

Slashcode を利用した コミュニケーションポータル作成入門

VA Linux Systems ジャパン OSDN 事業部

安井 卓

<tach@valinux.co.jp>, <tach@osdn.jp>

February 14, 2002

Agenda

- ▶ コミュニケーションポータルとは
- ▶ **Slashcode** とは
 - ▶ 管理者用インターフェース
 - ▶ モデレーションシステム
- ▶ **Slashcode** のインストール
 - ▶ インストール前に
 - ▶ インストール & 初期設定
- ▶ **サイトのカスタマイズ**
 - ▶ デザインのカスタマイズ
 - ▶ スラッシュボックスのカスタマイズ
- ▶ **Slashdot Japan** の運用から
 - ▶ Web クラスタリング
 - ▶ バックアップ

A large, stylized, light gray graphic of a human eye with a grid pattern overlaid on it, serving as a background for the title.

コミュニケーション ポータルとは

A large, stylized, light gray graphic of a human eye with a grid pattern overlaid on it, serving as a background for the text.

コミュニケーションポータルって？

コミュニケーションを行うためのサイト。掲示板をはじめ、さまざまなツールがあることが多い。

- ▶ 掲示板システム
- ▶ チャットシステム
- ▶ メールとの連携
- ▶ 他サイトとの連携
- ▶ アカウント管理
- ▶ ユーザホームページ(マイページ)

Slashcode とは

Slashcode って?

コミュニケーションポータルを作成するためのシステム
<URL:<http://slashcode.com/>>

- ▶ オープンソース(GPL)
- ▶ perl で書かれており, apache + mod_perl 上で動作
- ▶ データの保存に RDBM を利用
 - ▶ MySQL, PostgreSQL などに対応
- ▶ Slashdot(<http://slashdot.org/>) を構築するために作成
 - ▶ 一般化してさまざまなサイトで利用されている
- ▶ テンプレートによるサイトデザイン・カスタマイズ
 - ▶ すべて Web 上から操作可能 ... Admin インターフェース
- ▶ プラグインによる拡張が可能
 - ▶ Admin インターフェースや日記システムが用意されている

Slashcode の特徴

- ▶ **mod_perl で動作**
 - ▶ apache 上で高速に動作する
- ▶ **インストールすれば、あとはすべて Web 上で操作可能**
 - ▶ 各ストーリーコメントの管理
 - ▶ テンプレートによるサイトデザイン
 - ▶ 変数によるサイトの動作の調整
- ▶ **セクションとトピックでストーリーを分類**
 - ▶ 各ストーリーに対してコメントがつけられる
 - ▶ 各コメントの評価をするモデレーションシステム(後述)
- ▶ **ユーザごとに表示形式やコメントの書き方のカスタマイズ可能**
 - ▶ スラッシュボックス(後述)の配置

コンテンツ分類について

- ▶ **ストーリー**
 - ▶ トップに表示される記事・ネタふり
 - ▶ 各ストーリーに対してコメントをつける
- ▶ **セクション**
 - ▶ おおまかな枠組みでストーリーを分類
 - ▶ セクションごとのトップページ
- ▶ **トピック**
 - ▶ より細かい内容に対して分類
 - ▶ トピックアイコンで表示
- ▶ **ブロック**
 - ▶ スラッシュボックスなどの情報

サイト・コンテンツ管理

Admin インターフェース

- ▶ **Web 上でサイトを管理するためのインターフェース**
 - ▶ ストーリーの管理
 - ▶ セクションの管理
 - ▶ トピックの管理
 - ▶ ユーザの管理
 - ▶ コメントフィルタ
 - ▶ テンプレートエディタ
 - ▶ 変数の管理 など...
- ▶ **ユーザに権限レベルを設定して、特定の権限だけ与えられる**
 - ▶ (小)ストーリーの投稿・管理ができる
 - ▶ : セクショントピックの管理ができる
 - ▶ (大)ユーザ・サイトデザイン・変数の管理ができる

