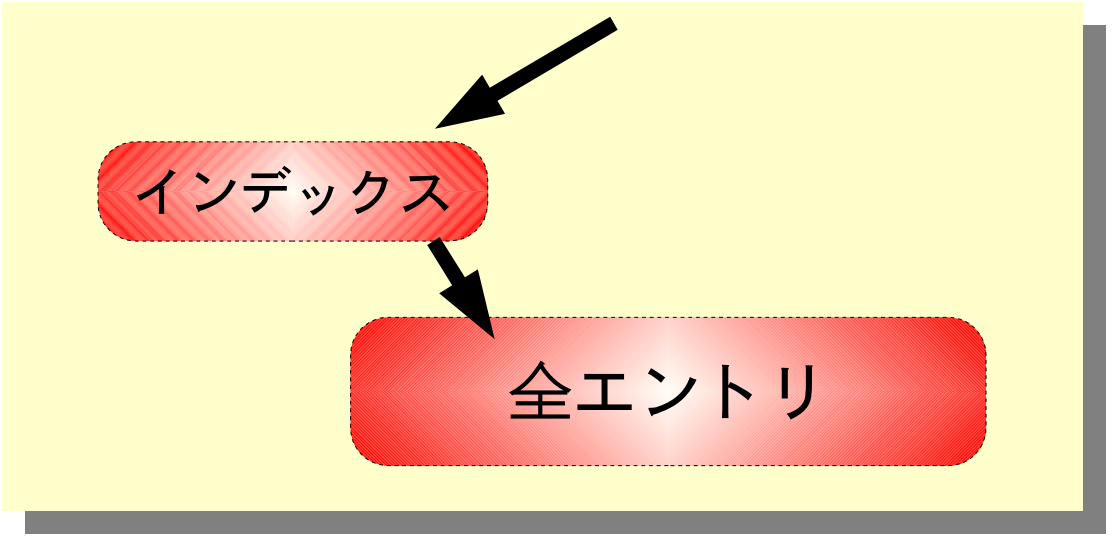
2.3 処理単価を減らす

VA LINUX SYSTEMS JAPAN

- 検索インデックスを作成する
  - インデックスがない場合は全件探索を行う



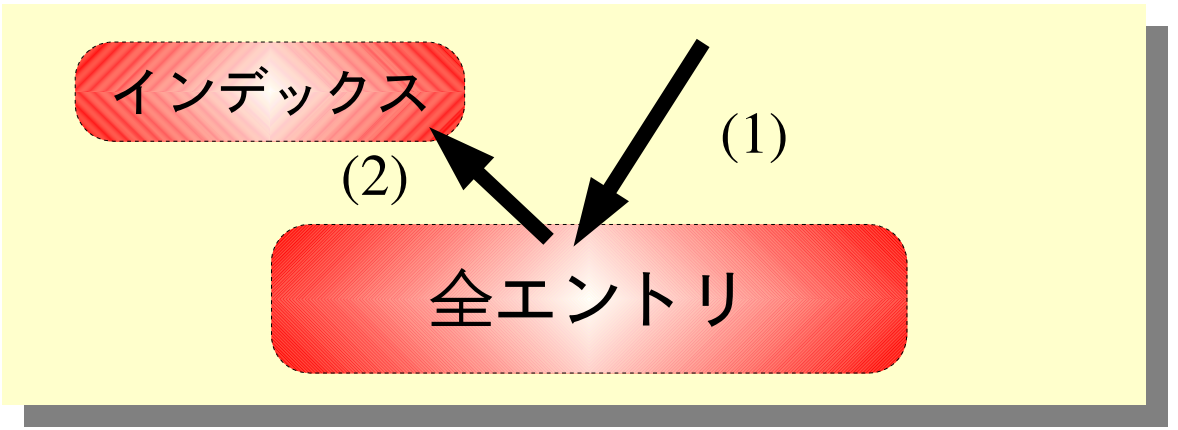
  - インデックスを利用することで全件探索を排除



2.4 インデックス生成の更新負荷

VA LINUX SYSTEMS JAPAN

- ディレクトリ更新時にインデックスも生成



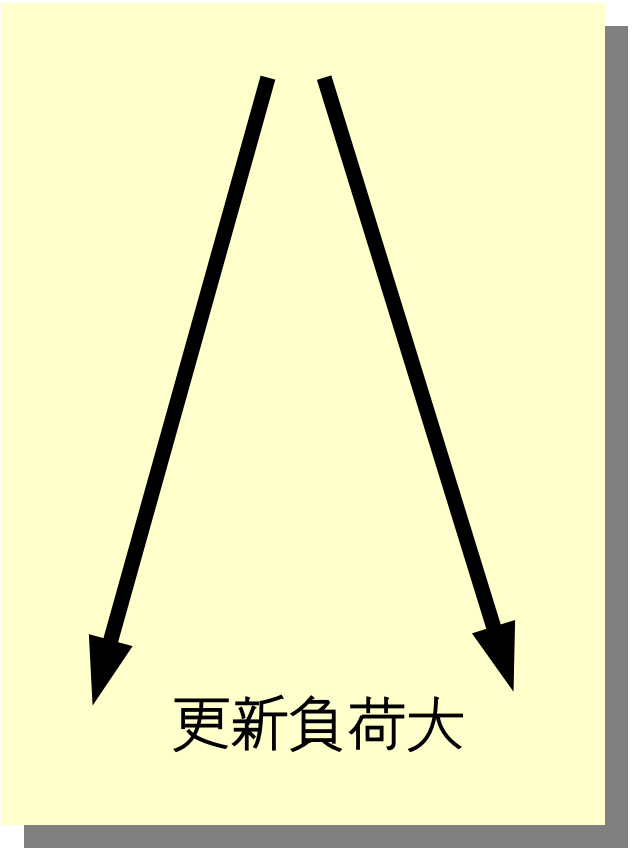
  - 不用意にインデックスを生成すると更新性能が大幅に劣化
  - 適切なインデックス設定が必要

2.5 インデックスの設定

- http://www.openldap.org/doc/admin22/slapdconfig.html

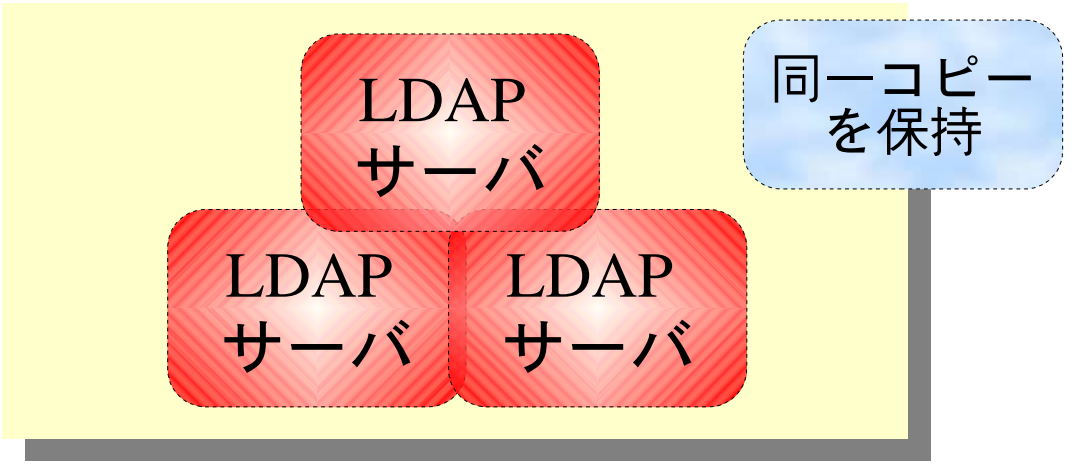
index {<attrlist> | default} [pres,eq,approx,sub,none]

  - 存在インデックス
    - foo=*
  - 同値インデックス
    - foo=bar
  - 部分一致インデックス
    - foo=*bar*
  - あいまいインデックス
    - foo~=bar

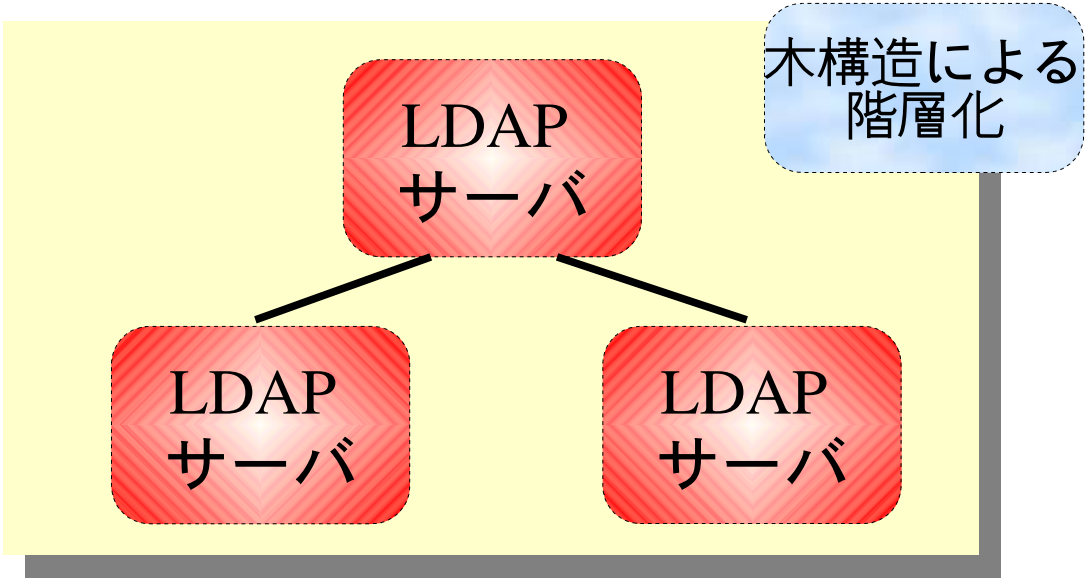


2.6 負荷分散

- レプリケーション(複製)による分散

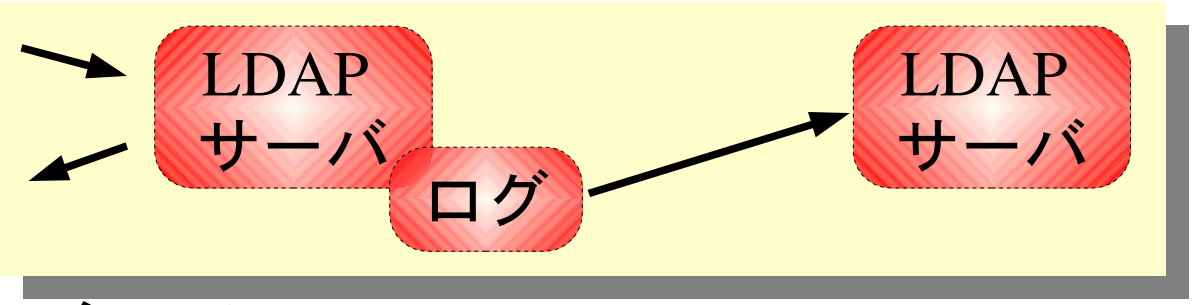


階層化による分散



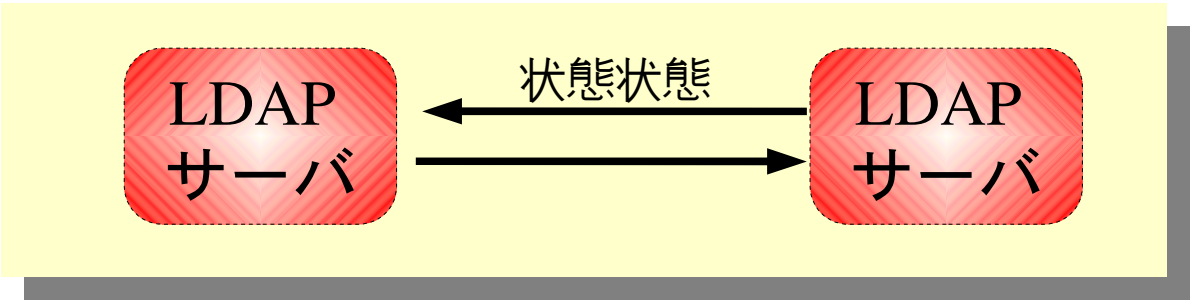
2.7 OpenLDAP の複製方式

- ログスタイルレプリケーション (slurpd)



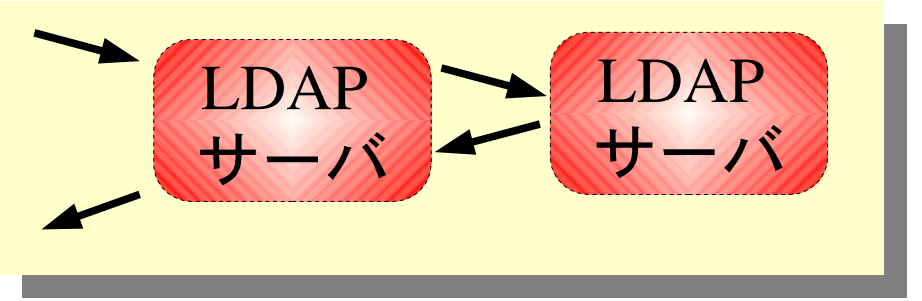
状態スタイルレプリケーション (syncrepl)

OpenLDAP 2.2 以降



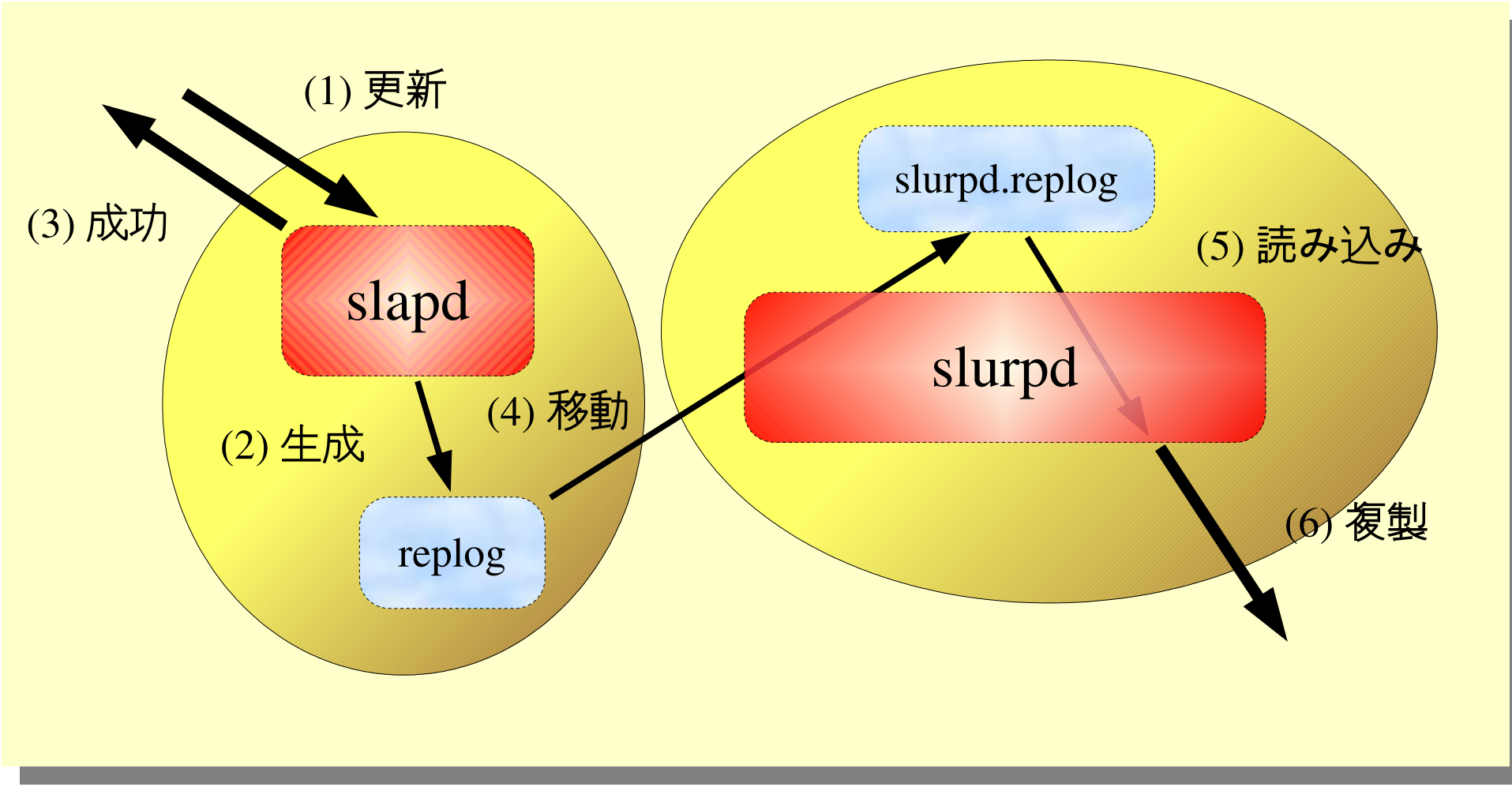
同期式レプリケーション (syncbackup)

