

パフォーマンス最適化のための サーバー構築

シックス・アパート株式会社

特記事項

- 本資料の記載内容は、できる限り正確を期すよう努めてはおりますが、いかなる明示または暗黙の保証も責任も負いかねます。
- 本資料の情報は、使用先の責任において使用されるべきものであることを、あらかじめご了承ください。
- 掲載情報は不定期に変更されることもあります。他のメディア等に無断で転載する事をご遠慮ください。
- 本資料をコピー等で複製することは、執筆者の承諾なしではできません。
- 本資料に記載された製品名または会社名はそれぞれの各社の商標または登録商標です。

MTの特性を知ればチューニングの ポイントがみえてくる

Movable Type の特性

- Perl で動作する CGI スクリプト
- 多くの DBMS に対応

Perl で動作する CGI スクリプト

- CPU コア単位で動作
 - マルチコアの数よりクロックに依存
- リクエスト毎に Perl が起動して実行
 - ディスク I/O に依存

Perl の初期化処理の時間を軽減

FastCGI の利用

FastCGI を利用するメリット

単位 (秒)	FastCGI 無効	FastCGI 有効 1 回目	FastCGI 有効 2 回目
ログイン画面の 表示までの時間	22	28	1
ログイン後、 ダッシュボード 表示までの時間	30	8	8
ダッシュボード からログアウト 完了までの時間	22	1	1

参考: ある環境での測定値

多くの DBMS に対応

- **リクエスト毎にサーバーに接続**
 - コネクションプーリング未使用
 - リクエストが無いと無風状態
- **ディスク I/O の影響が大きい**

DB 接続の時間を軽減

memcached の利用

memcached の利用

- 最近のウェブサイトではよく使われている技術 (例: Twitter、TypePad)
- DB アクセスが多いサイトでは意味がある技術
 - 検索の利用が多い、カスタムフィールドが多い

MySQL のチューニング

- **ストレージエンジンは InnoDB に**
 - MyISAM: キャッシュの対象がインデックス
- **my.cnf の値をチューニング**
 - mysqltuner.pl (<http://mysqltuner.com>)
 - mysql> show global status like 'Innodb_buffer_pool%';
を実行
 - innodb_buffer_pool_read/
innodb_buffer_pool_read_requestsの値を限り無く小さく

あるサイトでチューニング後のフル再構築時間

before : after = 9時間 : 1時間

ウェブ/アプリ/DBサーバーを どう構成するか

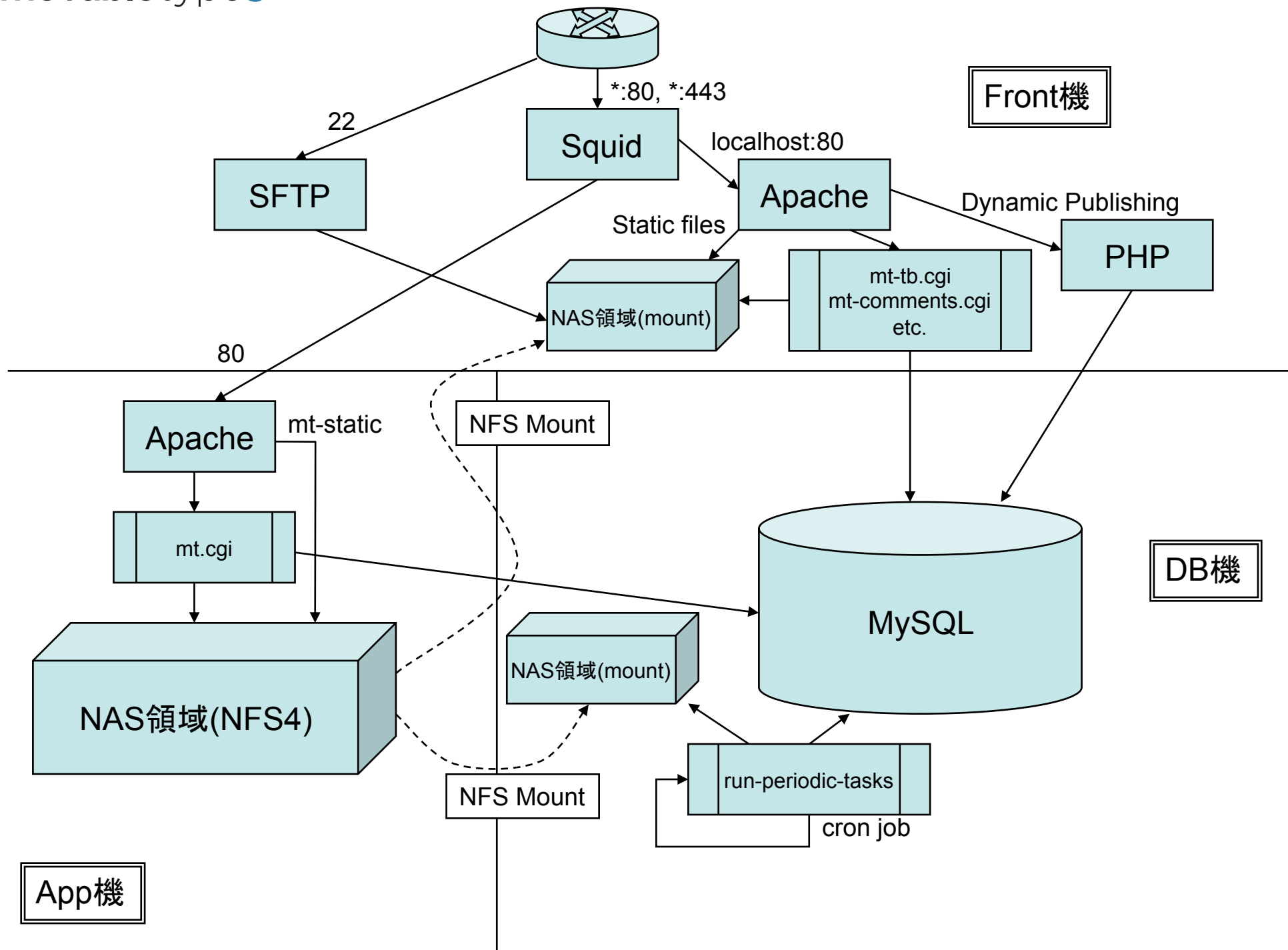
DB サーバーを別にすべきか否か

単位 (秒)	同居の環境	別サーバー環境
ログイン画面の表示までの時間	22	29
ログイン後、ダッシュボード表示までの時間	30	43
ダッシュボードからログアウト完了までの時間	22	29

参考: ある環境での測定値

パフォーマンスを意識したシステム構成

- **ウェブサーバーとアプリサーバーを分離する方が効果がやすい**
 - ウェブサーバーとアプリサーバーの Apache の設定を見直す
 - mod_deflate / mod_expires の利用
 - リバースプロキシ / キャッシュサーバーの利用



App機

Front機

DB機

まとめ

- **FastCGI / memcached の利用を検討しましょう**
- **ウェブ / アプリ / DB サーバーは適切に分離しましょう**
- **VPS / クラウドを利用し共有サーバーはやめましょう**