テンプレート機能

- ▶ **デザインはすべてテンプレートで行う**
 - ▶ 各部分ごとに分かれてテンプレートがある
 - ▶ <http://www.template-toolkit.org/>
 - ▶ Perl モジュール
 - ▶ 多機能のテンプレートツールキット
 - ▶ テンプレート内部で簡単な条件分岐などが可能
- ▶ **テンプレートの内容もすべてデータベースに入れる**
 - ▶ データの一括管理
- ▶ **Web 上で編集可能**
 - ▶ ホストに shell login する必要なし

スラッシュボックスとは

各種情報を表示するためのブロック群

- ▶ ユーザが表示順やなにを表示するかカスタマイズ可能
- ▶ ログインボックス
- ▶ 投票機能のボックス
- ▶ クイックリンク
- ▶ 各サイトのヘッドラインをリンク付きで表示
 - ▶ 各サイトから rdf ファイル(XML)をダウンロードして整形

```
Freshmeat
▪ Monkey HTTP Daemon 0.2.1
  (Development)
▪ Linux Virtual Server 0.9.6
  (IPVS for kernel 2.4)
▪ Lateral Weblog System 0.1.2
  (Development)
▪ Symbion SSL Proxy 1.0.0
▪ Floppy ISBN/DSL for Linux
  2.0.1 (Stable)
▪ Videop4max 0.9.0
▪ GProfFD 7.3.5
▪ GOpchop 0.1 (Development)
▪ Mantis 0.15.12
▪ Sendpage 0.9.19 (Development)
```

rdf って?

Resource Description Framework

<http://www.w3.org/RDF/>

ポータルサイトのヘッドラインを配布するための標準フォーマット

- ▶ XML で書かれている
 - ▶ XML パーサがあれば OK!
- ▶ 海外サイトの多くで配布している
 - ▶ Slashcode サイトなら自動生成
 - ▶ 日本でもごく一部は配布中
 - ▶ Slashdot Japan でも配布(<http://slashdot.jp/slashdot.rdf>)
- ▶ サイトを作成したら、ぜひ配布しましょう

rdf の例

<http://slashdot.jp/slashdot.rdf>

```
<?xml version="1.0" encoding="utf-8"?>
:
<channel rdf:about="http://slashdot.jp/">
<title>スラッシュドット ジャパン: スラッシュドット ジャパン: アレゲなニュースと雑談サイト</title>
<link>http://slashdot.jp/</link>
<description>アレゲなニュースと雑談サイト</description>
:
</channel>

<item rdf:about="http://slashdot.jp/article.pl?sid=01/10/29/0655217">
<title>一本足走行ロボット脚</title>
<link>http://slashdot.jp/article.pl?sid=01/10/29/0655217</link>
</item>

<item rdf:about="http://slashdot.jp/article.pl?sid=01/10/29/0651214">
<title>パロアルトの中学がノートパソコンの購入を強制</title>
<link>http://slashdot.jp/article.pl?sid=01/10/29/0651214</link>
</item>
:
```

Slashcode と日本語

基本的にそのままで日本語も通るが...

- ▶ 一部, **Jcode.pm** を利用して確実に日本語が通るようにした
 - ▶ メールを送信する部分などは iso-2022-jp に変換
 - ▶ そのほかは EUC-JP で統一
- ▶ メッセージは英語しかないので, 日本語に翻訳
 - ▶ ほとんどテンプレート内部だが一部コードに埋め込み
 - ▶ 多言語化のきっかけは作ってあるがほとんど取り組みはない
- ▶ フィルタも英語環境向け
 - ▶ 日本語はとりあえずそのまま通す

Slashalikes

Slashcode をもとにしたシステムはたくさんある

- ▶ **PHP**
 - ▶ PHPSlash
 - ▶ PostNuke
 - ▶ PHPNuke
 - ▶ thatphpware
 - ▶ sips
 - ▶ TWIG
 - ▶ phpWebLog
- ▶ **Perl**
 - ▶ scoop
 - ▶ localecho
- ▶ **Zope (Python)**
 - ▶ Squishdot