OpenLDAP 2.2 以降 (2.2.14/17/18)
パッチによる配布



2.8 slurpd レプリケーション

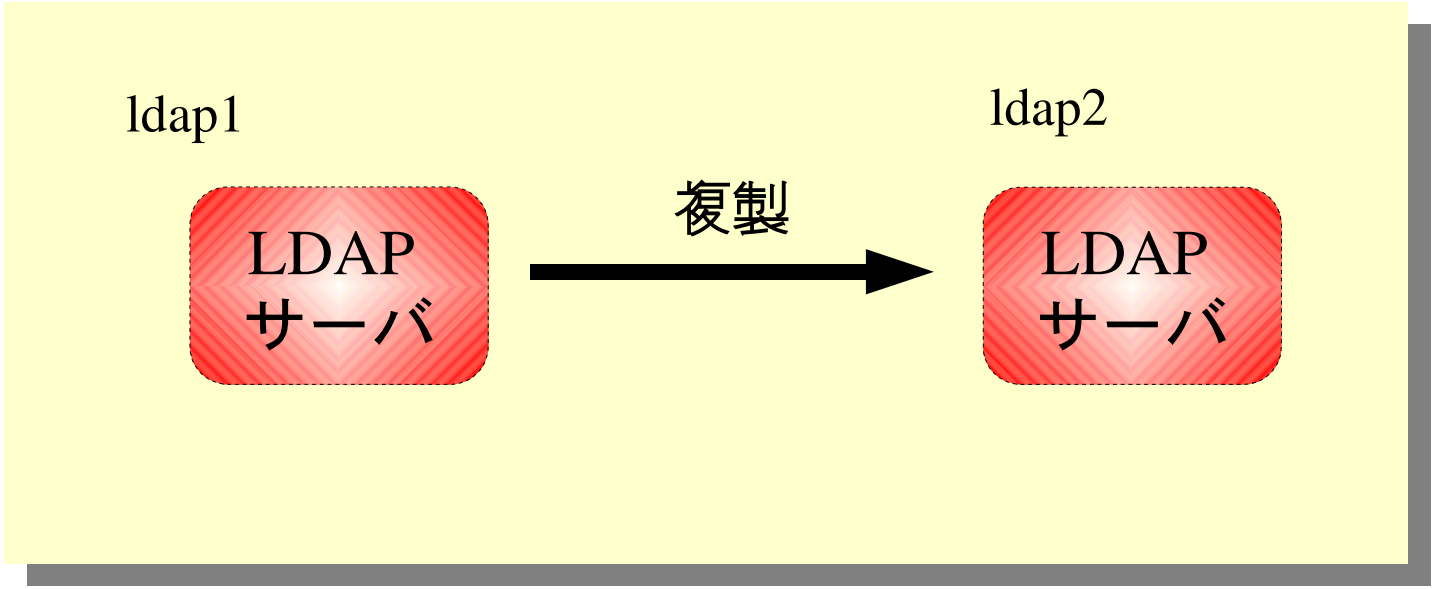
- slapd と slurpd の二つのデーモンによる複製



2.9 slurpd によるレプリケーションの例

VA LINUX SYSTEMS JAPAN

- 以下の環境を構築する
  - マスタサーバ ldap1, スレーブサーバ ldap2



2.10 slurpd 用の slapd.conf 設定

VA LINUX SYSTEMS JAPAN

- マスタサーバの設定

```
replica uri=ldap://ldap2
        bindmethod=simple
        binddn=cn=replica,ou=application,dc=ultrapossum,dc=org
        credentials=password
```

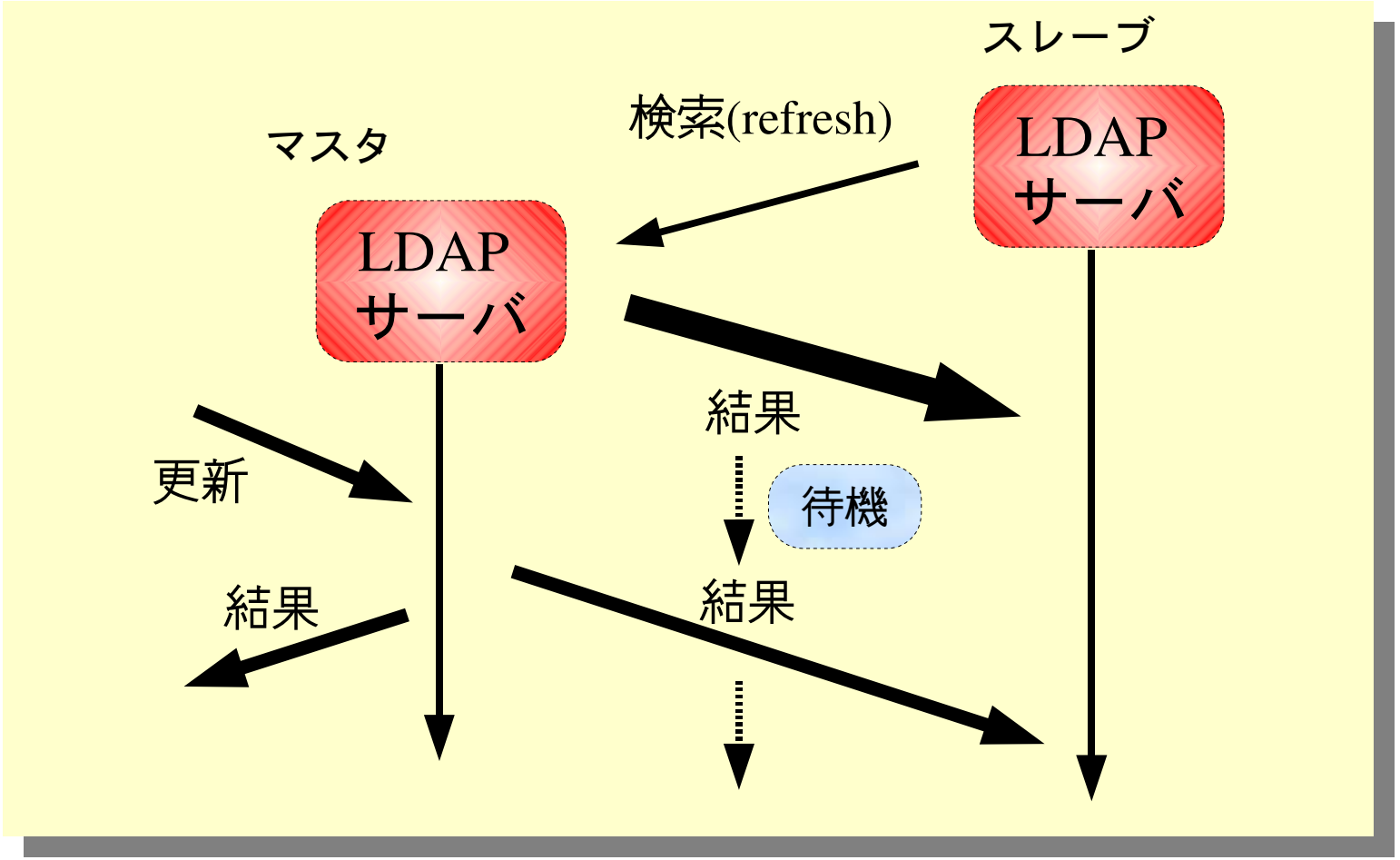
- スレーブサーバの設定

```
rootdn cn=replica,ou=application,dc=ultrapossum,dc=org
rootpw password

updatedn cn=replica,ou=application,dc=ultrapossum,dc=org
```

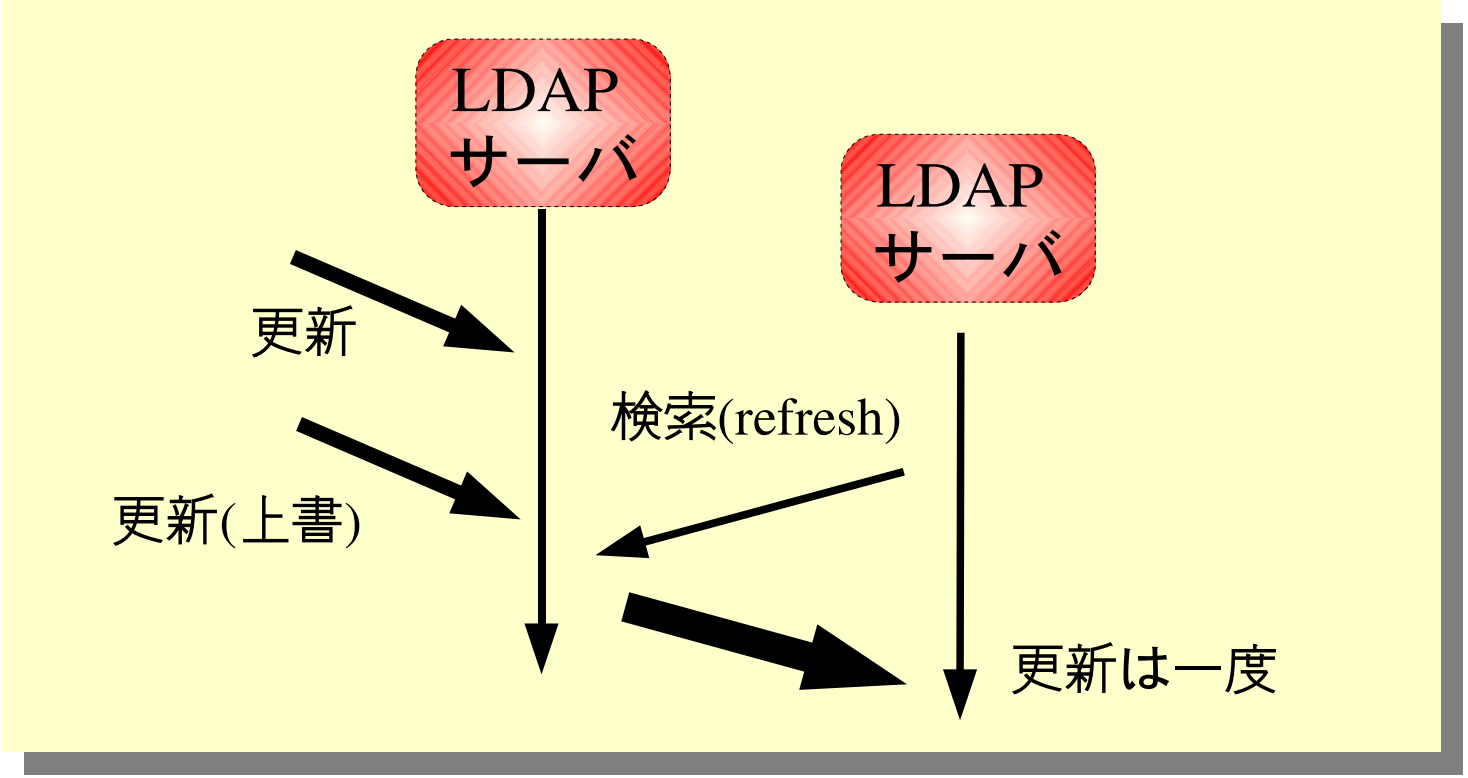
2.11 syncrepl レプリケーション

- スレーブサーバから複製開始を要求
  - 検索プロトコルの拡張
    - 検索権限あれば誰でもスレーブになれる



2.12 状態スタイルレプリケーション?

- ディレクトリの現在の状態を取得する

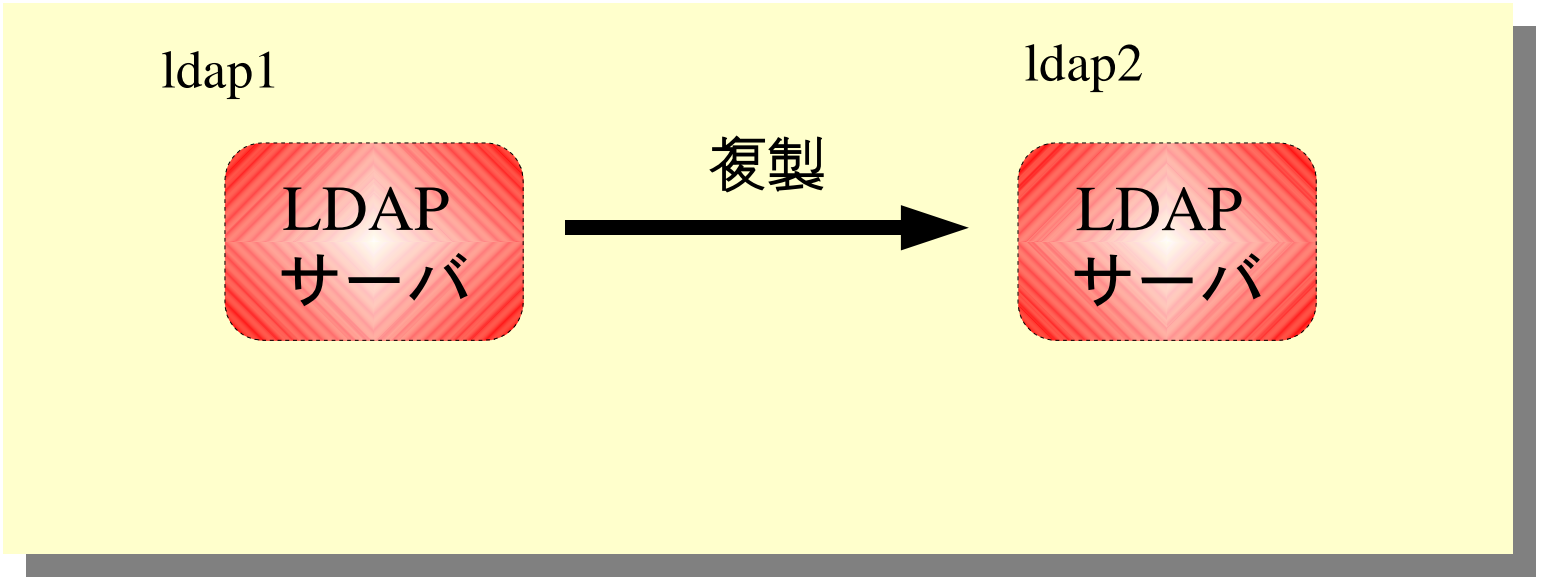


  - 同じエントリを何度更新しても複製は一回
  - ログが不要(実際に使うディレクトリがあれば良い)
    - 管理コストの削減
    - 動的追加が容易
    - refresh に長い時間かかることがある

2.13 syncrepl によるレプリケーションの例

VA LINUX SYSTEMS JAPAN

- 以下の環境を構築する
  - マスタサーバ ldap1, スレーブサーバ ldap2



2.14 syncrepl 用の slapd.conf 設定

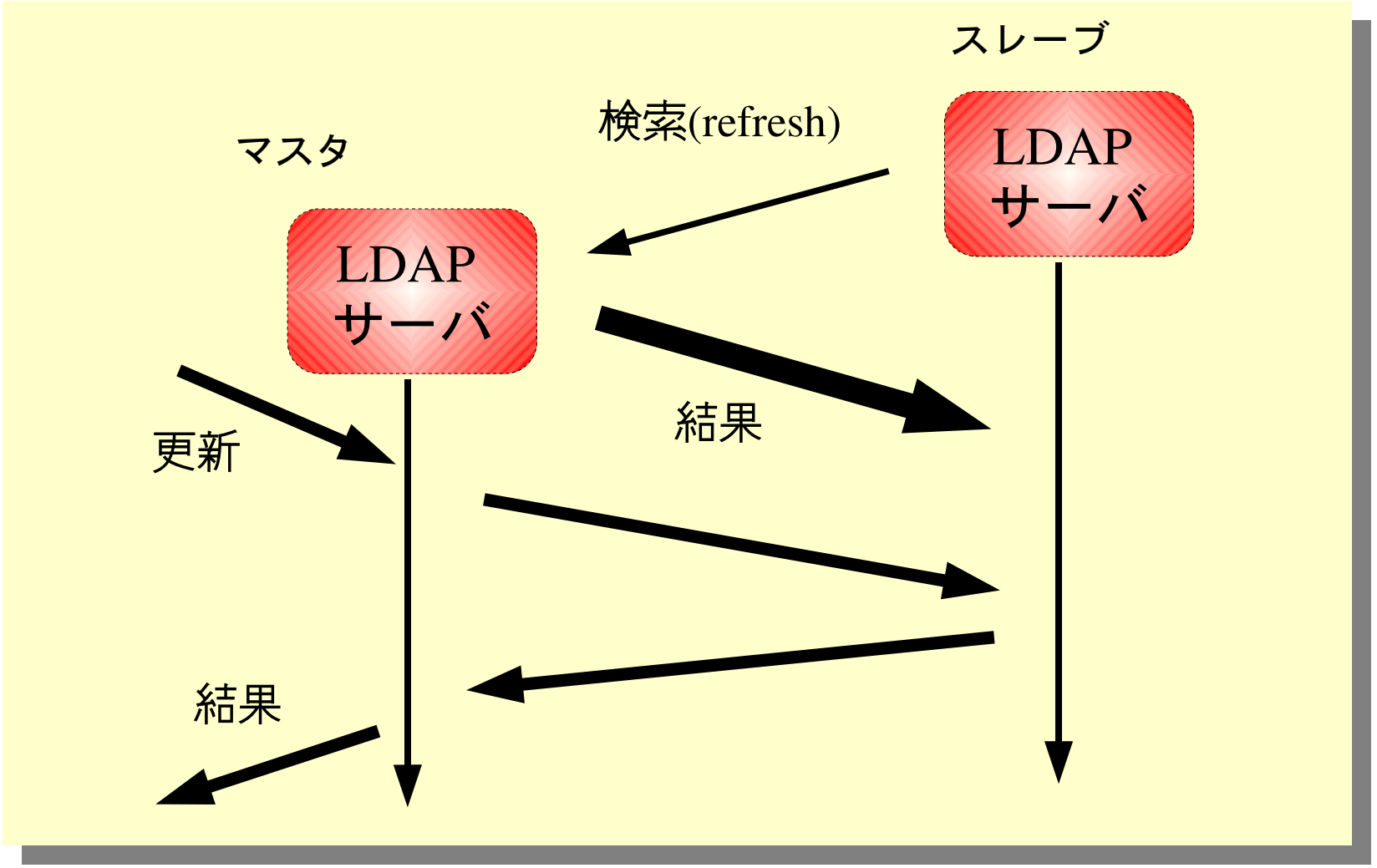
VA LINUX SYSTEMS JAPAN

- スレーブ設定

```
syncrepl  rid=0
provider=ldap://taru/
        type=refreshAndPersist
        retry=10,+
        searchbase=dc=ultrapossum,dc=org
        updatedn=cn=replica,ou=application,dc=ultrapossum,dc=org
        binddn="cn=replica,ou=application,dc=ultrapossum,dc=org"
        bindmethod=simple
        credentials=password
```

