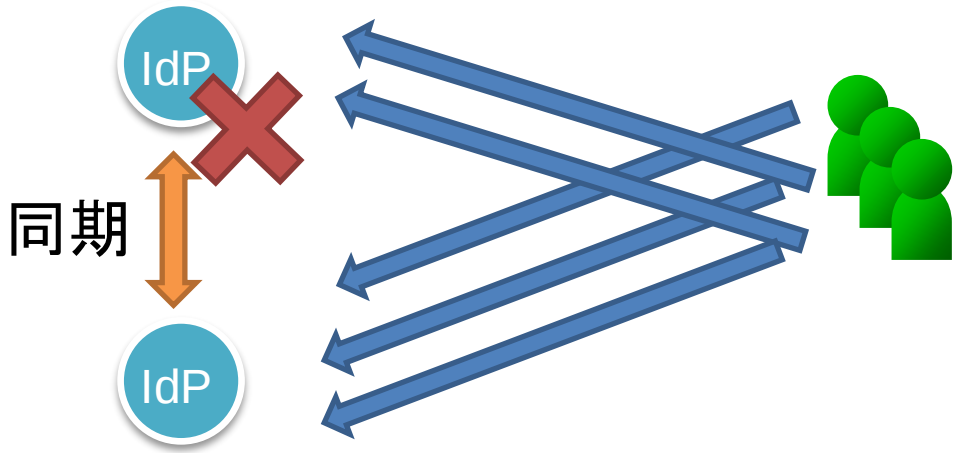
IdP冗長化の方法と仕様書の書き方

国立情報学研究所　西村　健

IdP冗長化のために

冗長化（クラスタリング）:

- 複数ノードでShibboleth IdPを起動する
- 1つのノードが落ちてもサービスを続ける
  →続けるためにはデータの同期が必要



同期すべき情報には2種類ある

- ユーザが認証済みという情報
  - ログイン後
  - 長期間存在する（～8時間）

- ログイン処理中のみ存在する情報
  - ログイン画面を出して，
    ID/パスワードを送信するまでの間
  - 両者が別のノードだったらどうなる？
  - 短期間

3つの冗長化方式

- Terracotta方式
- Stateless Clustering方式
- repcached(memcached)方式

Terracotta

Tomcatの汎用的な冗長化ソフトウェア

特長:

- 完全な冗長化
  - 全てのデータを(JavaのObject単位で)レプリケートする

デメリット:

- 複雑
- 相性問題(バージョン間の差異が問題に)

Stateless Clustering

- レプリケートすべきデータをcookieやID(transientID)に埋め込む！
- 軽い！
- ノード間の通信がゼロ
  - 同一ネットワークセグメントである必要がない

- 不完全な冗長化
- IdP機能制約あり　ー　学認が規定する範囲では問題なし
- IDが想定外に長くなることによりSP側で誤動作の可能性
  - 例: Web of Knowledge
    - SAML1でもフロントチャネルで回避

repcached(memcached)

- レプリケートすべきデータをmemcachedに置く
  - memcachedをDBとして用いる
  - repcachedはmemcachedをレプリケート対応したもの
- ある程度軽い
- SP・ユーザとの接点のデータ形式は変更しない

デメリット:

- 不完全な冗長化
- ノード間通信が少し

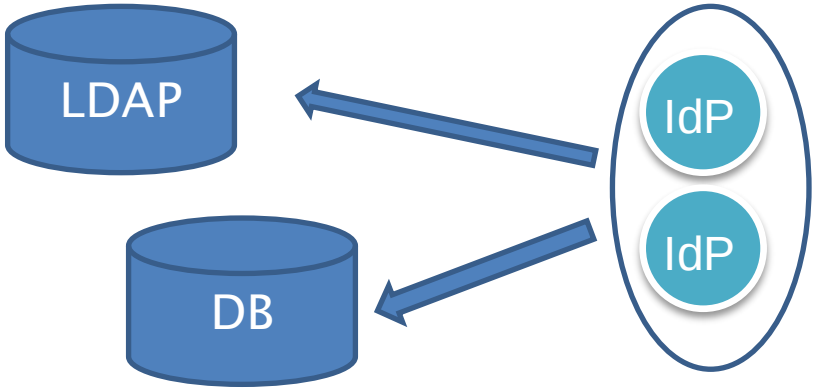
各ノードへのアクセス手段提供

ノードを複数にすることと，それらへのアクセス手段を提供することは別問題

- ネットワークプロトコルレベルで解決
  - VRRP Active-Standby
  - Anycast
  - L4/L7スイッチ
  - …
  - **ただし，stateless/memcached＋DNSラウンドロビン　はダメ**
- ロードバランサを入れる
  - sticky機能でログイン処理中は特定のノードに行かせる
  - そもそもShibbolethは
    ハードウェアロードバランサ＋Terracotta
    推奨
  - ただし，（ハードウェアのものは）高い！

これだけで大丈夫？

ではないです！



- バックエンド（LDAP）の冗長化
- StoredIDのためのDB(MySQL)の冗長化
- uApprove.jpのためのDB(MySQL)の冗長化

いろいろな既存技術があると思いますのでお好みで
（情報をいただければありがたいです）

ちょっとだけ性能比較

	MeanTestTime(ms)	TPS（トランザクション/s）
Terracotta	1235.99	15.22
Stateless	398.61	47.20
repcached	494.44	38.89

- あまり最適化してませんが…
- statelessはCryptoTransientID無効での計測
- 詳細は後述のウェブページ参照

仕様書の書き方

- 学認が公開している手順書を提示して
  「これこれに従ってN台の冗長構成にすること」
- バックエンドの冗長化も忘れずに

- どの方式を使うかはお好みで
  - 最も単純 ー Stateless Clustering
  - 最も複雑 ー Terracotta
  - その中間 ー repcached

IdP冗長化のための資料

- Shibboleth-IdP冗長化環境構築手順書(Stateless Clustering編)
- Shibboleth-IdP冗長化環境構築手順書(memcached編)
  - repcachedは最後にちょこっと
- Shibboleth-IdP冗長化環境構築手順書(Terracotta編)

金沢大学提供

  - 学認サイト > IdPカスタマイズ　の
「ロードバランサー配下のシボレスIdP環境設定に関する検証実験」
も参照のこと.
- Shibboleth-IdP冗長化パフォーマンス比較試験報告書

いずれもmeatwiki.nii.ac.jpの

GakuNinShare > 技術情報 > IdPClustering

にて公開しています.