モデレーション機能

掲示板サイトの悩みの種

荒らし投稿対策をどのように行うか

- ▶ **管理者による強制削除**
 - ▶ サイトの規模が小さいうちは有効
 - ▶ 規模が大きくなると、特定の管理者に負担がかかる
 - ▶ 管理者の意向に沿わないものも消されてしまう
- ▶ **投稿時に検閲**
 - ▶ 強制削除と同様の問題が発生

荒らし対策その1 ... コメントフィルタ

- ▶ **正規表現を利用したフィルタリング**
 - ▶ 特定の文字列を使用した投稿を排除できる
 - ▶ First Post...
 - ▶ アスキーアート
- ▶ **連続投稿のチェック**
 - ▶ 同じような投稿を連続して行えないようにチェックする
 - ▶ 前の投稿から 10 分以上経たないと投稿できない
 - ▶ 前のコメントと同じ文字が大量に含まれていないか

荒らし対策その 2 ... モデレーション

特定の管理者に依存しない画期的なシステム
荒らし対策というよりも、数多くのコメントの中から、良いコメント(宝石)を見つけだすためのもの

- ▶ ユーザがお互いにコメントの評価をしあうシステム
 - ▶ コメントに「スコア」をつけて管理をする
 - ▶ 決められたポイント内(5ポイント)でコメントのスコアを上げ下げする
 - ▶ 「削除」はしない
 - ▶ 読むか読まないかはユーザが判断する
- ▶ モデレータの権限がときどきユーザに与えられる

モデレーションの歴史

Slashdot.org で生まれたシステム

- ▶ はじめは管理者が削除していた
 - ▶ 投稿が増えてきて対処が困難に
- ▶ コメントにスコアをつけ、特定の人(25人くらい)にスコアを上げ下げするだけの権限を与えた
 - ▶ やはり、規模が大きくなると困難に
- ▶ モデレータを 400 人に増やした
 - ▶ 管理者が 400 人を管理できない
 - ▶ モデレータの質が落ちた
 - ▶ モデレータのパワーを制御するための機構が必要
- ▶ モデレータを制限
 - ▶ モデレーションできる数を制限
- ▶ そして、現在のモデレーションシステムが生まれた

誰がモデレートするのか

- ▶ 全ユーザがモデレーションの権利を持つことができる
- ▶ ただし、以下の条件に合致すれば
 - ▶ ログインしているユーザ(匿名ではない)
 - ▶ カルマ(後述)が一定以上あるユーザ
 - ▶ 頻繁にアクセスするユーザ
 - ▶ トップページだけ Reload するユーザはダメ
- ▶ モデレーション権限は 3 日間だけ
 - ▶ 特定のユーザが権限を持ち続けるのを防ぐ

カルマについて

カルマとは ... そのユーザのスコア(徳?)

- ▶ タレコミが採用されたときや、自分のコメントのスコアが上がったときに溜まる
- ▶ 逆に、自分のコメントのスコアが下がったときに下がる
 - ▶ 変な行動をしてると、あっというまにカルマが下がる
- ▶ カルマが高すぎてもあまり意味はなく、一定以上なら同じ
 - ▶ カルマを高くするためによい行動をする? → 発想が逆
 - ▶ 一定以上のレベルを保つためにカルマを導入しているだけ

どのようにモデレートするのか

- ▶ モデレート権限がまわってくると、各コメントごとにプルダウンメニューがつく
 - ▶ 普通(+0)
 - ▶ オフピック・フレームの元・荒らし・よけいなもの(-1)
 - ▶ すばらしい洞察・興味深い・参考になる・おもしろおかしい(+1)
 - ▶ (カスタマイズ可能)
- ▶ モデレートは一回 5 ポイントまで、3 日間のみ
 - ▶ (カスタマイズ可能)
- ▶ スコアの初期値
 - ▶ 登録ユーザのコメント = 1
 - ▶ 匿名ユーザのコメント = 0

モデレーションポイントの与え方

- ▶ ユーザ数・コメント数からプールポイントを生成
 - ▶ ユーザ数やコメント数が多いほど多くなる
- ▶ カルマが一定以上のユーザに対して、ランダムにプールポイントを割り振る
 - ▶ カルマの大小は関係ない
- ▶ プールポイントが一定以上になったユーザに対して権限を与える
 - ▶ 5 ポイント, 3 日間