2.15 syncbackup（同期式レプリケーション）

- syncrepl をベースに機能を追加
- スレーブへの複製が完了してから結果を返す



2.16 syncbackup の導入

- http://openldap-ha.sourceforge.net/
- http://openldap-ha.sourceforge.net/syncbackup.html.ja
- 不安定
- パッチは CVS

```
~$ CVSROOT=:pserver:anonymous@cvs.sourceforge.net:/cvsroot/openldap-ha
~$ export CVSROOT
~$ cvs login
Logging in to :pserver:anonymous@cvs.sourceforge.net:2401/cvsroot/openldap-ha
CVS password:
cvs login: warning: failed to open /home/taru/.cvspass for reading: No such file or directory
~$ cvs co .
```

```
~$ cd openldap-VERSION
~$ patch -p1 < パッチのパス/syncbackup-VERSION.patch
~$ ./configure
....
```

2.17 syncbackup 用の slapd.conf 設定

- マスタの設定

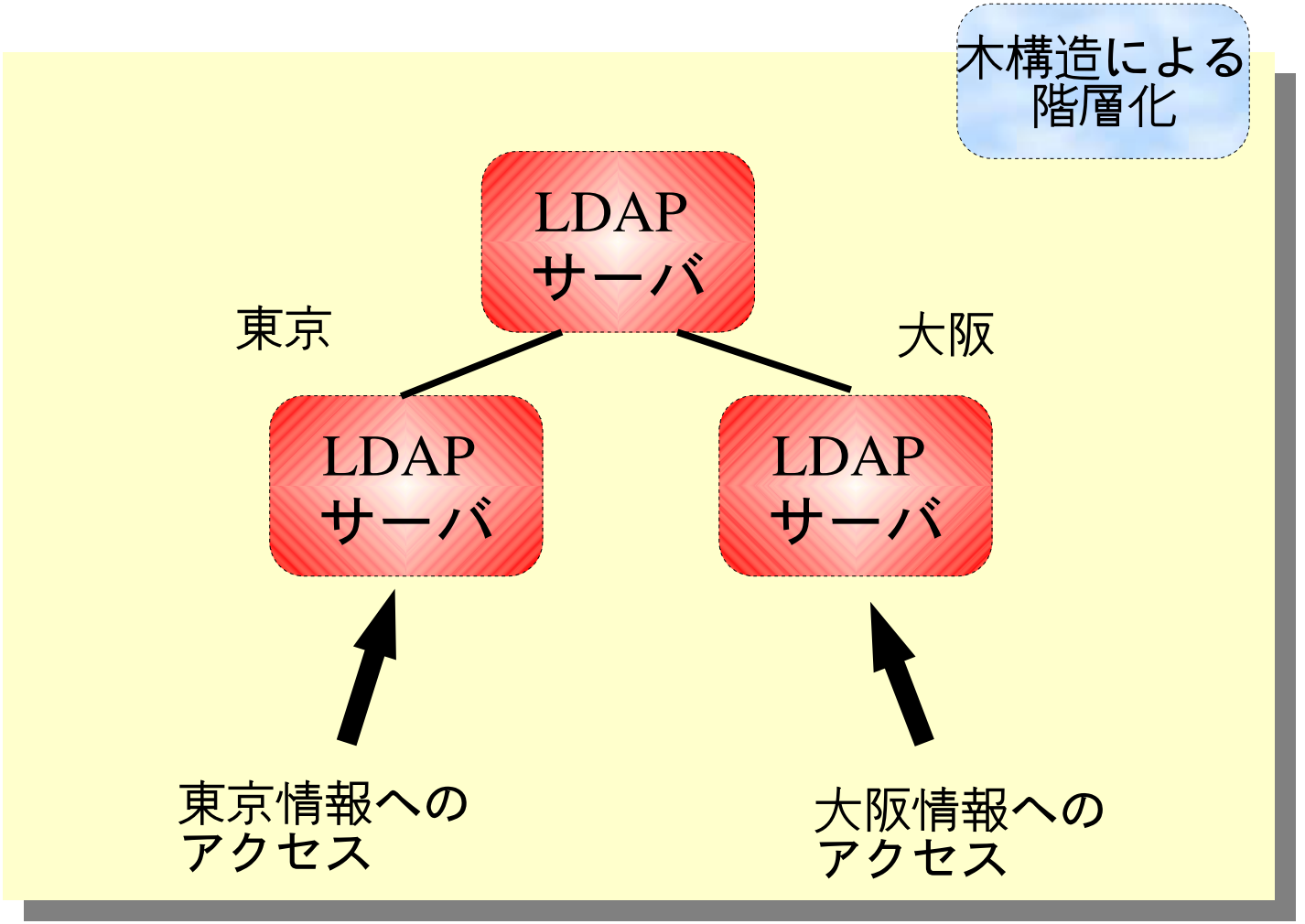
```
syncdn    "cn=admin,ou=application,dc=ultrapossum,dc=org"
replica host=ldap2
            syncid=ldap2
            bindmethod=simple
            binddn=cn=replica,ou=application,dc=ultrapossum,dc=org
            credentials=password
 weaksync on
```

- スレーブサーバの設定

```
updatedn    "cn=replica,ou=application,dc=ultrapossum,dc=org"
syncbackup syncid=ldap2
     provider=ldap://ldap1/
     binddn="cn=admin,ou=applicatin,dc=ultrapossum,dc=org"
     bindmethod=simple
     credentials=password
     checkinterval=10
updateref   "ldap://ldap1/"
```