Slashcode の インストール

インストール前

slashcode を実行するために必要な様々なソフトを先にインストールしておく

- ▶ apache
- ▶ mod_perl
- ▶ MySQL
- ▶ さまざまな perl module

apache,mod_perl のインストール

RedHat や Debian の場合、バイナリパッケージが用意されている

- ▶ RedHat の場合:
 - ▶ rpm -i apache_*.rpm
 - ▶ rpm -i mod_perl_*.rpm
- ▶ Debian の場合:
 - ▶ apt-get install apache libapache-mod-perl

ソースからのインストールは割愛...

<URL:http://www.din.or.jp/~bigstone/cgilab/cgitips/mod_perl.html> が詳しい

MySQL のインストール

こちらもディストリビューションでパッケージがある

- ▶ RedHat の場合:
 - ▶ rpm -i MySQL-*.rpm
- ▶ Debian の場合:
 - ▶ apt-get install mysql-server
 - ▶ potato(Debian 2.2) の MySQL は 3.22.32 なので、ライセンス (FPL) に注意
 - ▶ 必要なら unstable から最新のソースを持ってきてパッケージを作り直す

オリジナルソースは以下から:

<URL:<http://www.mysql.com/>>

perl モジュールのインストール

CPAN.pm の機能を利用して, CPAN からインストール

```
# perl -MCPAN -e shell
cpan> install Bundle::Slash
```

Debian の場合:

```
apt-get install libdbi-perl libdbd-mysql-perl libapache-dbi-perl libapache-dbilogger-perl
libnet-perl libhtml-parser-perl libxml-parser-perl libmime-base64-perl
libdigest-md5-perl liburi-perl libwww-perl libdate-manip-perl libtemplate-perl
libmail-sendmail-perl libapache-request-perl libimage-size-perl
libcompress-zlib-perl libtime-modules-perl libemail-valid-perl
libhtml-format-perl libtime-hires-perl libsched-cron-perl
```

非常に多くのモジュールに依存している

DBIx::Password のインストール

RDBM に接続するユーザ・パスワード・ホストをひとつの仮想ユーザで扱うためのモジュール

▸ パスワードなどをモジュールファイル内部に書くため, permission に注意

```
# perl Makefile.PL
What is the name of the Virtual User?
slash
What is the dbi driver? (AKA mysql)
mysql
What is the name of the database?
slashcode
What is the name of the machine that the database is on?
db-host
Is the database on any special port(you should probably just hit return)?

What is the username?
db-user
What is the password?
db-pass
```

DBIx::Password のインストール (2)

What attributes would you like to add?
(Enter nothing to skip or finish)

What is the name of the Virtual User?
(Enter nothing if you are finished adding users.)

Checking if your kit is complete...
Looks good
Writing Makefile for DBIx::Password

```
# make install
```

DB のアクセス権の設定

▶ データベースはあらかじめ作っておく

```
# mysqladmin create slashcode
```

▶ そのデータベースに対して全権限を与える

▶ ユーザ: slash

▶ パスワード: bar

```
# echo 'GRANT ALL PRIVILEGES ON slashcode.* TO slash@localhost IDENTIFIED BY "bar";' | mysql -u root
```


Slashcode のインストール

ソースの取得・展開

<URL:http://sourceforge.net/project/showfiles.php?group_id=4421>

```
# tar xvzf slash-2.2.x.tar.gz
# cd slash-2.2.x
# make
:
# make install
```

インストール先は /usr/local/slash(デフォルト)

変更するときは, make SLASH_PREFIX=/foo/bar とする

Slash サイトの立ち上げ (1)

▶ サイトの雛形を作成する

- ▶ ホスト名・権限
- ▶ どのテーマを採用するか
- ▶ どのプラグインを使うか
- ▶ 管理者ユーザ名・パスワード

```
# /usr/local/slash/bin/install-slashsite -u slash
```

▶ (実際に運用するホスト名)

What is hostname of your Slash site
(e.g., www.slashdot.org)? [your.host.name]

▶ (動作するユーザ権限)

What user would you like to run your Slash site as? [nobody]

▶ (動作するグループ権限)

What group would you like to run your Slash site under? [nogroup]

Slash サイトの立ち上げ (2)

▶ (slash ホスト名)

OK, I am planning on user your.host.name as the unique name for the Slash site. If this is not ok, you need to fill in something else here. [your.host.name]

▶ (サイトのテーマ)

Which theme do you want to use?