2.18 階層化による分散

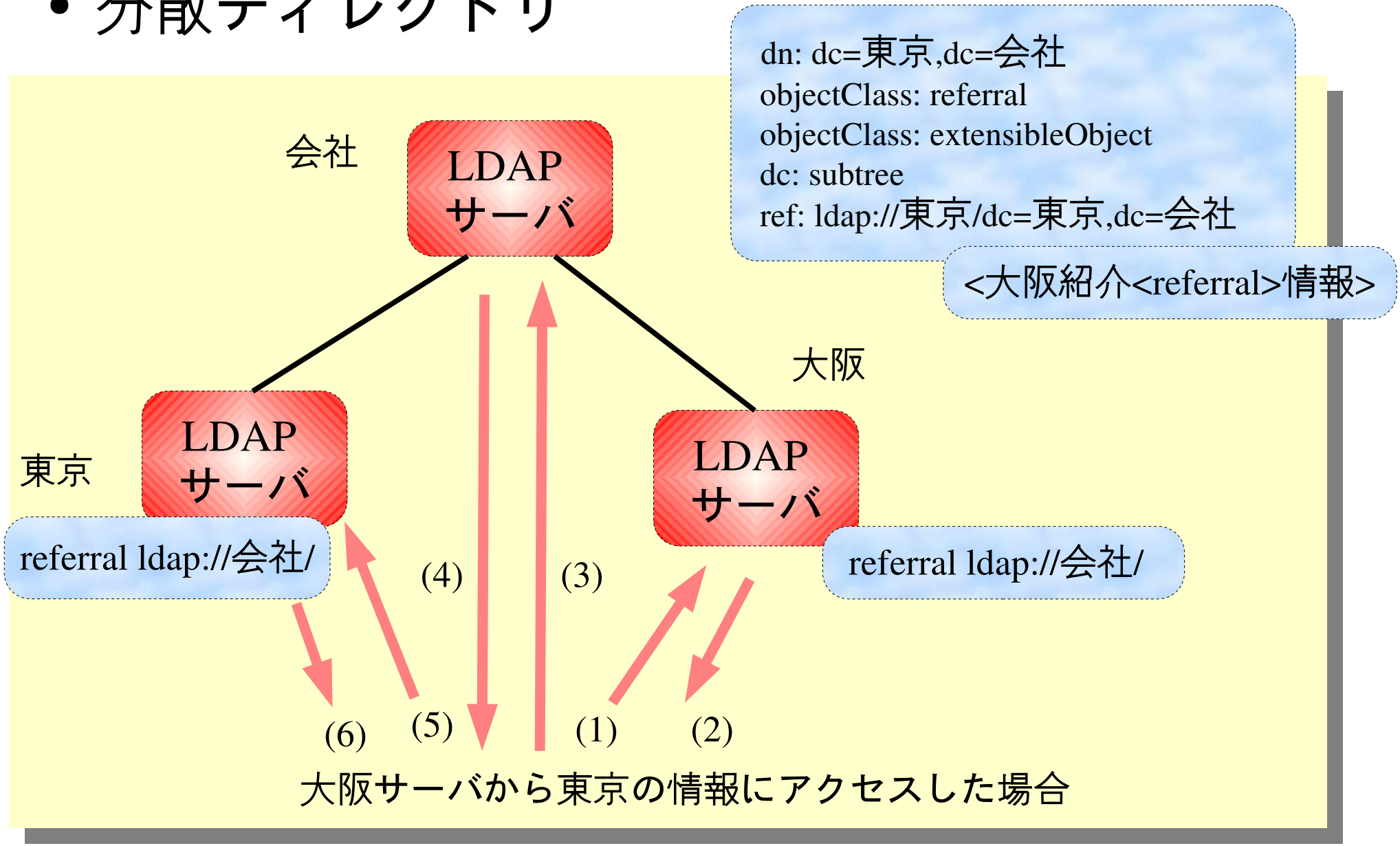
- 分散ディレクトリ
- エントリの更新サービスも分散可能



2.19 分散ディレクトリ用の slapd.conf 設定

VA LINUX SYSTEMS JAPAN

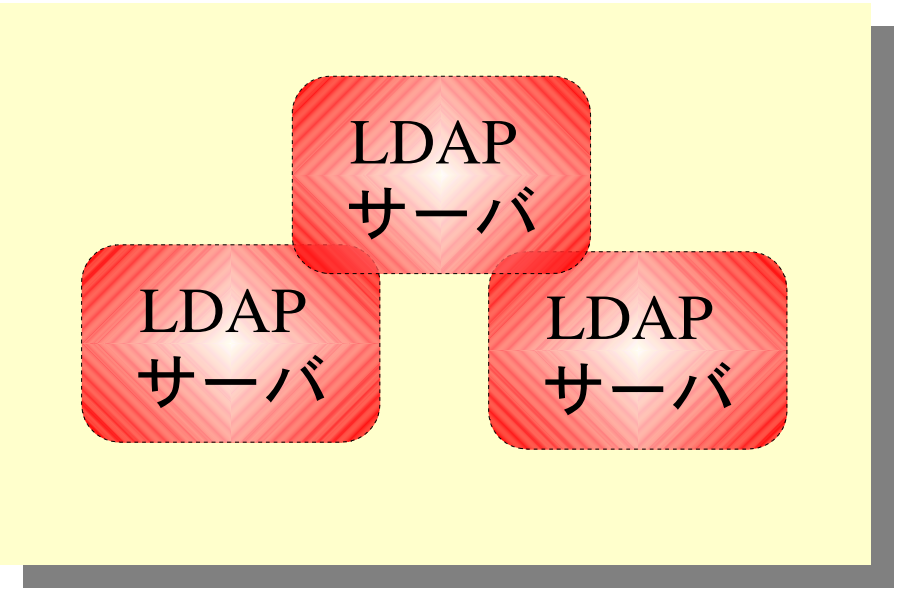
- 分散ディレクトリ



2.20 高可用性（High-availability）

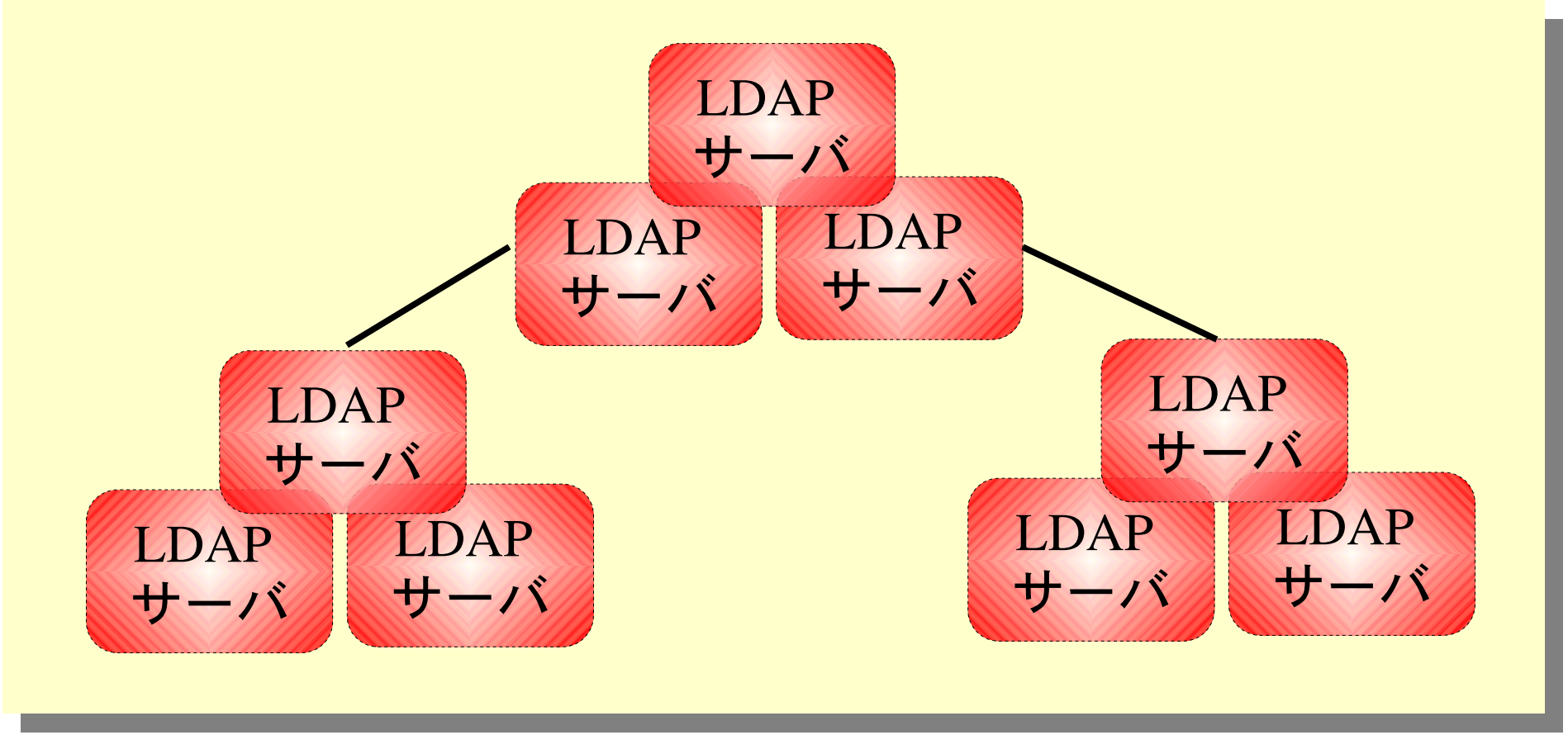
VA LINUX SYSTEMS JAPAN

- 高いサービス稼働率
  - LDAP では検索サービス/更新サービスに分けて考える
- 複数のサーバによる冗長化



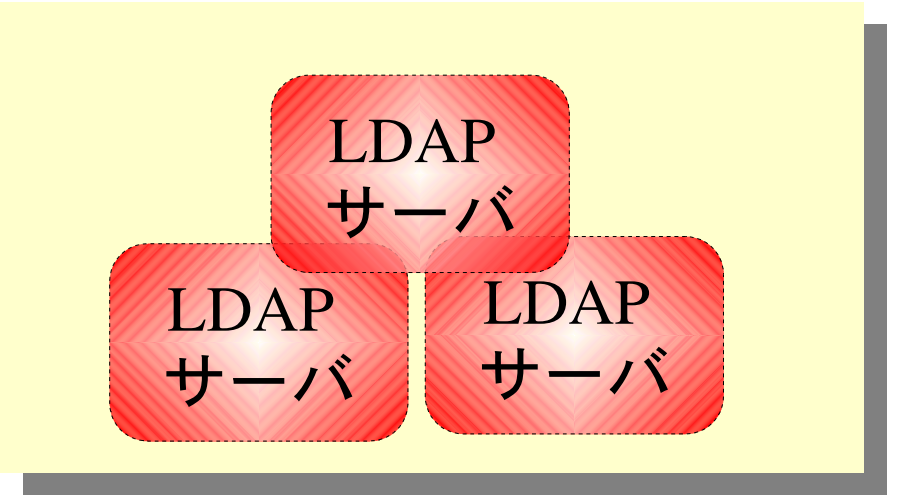
2.21 検索サービスの高可用性

- 複製を用意することで高可用性の実現
  - 分散ディレクトリの場合、分散単位毎に複製が必要

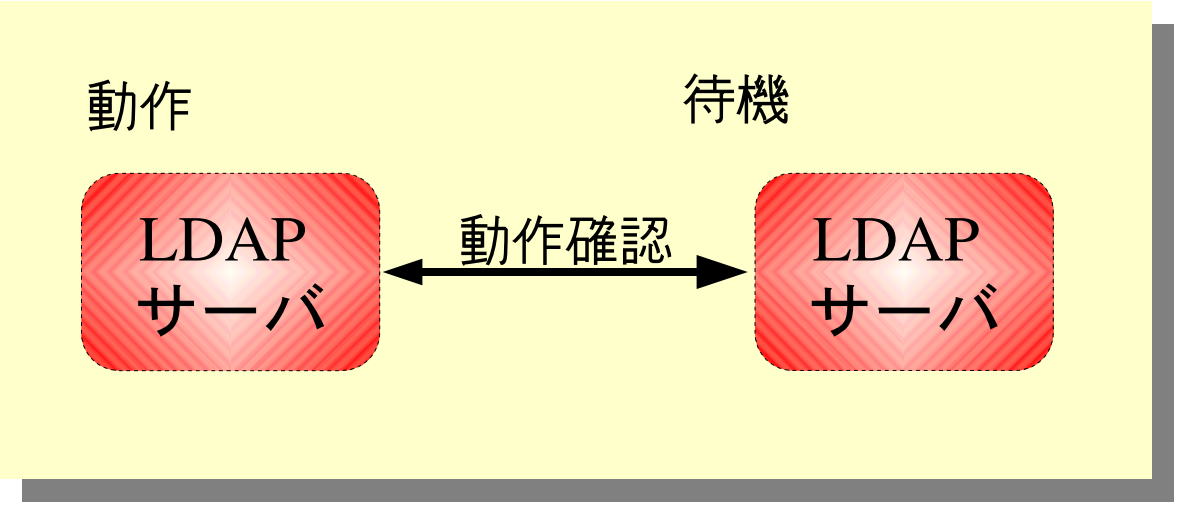


2.22 更新サービスの高可用性

- 更新サービスの冗長化 (マルチマスタ)
  - 整合性問題

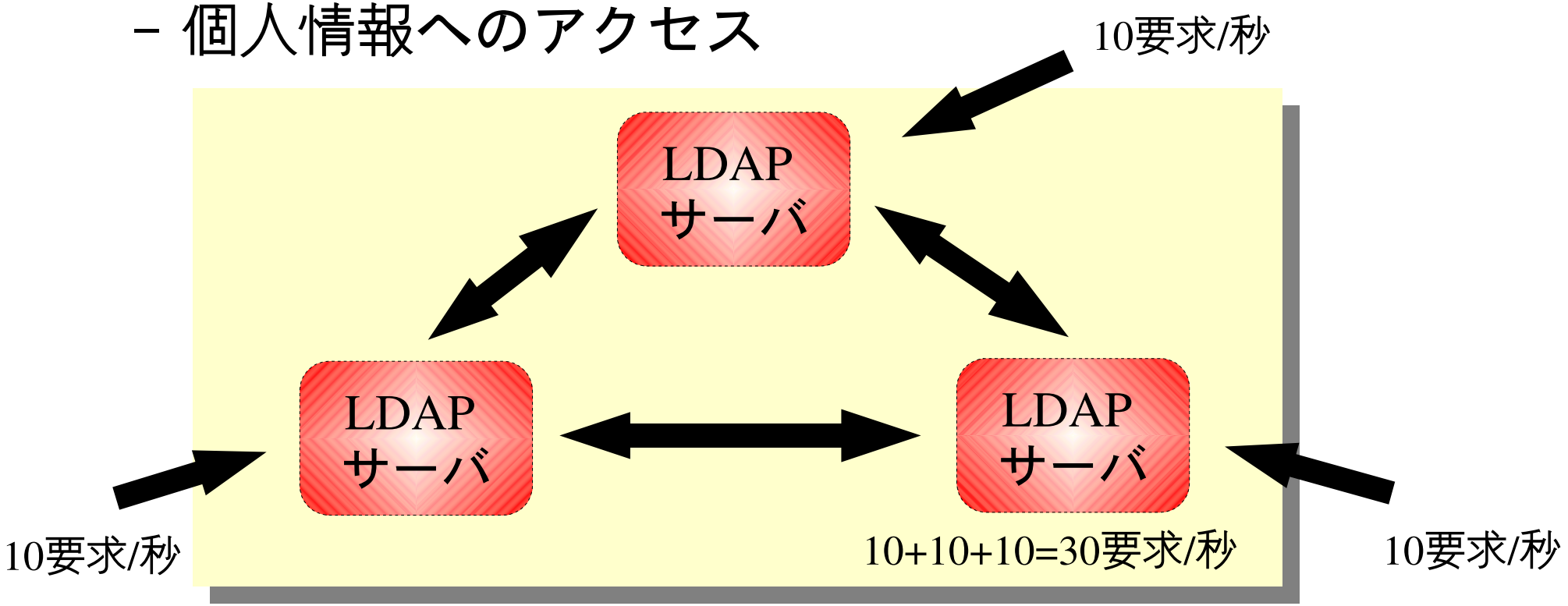


ファイルオーバによるサービス引き継ぎ



2.23 マルチマスタ

- 一時的な高負荷に対応
- 不整合情報の取扱い
  - 個人情報へのアクセス

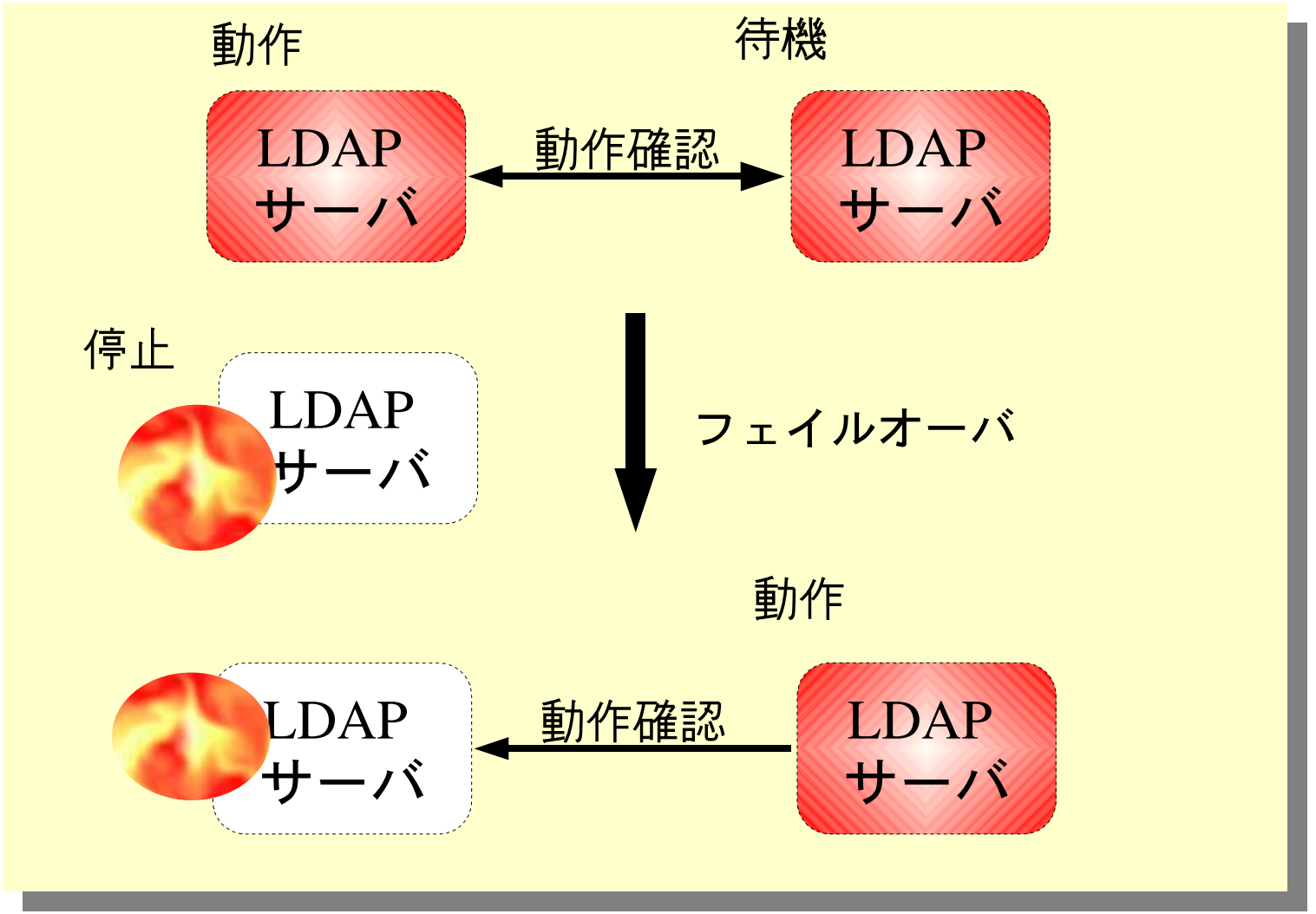


レプリケーションを後回しにすれば
高負荷時の更新受付数を増やせる

不整合発生の可能性があがる
(管理者による修正)

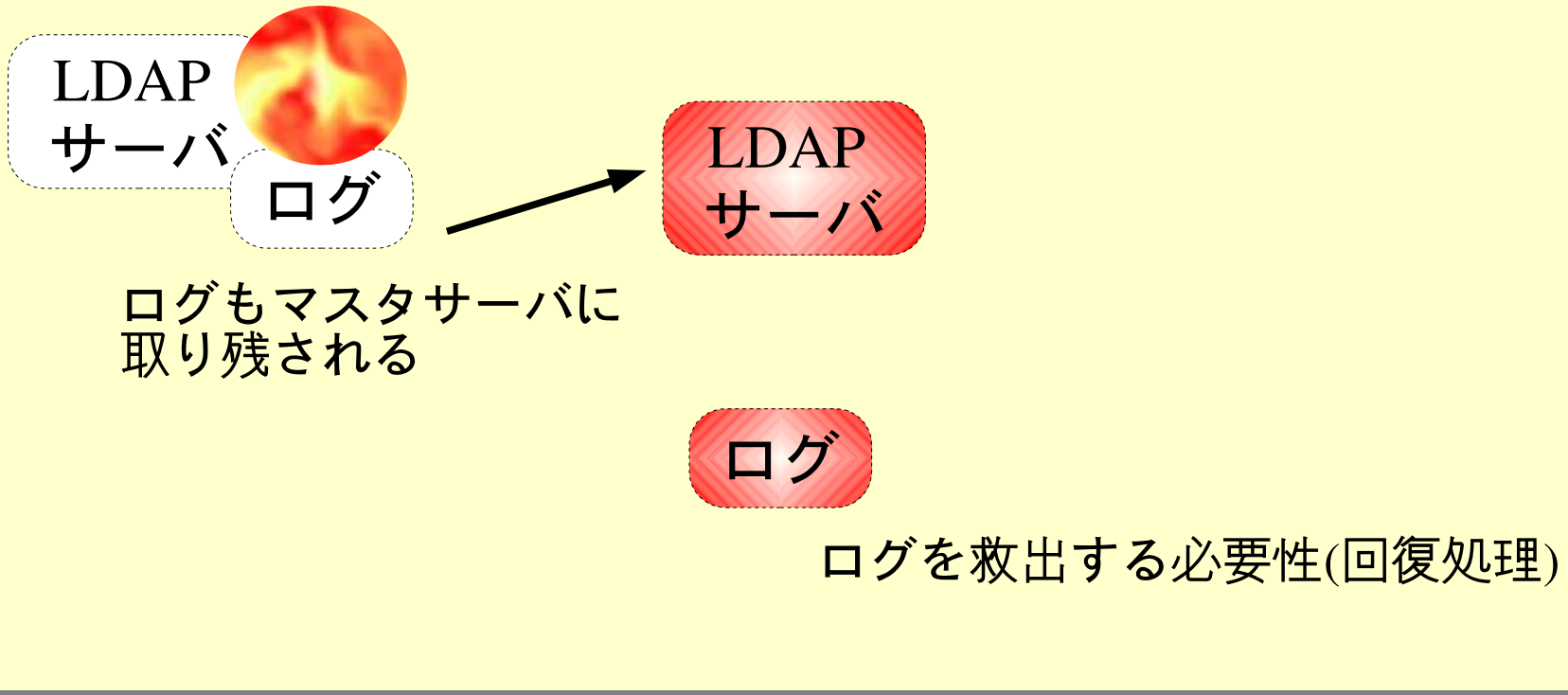
2.24 フェイルオーバ

- 更新サーバ停止時に待機サーバがサービスを引き継ぐ



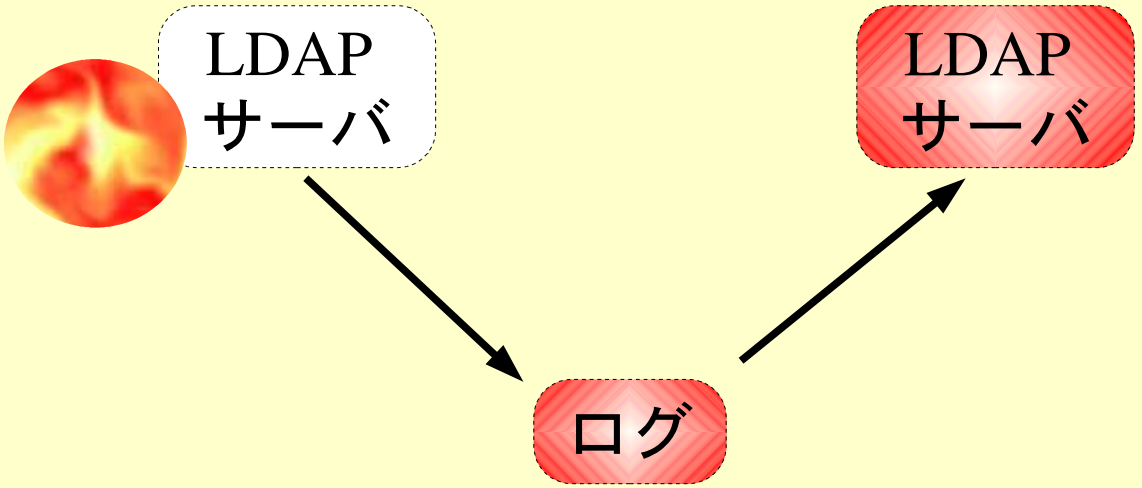
2.25 引き継ぎ時の問題

- 更新情報の一貫性
  - 回復処理が必要



2.26 slurpd 方式の場合

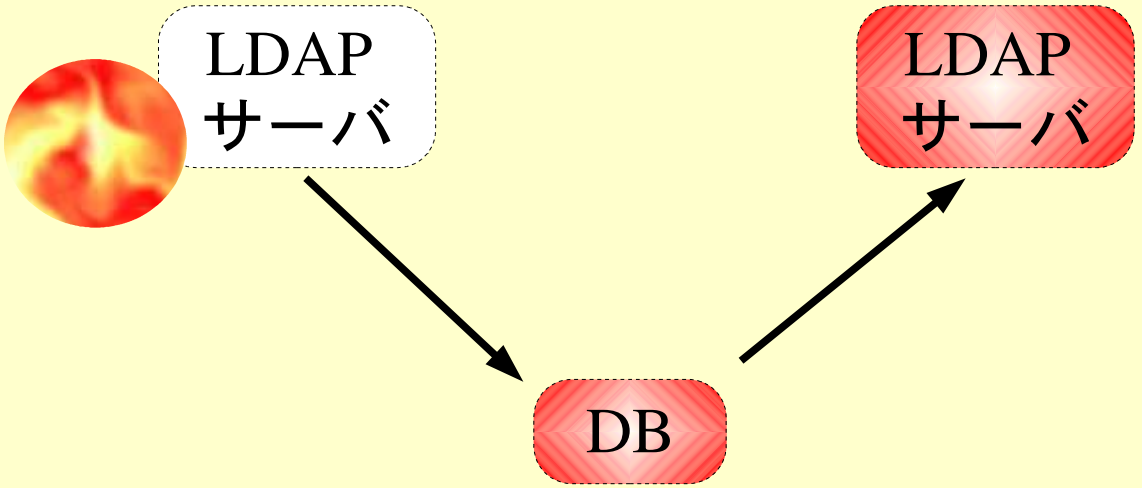
- レプリケーションログを共有する
  - NFS/共有ディスク



UltraPossum は NFS を用いたログ共有に対応

2.27 syncrepl 方式の場合

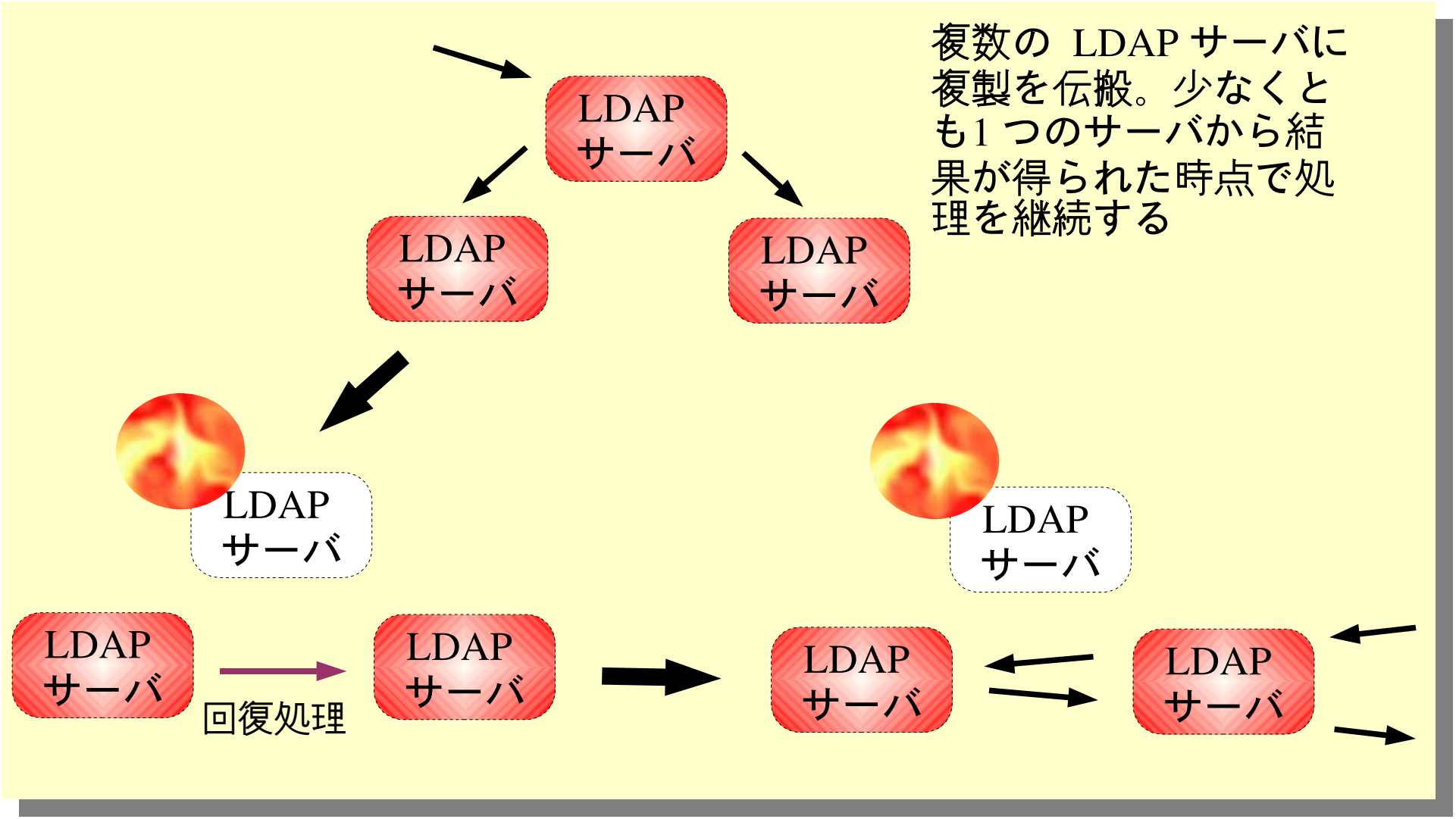
- DB 全体を共有する
  - 共有ディスク



共有ディスクの排他制御を確実に行う
同時にアクセスするとファイルシステムが破壊される

2.28 syncbackup 方式の場合

- プロトコルによる自動回復



2.29 まとめ

VA LINUX SYSTEMS JAPAN

- 3つのレプリケーションモデルの比較

レプリケーション モデル	slurpd	syncrepl	syncbackup
特徴	ログ	状態	状態+同期
方式安全性	×	○	◎
管理コスト	×	○	○
コード安定性	◎	△	×
検索分散化適用性	◎	◎	◎
更新冗長化適用性	○(ログの共有)	△(DB 全体の共有)	◎
更新性能	◎	○	△
大規模適用性	◎	○(同期負荷大)	○(同期負荷大)

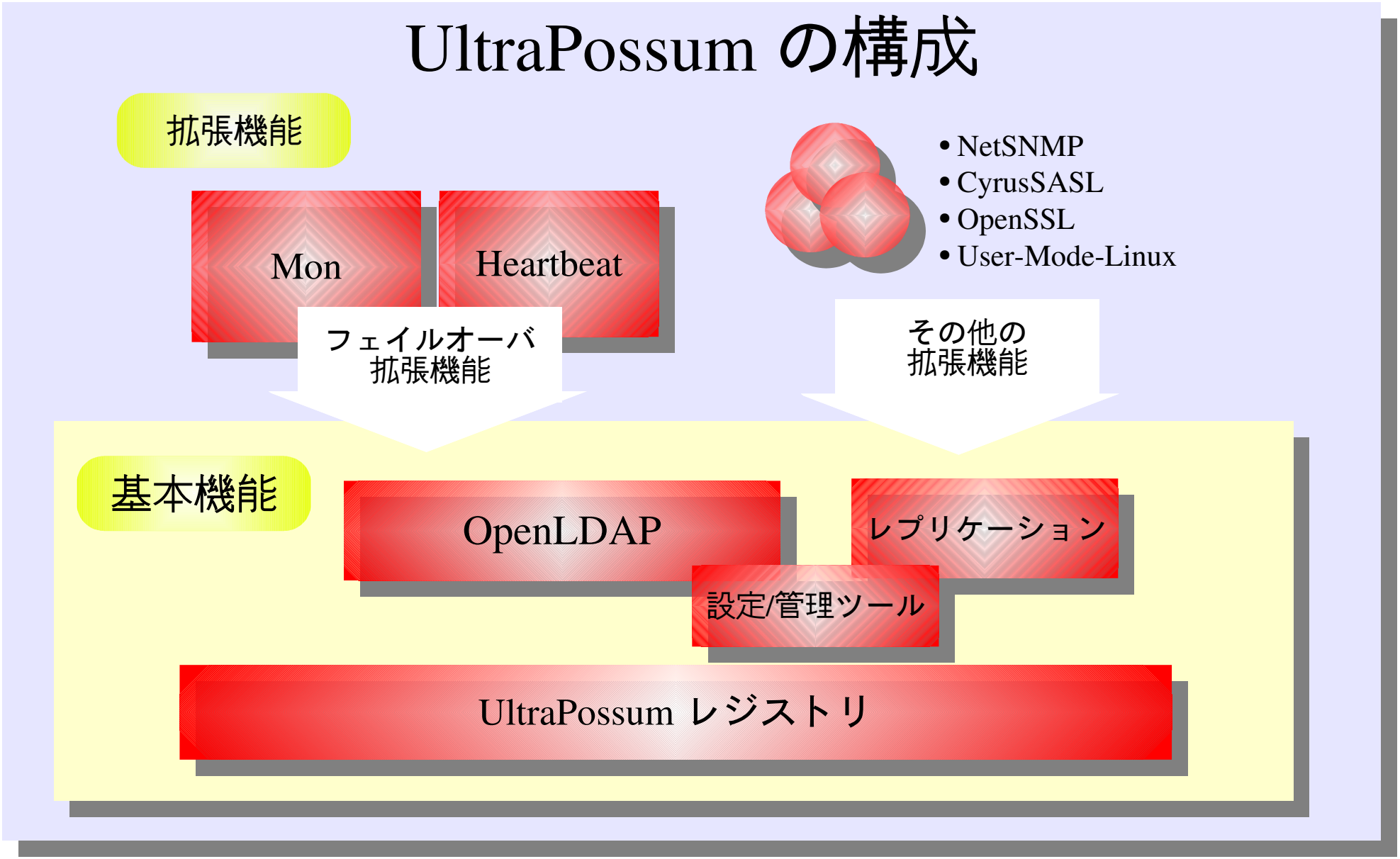
VA LINUX SYSTEMS JAPAN

3. UltraPossum 概要

3.1 UltraPossum とは?

- オープンソースディレクトリソリューション
  - 複数のオープンソースアプリケーションを統合
    - OpenLDAP/Heartbeat/Mon/SASL/OpenSSL 等
    - アプリケーション毎に異なる設定書式を統一
  - 大規模 LDAP システムでの使用を考慮
    - 機能拡張可能な設計 (Pluggable Module)
      - 基本サーバ機能/レプリケーション機能
      - 更新サービスフェイルオーバ機能
      - 通信路暗号化機能 (startTLS/SSL) / SASL 認証機能
      - SNMP 監視機能
    - 検証環境構築ツール
      - User-Mode-Linux を使った検証環境構築
    - SI テストツール
      - 実環境上での動作確認テストを自動化

3.2 UltraPossum の構成



3.3 UltraPossum レジストリ

- 環境固有の情報

エディタによる編集が可能

UltraPossum
レジストリ

(2)サーバ構築

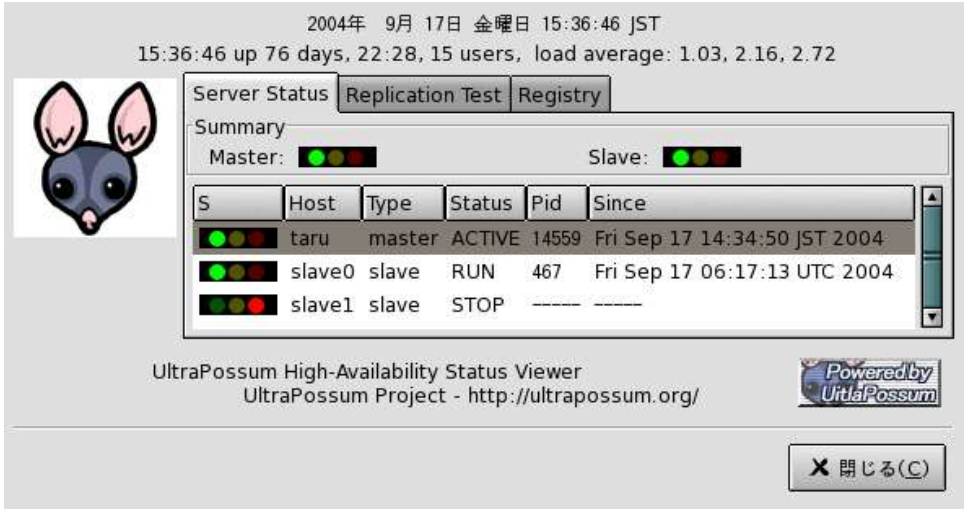
サーバを構築した設定情報を
クライアントがそのまま利用

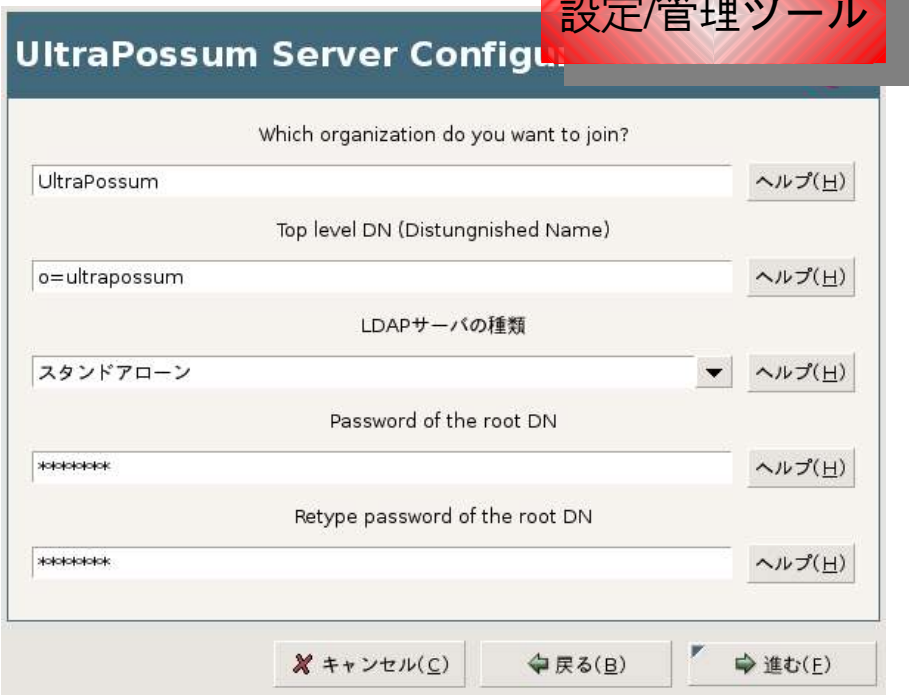
(3)

(1)

設定/管理ツール

アプリケーションがクラスタ情報を取得





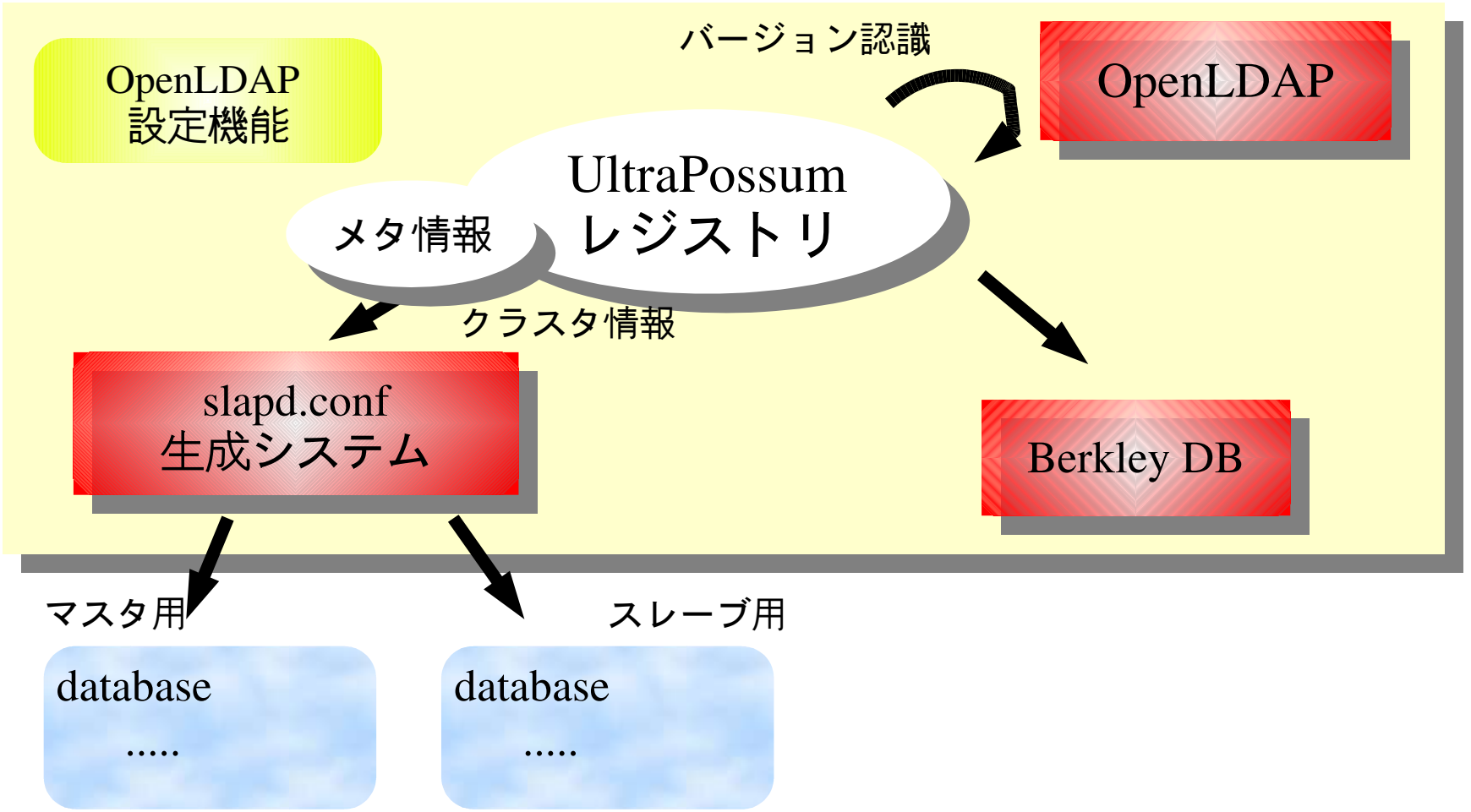
3.4 基本機能

VA LINUX SYSTEMS JAPAN

- OpenLDAP 設定
  - メタ情報による設定（クラスタ情報等）
  - OpenLDAP バージョン管理
- LDAP を利用するアプリケーションの管理
  - スキーマ/管理ID/初期エントリ/インデックス
- ローカルデータベースダンプ
  - バックアップ/リストア機能として利用可能