(*)1. slashcode "Slashcode.com theme"

Skipping theme select since you only have one theme!

Theme selected: slashcode

Slash サイトの立ち上げ (3)

▶ (プラグインの選択)

Please select which plugins you would like (* marks default).

(*) 1. Admin - "Admin Interface"

() 2. BunchaBlocks - This is a bunch of portald blocks you can add

() 3. CheesyPortal - CheesyPortal is a script to get an overall look at portal boxes

() 4. ForumZilla - "ForumZilla support"

(*) 5. Hof - "High score stuff"

(*) 6. Journal - "Journal system for users"

(*) 7. Messages - "Messaging system"

(*) 8. PubKey - "Just allows a user's pubkey to be displayed"

(*) 9. Search - Slash Search is the default search engine for Slash.

Hit 'a' to select all, otherwise select comma separated numbers or 'q' to quit

q

Plugins selected: Admin Hof Journal Messages PubKey Search

▶ (symlink かコピーか)

Would you like to install all the files as symlinks to the original? (If not, each file will be copied to your Slash directories). [Y]

Slash サイトの立ち上げ (4)

- ▶ (サイト管理者の Web アカウント)
Create a name for the site's admin account (8 characters or less). [nobody]
- ▶ (サイト管理者の Web パスワード)
Create a password for the site's admin account. ('QUIT' exits):
- ▶ (サイト管理者のメールアドレス)
What is the email address of the account? [nobody@your.host.name]

これで, `/usr/local/slash/site/your.host.name` 以下にファイルが作成され, データベースに初期データが挿入される

slash site の起動

apache の設定

- ▶ `/usr/local/slash/slash.conf` を `httpd.conf` で Include

```
Include /usr/local/slash/httpd/slash.conf
```