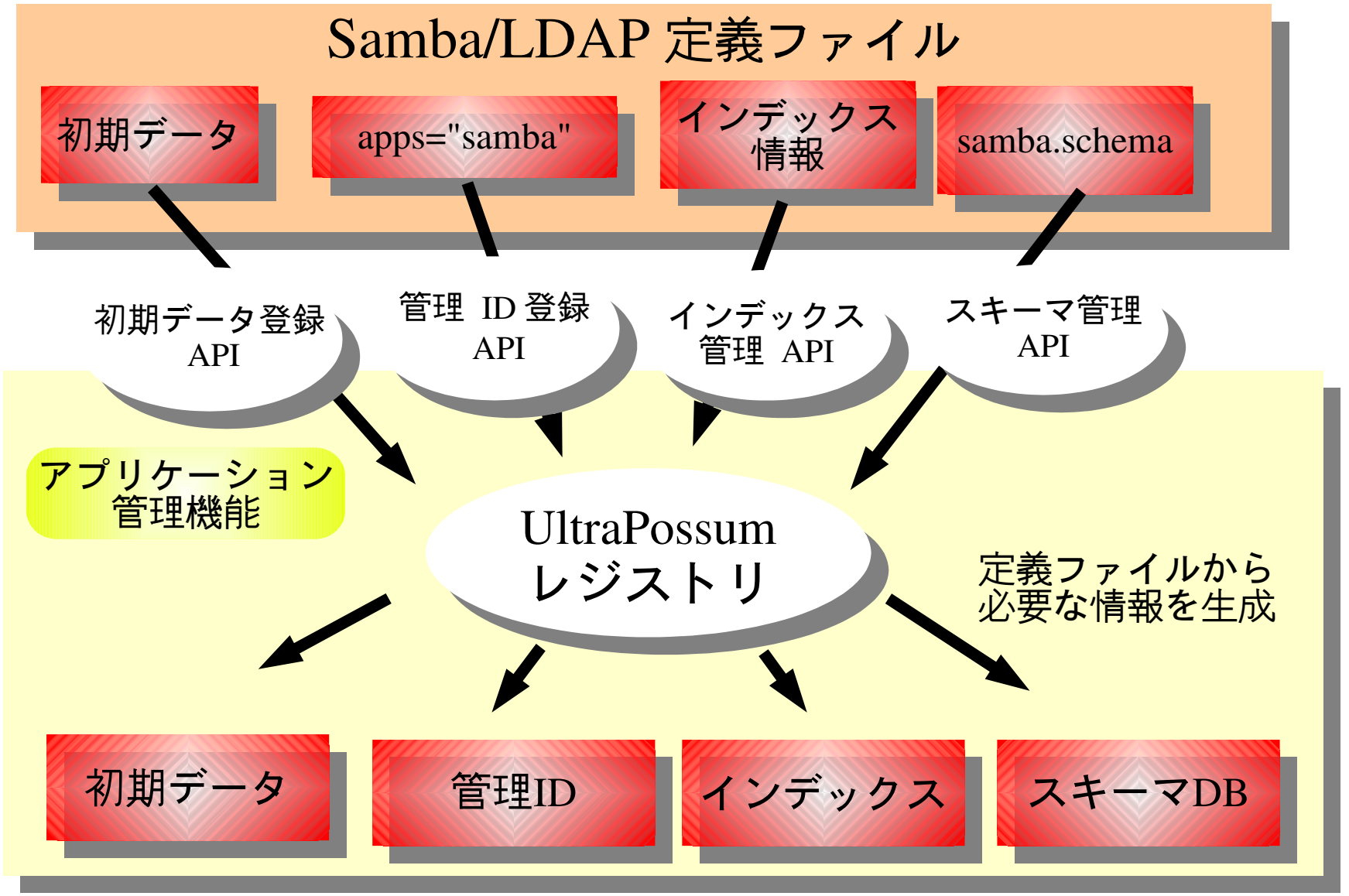
3.5 OpenLDAP 設定

- クラスタ構成を解釈して適切な設定ファイルを生成する

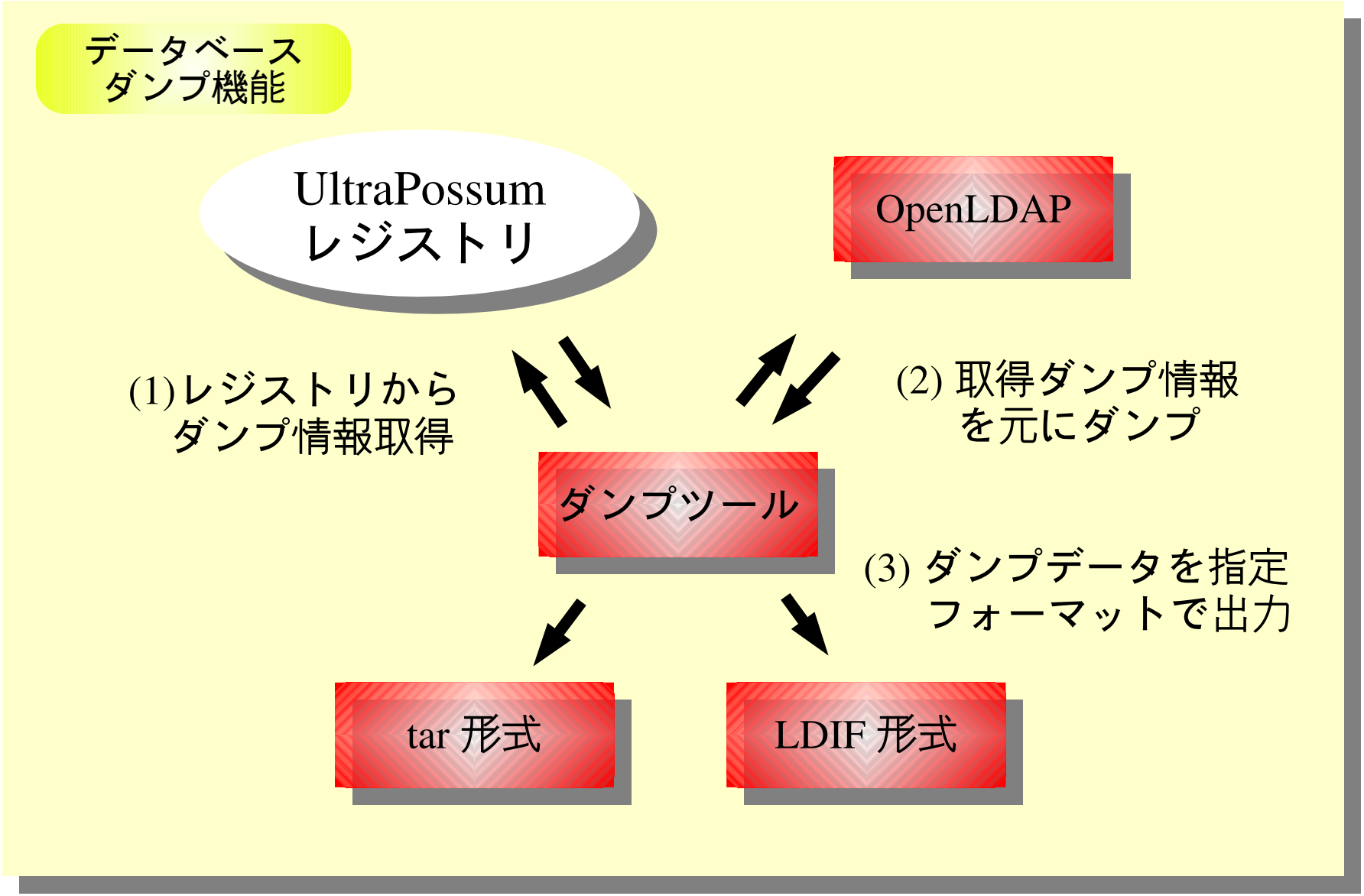


3.6 アプリケーション管理

Samba を UltraPossum で利用する例

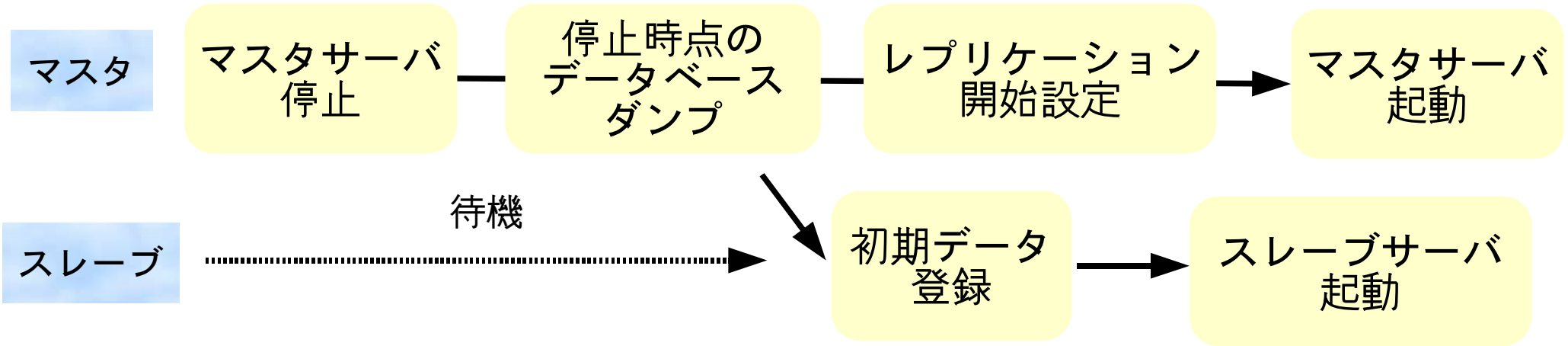


3.7 データベースダンプ機能



3.8 レプリケーション機能

- 複数のレプリケーションモデルをサポート
  - ログベースレプリケーション（slurpd）
  - 状態ベースレプリケーション（syncrepl）
  - 同期式レプリケーション（syncbackup）
- スレーブサーバの動的追加機能
  - 厳密な手順を UltraPossum が代行（slurpd）



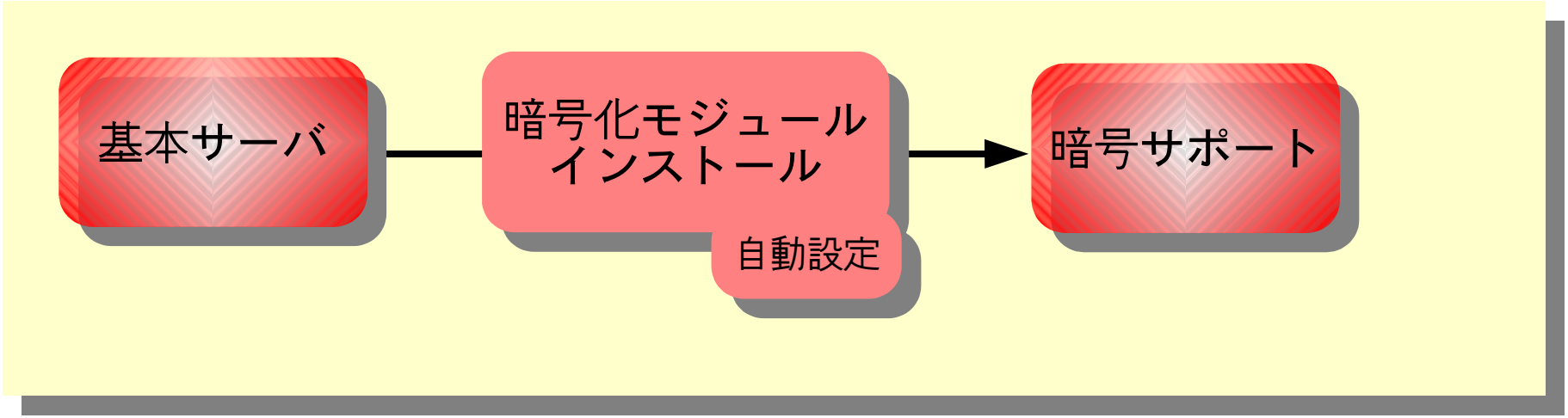
3.9 更新サービスフェイルオーバ機能

VA LINUX SYSTEMS JAPAN

- UltraPossum によるクラスタリング
  - 別途クラスタ製品の用意が不要
- ACTIVE/STANDBY 形式の二重化
  - slurpd, syncbackup レプリケーションモデルに対応
- 整合性確保のための回復処理
  - 共有ディスクを用い、ログを救出（ slurpd ）
  - プロトコル自身によるサポート（ syncbackup ）

3.10 通信路暗号化機能

- TLS/SSL のサポート
- 証明書管理



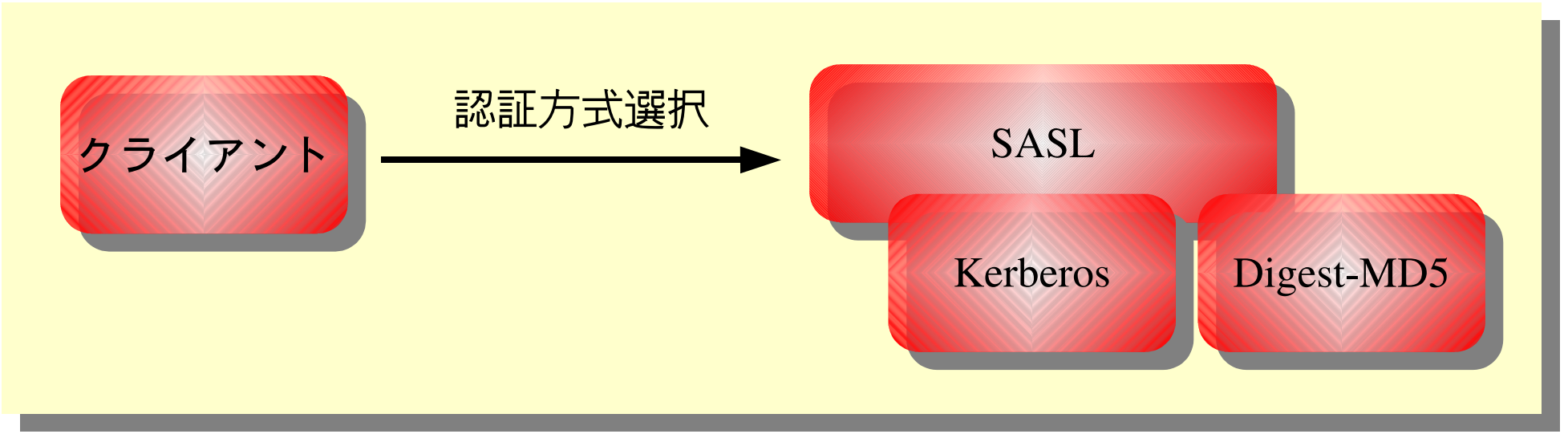
簡単インストール

```
~# apt-get install ultrapossum-tls
ultrapossum-tls (0.1beta13-0pre3) を設定しています ...

Creating config file /etc/ultrapossum/module.d/startTLS.cf with new version
Configuring startTLS extension... done
Restarting UltraPossum Server: slapd.
```

3.11 SASL 認証機能

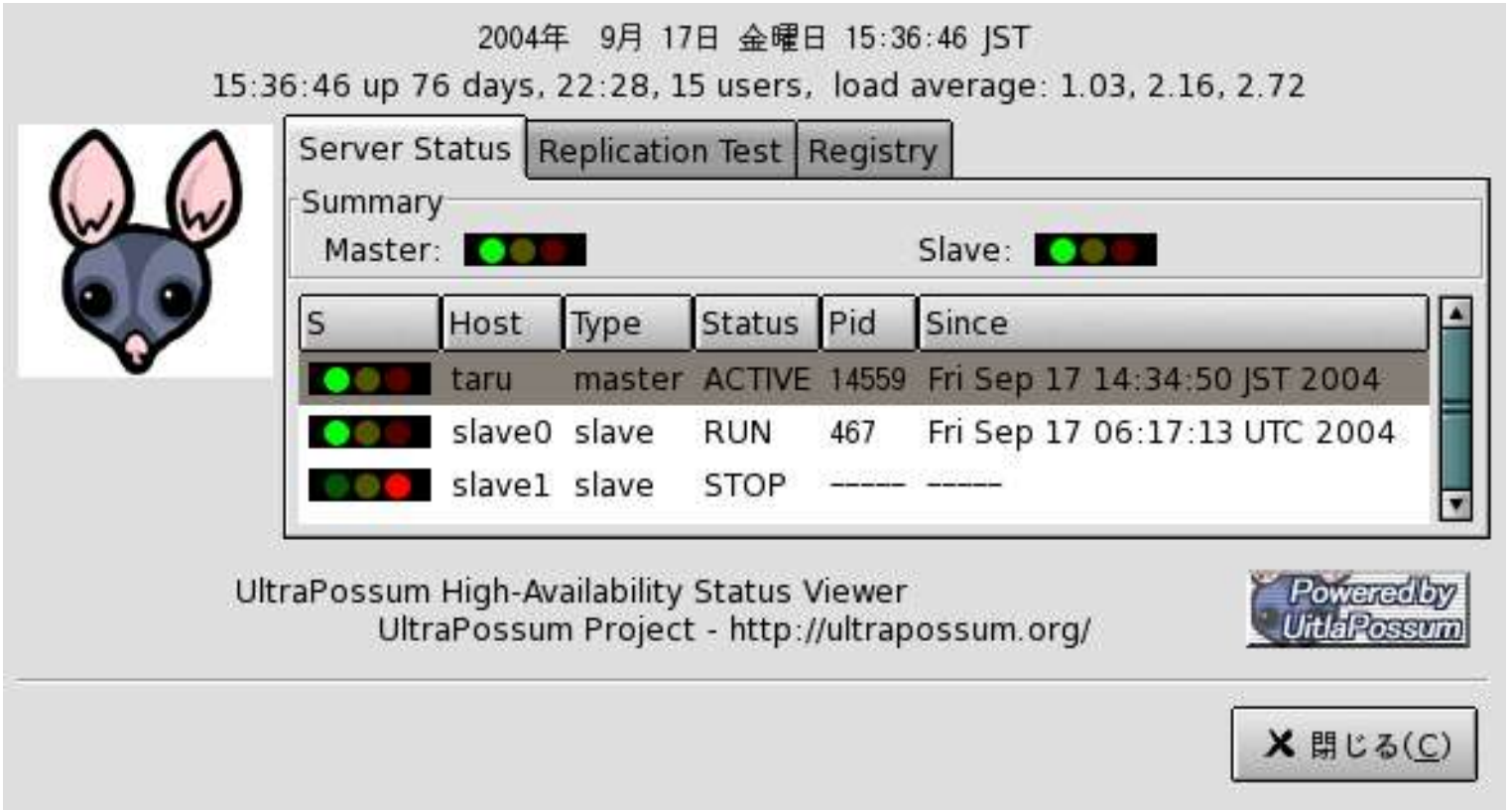
- LDAPv3 SASL 認証
  - Simple Authentication and Security Layer
    - インターネット上での汎用認証レイヤ




```
~# apt-get install ultrapossum-sasl
```

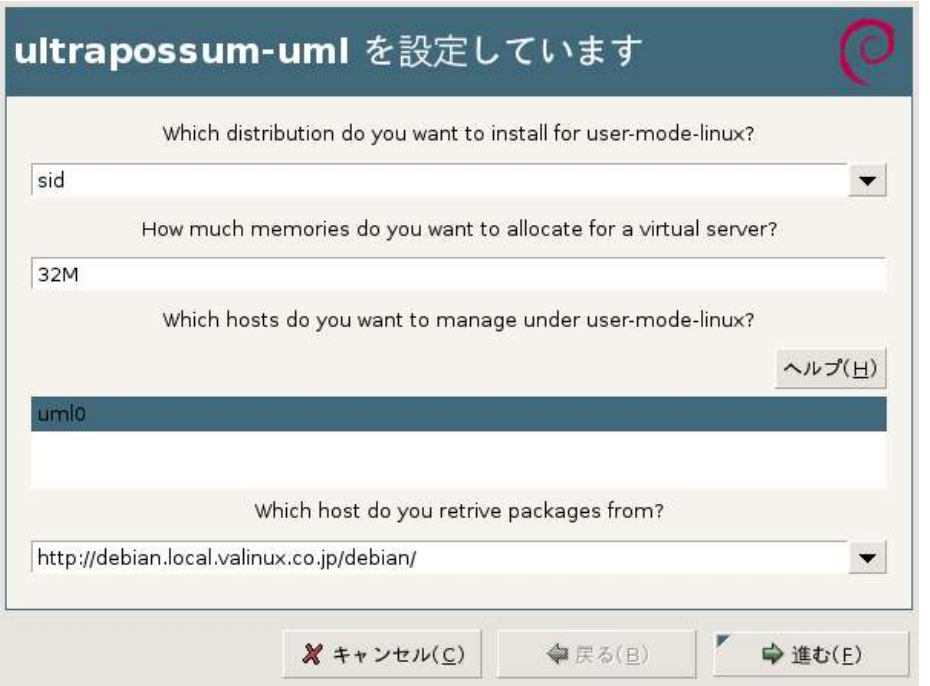
3.12 SNMP 監視機能

- SNMP による UltraPossum の状態監視
  - 動作/停止/異常終了/フェイルオーバ



3.13 検証環境構築機能

- User-Mode-Linux による検証環境
  - ファイルシステム構築
  - ネットワーク設定
  - 自動起動/停止
  - バックグラウンド実行




```
 ~#  /etc/init.d/ultrapossum-uml start
Starting UltraPossum inside User-Mode-Linux: uml0.

~# /etc/init.d/ultrapossum-uml stop
Stopping UltraPossum inside User-Mode-Linux: .. uml0.
```

3.14 SIテスト機能

VA LINUX SYSTEMS JAPAN

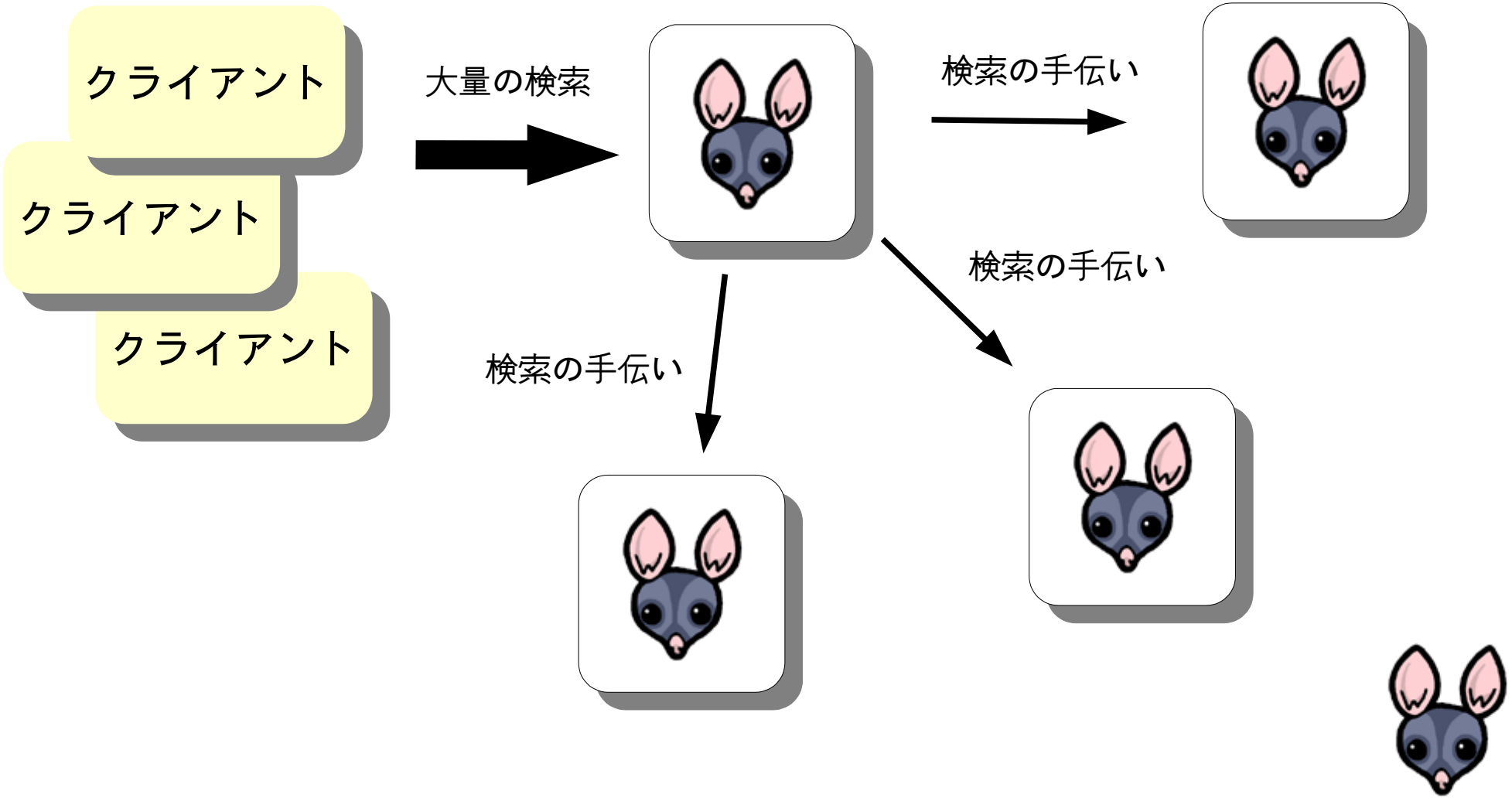
- 実環境上での動作確認フレームワーク
  - テストスクリプトが環境を自動的に認識し、環境に応じたテストを実行

```
~# /usr/local/sbin/ultrapossum-test
Testing /usr/lib/ultrapossum/test/test.d/00protocol... ok
Testing /usr/lib/ultrapossum/test/test.d/05replication... ok
Testing /usr/lib/ultrapossum/test/test.d/10clsfailover... ok
```

- 標準添付スクリプト
  - 追加/更新/検索/削除テスト
  - レプリケーションテスト
  - フェイルオーバ/フェイルバックテスト

3.15 その他の機能（ 検索負荷分散 ）

- UltraPossum 単体で IP 負荷分散を行う



3.16 対応 OS

- 全機能サポート
  - Debian GNU/Linux (sarge 以降)
- 制限付きサポート (対応拡張機能に制限)
  - Debian GNU/Linux (woody)
  - RHEL 3.0
  - Solaris 8
- 標準的な UNIX マシンでは動作可能。
  - 必須ソフトウェア
    - OpenLDAP 2.0 以降
    - bash
    - perl 5.8 以降 (更新サービスフェイルオーバ)

VA LINUX SYSTEMS JAPAN

4. UltraPossum の導入
- フェイルオーバ機能使用 -

4.1 必要なもの

- ソフトウェア
  - bash
  - OpenLDAP 2.0 以降 (BerkleyDB)
    - http://openldap.org/
  - heartbeat/mon/sudo (フェイルオーバ機能使用時)
    - http://linux-ha.org/
    - http://www.kernel.org/software/mon/
- その他
  - NFS サーバ (slurpd によるフェイルオーバ機能)

4.2 OpenLDAP インストール

VA LINUX SYSTEMS JAPAN

- Quick Start Guide
  - http://www.openldap.org/doc/admin22/quickstart.html
  - UltraPossum を利用する場合は (7) までで良い

```
(2)   ~$ gunzip -c openldap-VERSION.tgz | tar xvfB -
      ~$ cd openldap-VERSION
(4)   ~$ ./configure
(5)   ~$ make depend
      ~$ make
(6)   ~$ make test
(7)   ~$ su root -c 'make install'
```

4.3 heartbeat インストール

- フェイルオーバ機能使用時のみ
- Getting Started with Linux-HA (heartbeat)
  - http://linux-ha.org/download/GettingStarted.html

```
~$ ./ConfigureMe configure
~$ make
~$ su -c “make install"
```

4.4 mon インストール

- フェイルオーバ機能使用時のみ
- INSTALL ファイル
  - http://cvs.sourceforge.net/viewcvs.py/mon/mon/INSTALL?view=markup
- 必要な Perl モジュール（CPAN）
  - Time::Period
  - Time::HiRes
  - Convert::BER
  - Net::LDAP
  - Mon::*

```
prefix=/usr/local; mon=$prefix/lib/mon/mon.d
alert=$prefix/lib/mon/alert.d; etc=$prefix/etc/mon

install -d $prefix/sbin $prefix/bin $mon $alert $etc

cp mon $prefix/sbin
cp clients/moncmd clients/monshow \
                    clients/skymon/skymon $prefix/bin
cp etc/auth.cf $etc
cd mon.d; make LDFLAGS="-lnsl -lsocket"; cd ..
cp mon.d/*.monitor $mon
cp alert.d/*.alert $alert
```

4.5 UltraPossum インストール

- フェイルオーバ環境構築クイックガイド
  - http://www.ultrapossum.org/quickguide_misc2.html

```
~$ tar zxvf ultrapossum-VERSION.tar.gz
~$ cd ultrapossum-VERSION
~$ ./configure
....
built-in modules: client server test failover ← ここで failover モジュール
                                                がインストール対象に
                                                なっていることを確認
...
~$ make check
~$ su -c "make install"
```

build-in modules で failover モジュールがない場合は必要な
ソフトウェアが自動検出できなかったことを表す

./configure の出力を確認

4.6 UltraPossum 主な設定ファイル

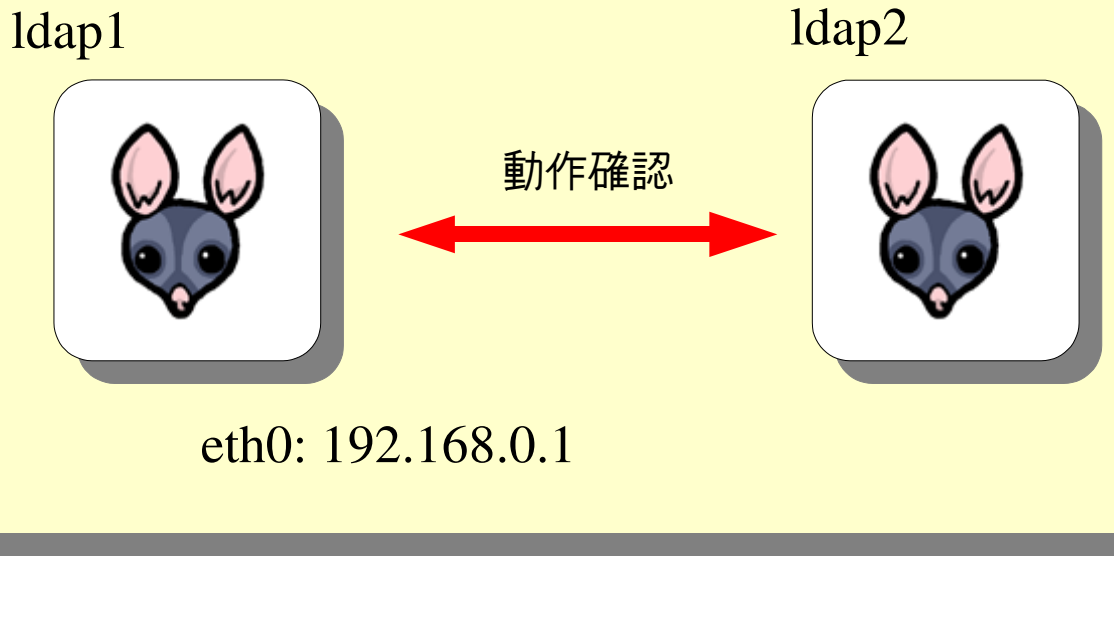
VA LINUX SYSTEMS JAPAN

- /usr/local/etc/ultrapossum/ultrapossum.cf
  - 全体の設定
- /usr/local/etc/ultrapossum/default.cf
  - heartbeat のデフォルト設定情報
    - heartbeat の設定ファイルが存在しない時に利用される
- /usr/local/etc/ultrapossum/module.d/*.cf
  - 各機能毎の設定
    - server.cf - 基本機能の設定
    - failover.cf - フェイルオーバ機能の設定

4.7 設定環境

- マスタサービスの仮想 IP – 192.168.0.1
- マスタサーバのホスト: ldap1
- バックアップサーバのホスト: ldap2

設定ファイルはクラスタの全ホスト間で同一にする

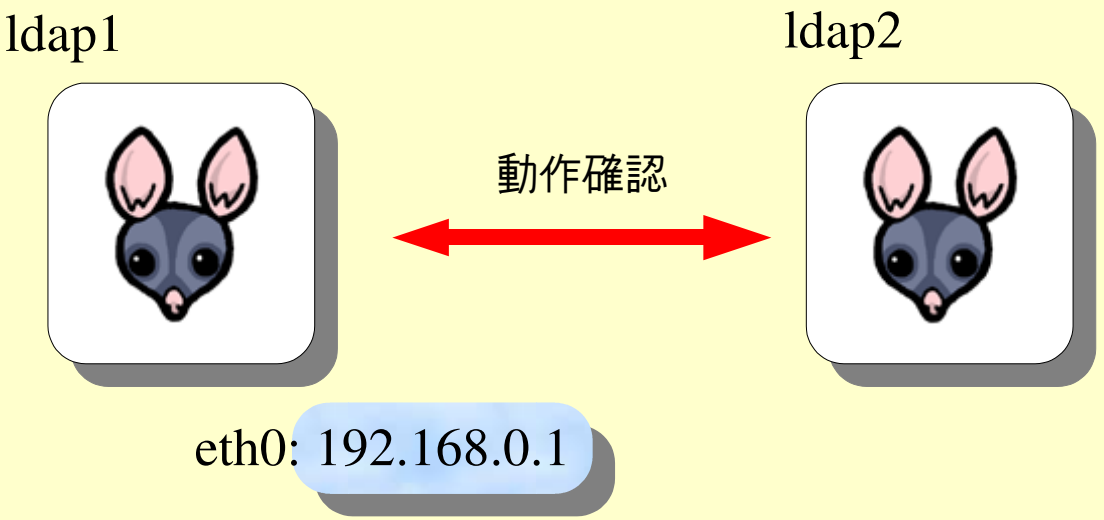


4.8 ultrapossum.cf の設定

- /usr/local/etc/ultrapossum/ultrapossum.cf

```
# Master Host
LDAPMASTER="192.168.0.1"
```

マスタサービスのホスト名または
IP アドレスを設定する



4.9 server.cf の設定

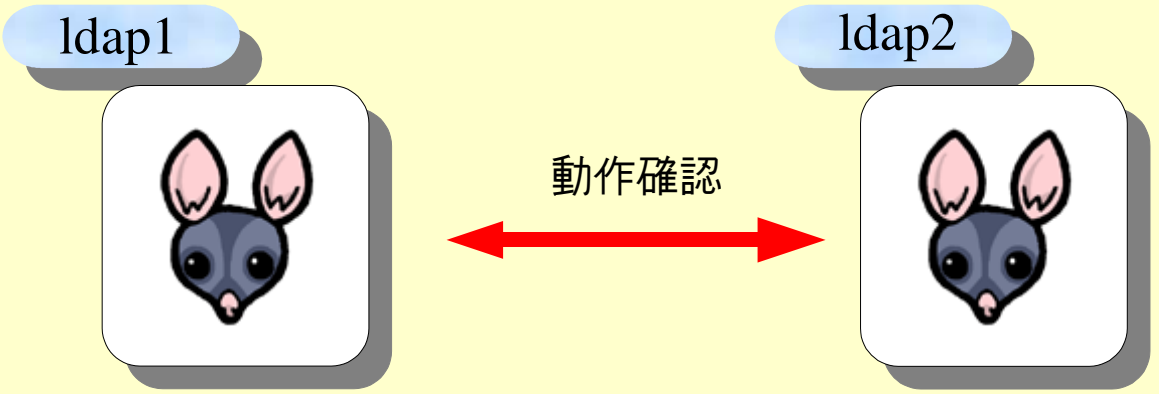
- /usr/local/etc/ultrapossum/module.d/server.cf

```
# PATH to slaptools
PATH=$PATH:/usr/local/sbin

# Actual Master host
MASTER="ldap1"

# List of replica servers by slurpd
SLURPDSLAVES="ldap2"
```