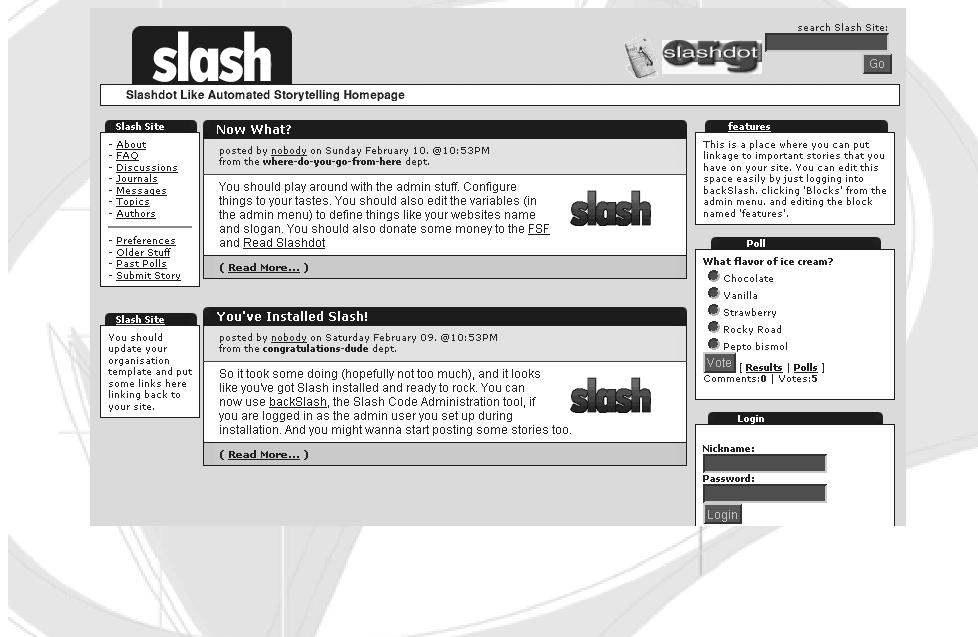
slashd の起動

- ▶ `/etc/init.d/slash start`

apache の起動

- ▶ `/etc/init.d/apache start`

最初に表示されるサイト



コンテンツ管理

コンテンツ管理について

基本的にすべて Web 上で行うことができる
まず、admin アカウントでログインする



A screenshot of a login form. The form has a title "Login" and a "Login" button. There are two input fields: "Nickname:" with the text "nobody" and "Password:" with masked characters. Below the form is a link "(create a new account)".

初期設定時に設定した ID とパスワードでログイン

Admin メニュー

一般ユーザにはない、管理者専用メニューがでてくる

[Home](#) | [Topics](#) | [Stories](#) | [New](#) | [1 Submissions](#) | [Poll](#) | [Blocks](#) | [Colors](#) | [Filters](#) | [Sections](#) | [Users](#) | [Keywords](#) | [Templates](#) | [Vars](#) | [Site](#) | [Help](#)

- ▶ Home : トップページへ
- ▶ Topics : トピックリスト
- ▶ Stories : ストーリーリスト
- ▶ New : 新規ストーリーの作成
- ▶ x Submissions : ユーザからの投稿
- ▶ Poll : 投票の設定
- ▶ Blocks : ブロック(スラッシュボックスなど)の設定
- ▶ Colors : サイトの色の設定
- ▶ Filters : コメントフィルタの設定
- ▶ Sections : セクションの設定
- ▶ Users : ユーザの管理
- ▶ Keywords : キーワードの設定
- ▶ Templates : デザインテンプレートの設定
- ▶ Vars : 変数の設定
- ▶ Site : サイト情報の表示
- ▶ Help : ヘルプ画面の表示

New: 新規ストーリーの作成

save preview News Articles nobody

Title ①

Dept

always Display Comments Enabled 2002-02-12 13:0

fixquotes autonode story shortcuts ③

Intro Copy

save preview

Extended Copy

1. セクショントピック・著者を設定
2. 表示設定(表示するしない-セクションページにのみ表示)
3. コメント設定(コメント許可-不許可)
4. ストーリーを書いて、「preview」してみて問題なかったら「save」

Poll: 投票の設定

New Poll

The Question (followed by the total number of voters so far)

Poll Topic (this doesn't do much right now but in the future it will)

News トピック

The Answers (voters) 答え ①

Story ID (if this matches a story's ID it will appear with the story)

ストーリーと連携させるならsidを書く

SlashBox Poll (if checked this will be the poll used in slashboxes)

(appears on homepage)

Save トップのスラッシュボックスに表示

1. 投票のネタ・答えを書く
2. ストーリーと連携させるなら、ストーリー ID(sid) を書く
3. トップページのスラッシュボックスにするかどうかチェック
4. 「save」

意図的に数値を変更することも可能 ^^;

Block: ブロックの設定

Static and Color Blocks userlogin Edit Block Delete Block

Portald Blocks poll Edit Block Delete Block

Block ID userlogin ブロック ID (一意)

Title Login

Seclev 1000 セキュリティレベル

Type static タイプ

Section index 表示するセクション

Ordernum 4 表示順

Portal - check if this is a slashbox. スラッシュボックスか？

Retrieve RDFをとってくるか？

URL

RDF

Description

Block

Save Block Revert to default Save as default (Make sure this is what you want)

1. 編集するブロックを選択(新しいブロックなら「Create New」を)
2. 各種情報を記入
3. 「save」

Sections: セクションの設定

save preview News Articles nobody

Title

Dept

Always Display Comments Enabled 2002-02-12 13:0

fixquotes autonode story shortcuts

Intro Copy

save preview

Extended Copy

1. セクショントピック・著者を設定
2. 表示設定(表示する・しないセクションページにのみ表示)
3. コメント設定(コメント許可・不許可)
4. ストーリーを書いて、「preview」してみても問題なかったら「save」