- slapcat, slapadd ツール等のパス
- 実際のマスタサーバ
- slurpd によるスレーブサーバ



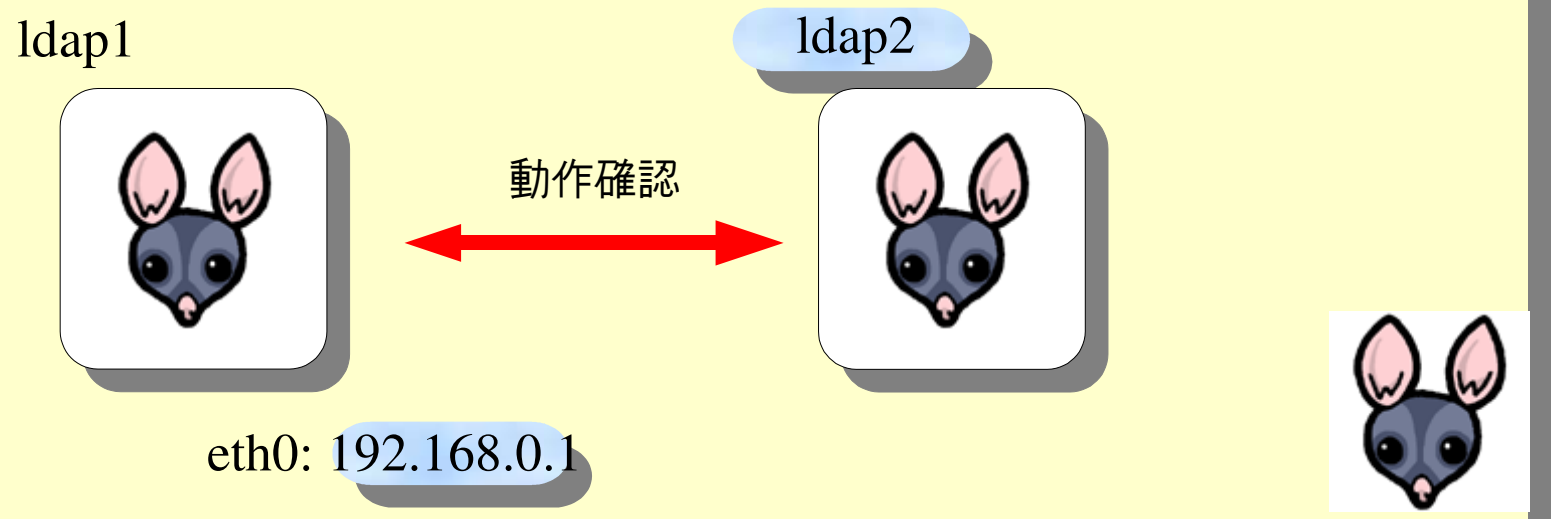
4.10 failover.cf の設定

- /usr/local/etc/ultrapossum/module.d/failover.cf

```
# Virtual IP for master service
VIRTUAL="192.168.0.1"

# Backup host for failover
BACKUP="ldap2"
```

- マスタサービスの仮想 IP
  - 今回は LDAPMASTER に IP アドレスを用いているため同じ値
  - ここを空白にすると仮想 IP の引き継ぎを行わない(負荷分散装置等で対応)
- バックアップサーバホスト名
  - 設定済スレーブサーバの中から選ぶ



4.11 NFS ディレクトリの設定

VA LINUX SYSTEMS JAPAN

- NFS を既にマウントしてある場合は自動設定

  UltraPossum レジストリから NFS ディレクトリ情報を取得

```
~$ /usr/local/bin/ultrapossum-config get NFSDIR
NFSDIR="/share"
```

- NFS をマウントしていない場合はマウントする
- NFS マウントが 2 箇所以上ある場合は初めに見付かったものが優先
- デフォルト以外の値を使う場合は failover.cf で NFSDIR を設定

4.12 default.cf の設定

- heartbeat の設定を簡素化する設定
  - heartbeat の設定を自動化したい時に利用

```
logfile="/var/log/ha-log"
serial="/dev/ttyS0"
udp="eth0" ← heartbeat1.2.3のbcast
auto_failback="on"
```

- heartbeat の ha.cf が存在しない場合時のみ利用される
  - テンプレート生成程度の意味合い
  - 再設定を行う場合は手で編集するか明示的にファイルを削除する
  - UltraPossum のみで heartbeat を使う場合は以下のファイルを削除
    - /usr/local/etc/ha.d/authkeys
    - /usr/local/etc/ha.d/ha.cf
    - /usr/local/etc/ha.d/haresources
    - /usr/local/etc/mon/mon.cf

4.13 環境依存初期エントリ登録

VA LINUX SYSTEMS JAPAN

- UltraPossum は最低限必要なエントリを自動生成
  - トップエントリ/ルートエントリ/管理エントリ
- それ以外のエントリを登録する場合
  - /usr/local/etc/ultrapossum/in.d/init.ldif.in
    - UltraPossum レジストリの情報を利用

```
dn: ou=People,#SUFFIX#
objectClass: organizationalUnit
ou: People
```

  - /usr/local/etc/ultrapossum/init.ldif.d/
    - 登録するエントリを LDIF 形式で配置

4.14 設定反映

- 設定を反映

```
~# /usr/local/sbin/update-ultrapossum -f configure
```

  - 起動に必要な全設定が行われる
    - エントリの生成
    - フェイルオーバモジュールの有効化

- 登録を確認

```
~$ ultrapossum-config status failover
failover=installed
```

設定ファイルに誤りがある場合はエラーで終了する

```
~# /usr/local/sbin/update-ultrapossum -f configure
...
E: BACKUP 'foo' is not a slave server
```

マスタサーバ、バックアップサーバの両方で行う

4.15 UltraPossum 起動

- マスタサーバ起動

```
~# /usr/local/sbin/ultrapossum-server -v start
```

マスタサーバがスレーブサーバの起動を確認するまで待機

- スレーブサーバ起動

```
~# /usr/local/sbin/ultrapossum-server -v start
```

マスタサーバとスレーブサーバがお互いを認識し UltraPossum を起動

- テスト

```
~# /usr/local/sbin/ultrapossum-test
```

4.16 その他の準備(起動)

VA LINUX SYSTEMS JAPAN

- システム起動時に毎回起動させる
  - init スクリプトを登録する
    - Debian ではパッケージインストールで自動的に登録
    - RHEL: ソースアーカイブの redhat/ultrapossum.init
    - Solaris: ソースアーカイブの tools/solaris.init

```
~# /etc/init.d/ultrapossum-server start
Starting UltraPossum Server: slapd slurpd.
```

4.17 運用時のヒント(初期化)

VA LINUX SYSTEMS JAPAN

- ディレクトリを初期状態に戻す
  - 基本機能ツールディレクトリの formatdb.sh

```
~# ultrapossum-config get MODULEDIR
MODULEDIR="/usr/share/ultrapossum/module.d"
~# /usr/share/ultrapossum/module.d/server/formatdb.sh -f
```

- サーバ起動中は初期化できない
  - -f オプションをつけるとサーバを停止して初期化する
- クラスタ環境を構築しているときは初期化を全てのサーバで実行する必要がある

4.18 運用時のヒント(起動状況確認)

VA LINUX SYSTEMS JAPAN

- LDAP サーバが起動しているかを調べる
  - 基本機能ツールディレクトリの startup status

```
~# ultrapossum-config get MODULEDIR
MODULEDIR="/usr/share/ultrapossum/module.d"
~# /usr/share/ultrapossum/module.d/server/startup status
running
```

4.19 その他

VA LINUX SYSTEMS JAPAN

- 情報源
  - http://ultrapossum.org/
  - メーリングリスト(日本語)
    - http://lists.sourceforge.jp/mailman/listinfo/ultrapossum-users
- 協力者募集
  - ドキュメント
  - バグ報告
  - 他 OS/ ディストリビューションでの動作検証
  - コーディング

VA LINUX SYSTEMS JAPAN

5. 大規模システムでの注意点

5.1 注意点

VA LINUX SYSTEMS JAPAN

- オープンソースである

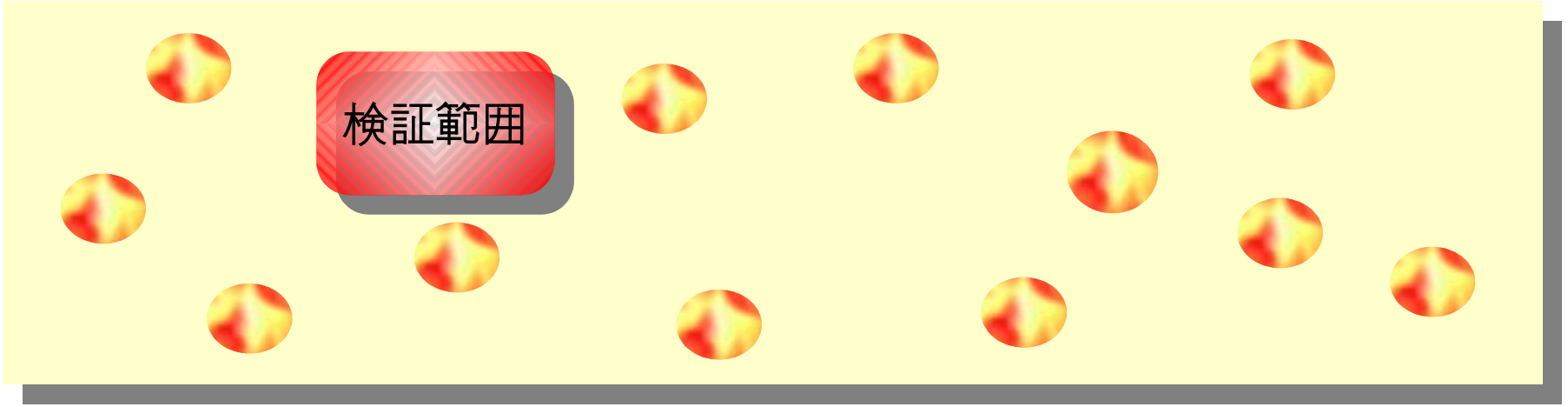
5.2 オープンソース

VA LINUX SYSTEMS JAPAN

- AS IS
  - 自由に手に入る ＝全て自己責任
- Release Often
  - バグが修正される利点
  - 新しいバグが生まれる欠点
    - 新機能をリリースしたい開発者心理
    - リリース版に入れないとテストユーザを確保できないジレンマ
- 開発者の興味は最新版
  - 古いバージョンは放置される

5.3 地雷

- 安全そうに見えて突然爆発
  - ○○は安定している
    - また聞き
    - 一部を見て全体を判断



オープンソースに限らずソフトウェア全般に存在

5.4 heartbeat カウンタオーバフローバグ

- 1.2.2 までの全バージョン
  - カーネル起動後 497 日目(*)に heartbeat が発狂
- VA Linux Systems Japan, K.K. より報告
  - 1.2.3 リリースにパッチ採用

(*) 発狂日は OS に依存

5.5 OpenLDAP slurpd 異常終了

- 2.0 の全バージョン, 2.1 の特定バージョン
  - レプリケーションログの使いかたによってセグメンテーションフォルト
- 2.1.1X にて修正

5.6 OpenLDAP レプリケーションログ破損

- レプリケーションログ操作関数でのエラー処理
  - タイミングによりレプリケーションログ破損
- 未修正
  - VA Linux Systems Japan, K.K のコンサル事業では対応済み (コミュニティには未報告)

5.7 OpenLDAP インデックスアルゴリム問題

VA LINUX SYSTEMS JAPAN

- インデックス情報生成アルゴリズムによる仕様
  - マルチバイト圏でのみ発生
  - 2.0 では重要な問題
  - 2.1 までは大きな制限
  - 2.2 では多少回避
- 特定の検索フィルタ式を用いる事でサーバを高負荷状態にできる

5.8 syncrepl psearch consistency バグ

- 2.2.14/2.2.17
  - refreshAndPersist モードでのバグ
    - エントリ削除時にその情報が正しくスレーブに伝わらないことがある

5.9 オープンソースの利点

VA LINUX SYSTEMS JAPAN

- 緩い制約
- ノウハウの蓄積
- ソースレベルの解析
  - 正確な仕様書
- 開発者との密な関係

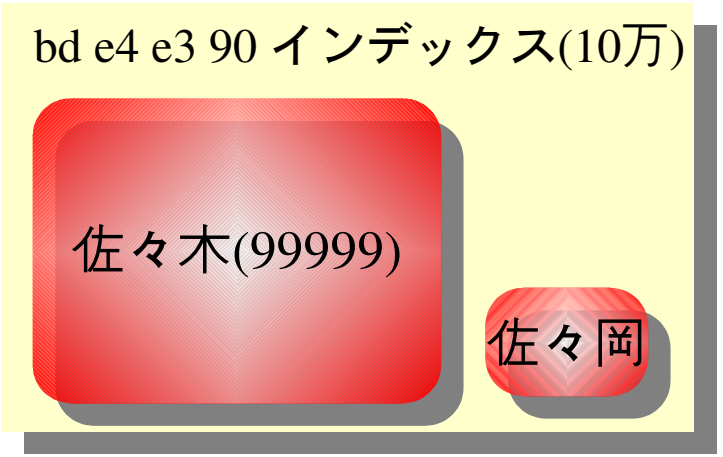
5.10 問題点の対処例(1)

VA LINUX SYSTEMS JAPAN

- heartbeat オーバーフローバグの解決まで
  - システム高負荷発覚の報告
  - 高負荷の原因を探る
    - heartbeat がシステム時間を 99% 使用している
  - 発狂する閾値になるような値を探す
    - IA-32 であれば 2^32 – 1, 2^31-1
      - プロセス起動時間, カーネル起動時間, オープン中のファイルサイズ
      - 497 日問題 (秒 * クロック数(100Hz))
  - ログを見て通常と違う箇所を探す
    - 通常より間隔が異常に短いメッセージを探す
      - そのメッセージをキーに関数を探す (オープンソース特有)
      - その関数から呼んでいるシステムコールの中でカウンタが一周するようなものがあるか探す (例: times(2))
      - 返り値の使い方を調べ、バグを発見する

5.11 問題点の対処例(2)

- OpenLDAP インデックス問題の調査
  - 時おりサーバが高負荷になる
    - 高負荷時の検索フィルタログを収集
    - 抽象性を抽出する
      - 日本語文字の部分一致検索で負荷が高くなるようだが全てではない
        - 「佐々岡」,「田中山」等の検索が遅い
    - インデックス生成アルゴリズムを調査する
      - インデックスの生成が最初の 4 バイトまでであることがわかる
        - 佐々木の UTF-8 バイト列は “bd e4 e3 90 85 80 9c e6 0a a8"
        - 佐々岡の UTF-8 バイト列は “bd e4 e3 90 85 80 b2 e5 0a a1"

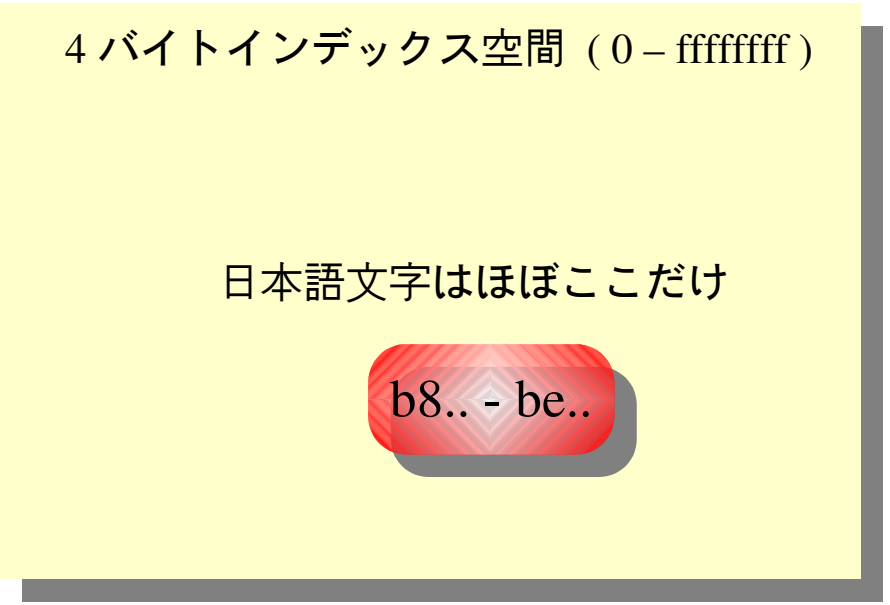


•佐々木さんの検索はサイズリミットで中断
•佐々岡さんの検索は 10 万エントリ全て探索

5.12 CJK 文字コードの局所性問題

VA LINUX SYSTEMS JAPAN

- 日本語文字群は先頭のバイト列が似ている
  - Unicode - CJK 統一表意文字(一(4E00) ～ 龥(9FA5))
    - 龥 - http://en.wiktionary.org/wiki/%E9%BE%A5
  - UTF-8 エンコーディング
    - 一 (b8 e4 0a 80) ～ 龥 (be e9 0a a5)
  - 現在の対処法
    - インデックス幅を大きくする
    - クエリキャッシュを用いる
    - リミット値を設定する
      - man slapd.conf(5)
    - 部分一致検索を使わない



まとめ

VA LINUX SYSTEMS JAPAN

- OpenLDAP を大規模システムで使う方法
- UltraPossum の概要/導入
- オープンソースを使う上での注意/対処の具体例

開発に参加してください

- コーディング/ドキュメント/バグ報告