Users: ユーザの管理

UID/NetID/Nickname: <input type="text"/>	User Info	Edit User	Change Password
	Customise Homepage	Customise Comments	
Seclev	<input type="text" value="10000"/>	ユーザのセキュリティレベル	
Author flag (user is an author)	<input checked="" type="checkbox"/>	著者フラグ	
Default points (threshcode)	<input type="text" value="1"/>	デフォルトのスコア	
Read-only 読むだけに制限	<input type="checkbox"/>	Reason: <input type="text"/>	
Comments	<input type="checkbox"/>	Reason: <input type="text"/>	
Submissions	<input type="checkbox"/>	Reason: <input type="text"/>	
Banned サイトへのログイン禁止	<input type="checkbox"/>	Reason: <input type="text"/>	
Banned from Slash Site	<input type="checkbox"/>	Reason: <input type="text"/>	
Expire User ユーザの削除	<input type="checkbox"/>	Reason: <input type="text"/>	
Real time black list ブラックリスト	<input type="checkbox"/>	Reason: <input type="text"/>	
Save Admin: <input type="button" value="Save Admin"/>			

管理したいユーザの ID やニックネームを選択 → 「Edit User」

Seclev: セキュリティレベル

一般ユーザ : 1

ストーリーの管理ができるユーザ : 100

ブロックピック・セクションの管理ができるユーザ : 1000

サイトデザインなどすべての管理ができるユーザ : 10000

著者フラグ: 著者にするときはチェック

サイトのカスタマイズ

Colors: サイトの色の設定

Select the color block to edit: colors Edit Colors ①

Foreground color 0	#FFFF	Foreground color 0	
Foreground color 1	#222222	Foreground color 1	
Foreground color 2	#111111	Foreground color 2	
Foreground color 3	#DDDD	Foreground color 3	
Foreground color 4	#999999	Foreground color 4	
Background color 0	#DDDD	Background color 0	
Background color 1	#FFFFFF	Background color 1	
Background color 2	#DDDD	Background color 2	
Background color 3	#880000	Background color 3	
Background color 4	#BBBB	Background color 4	

Preview Save Colors ② Revert to saved Revert to default Save as default (Make sure this is what you want!)

1. どのセクションの色を変更するか選択(デフォルトは colors)
2. 色の記入(RGB)
3. 「save」

注意: 設定した色は、apache を再起動しないと反映されない

Vars: 変数の設定

Select Variable Name: adminmail Select Var ①

Variable Name: adminmail

Value: nobody@hoge.arege ②

Description: All admin mail goes here

Save ③

1. 編集する変数を設定
2. 変数の値を変更
3. 「save」

代表的な変数:

- ▶ adminmail = 管理者のメールアドレス
- ▶ allow_anonymous = 匿名投稿を許可するかどうか
- ▶ allow_moderation = モデレーションシステムを有効にするかどうか
- ▶ approvedtags = コメントでの使用を許可する HTML タグ
- ▶ http_proxy = rdf をとってくる時のプロキシ
- ▶ use_dept = 「dept.(部門)」を使うかどうか
- ▶ send_mail = メッセージメールを使うかどうか
- ▶ sitename = サイトの名前
- ▶ slogan = サイトのキャッチフレーズ
- ▶ smtp_server = メールを送る時のサーバ

その他, いろいろなパラメータ指定など

Template: サイトのデザインの変更

各部分がそれぞれテンプレート化されており、そのテンプレートを変更することで、サイトのデザインを変更することができる。

▶ テンプレートの名称

- ▶ 「テンプレート名;対象ページ;セクション」
- ▶ セクションが「default」のものはデフォルトで使われる
- ▶ 例: index;index;default, fancybox;misc;default

Section: default (1) Select section
Page: misc (1) Select page
Templates: currentAdminUsers (68) Edit Template Delete Template
Create a new template: Create a new template (2)

1. セクション・ページ・テンプレート名を選択
2. 「edit template」

Template: サイトのデザインの変更 (2)

Save template in section: default テンプレートのセクション
Template section (new template section?) default
Save template in page: misc テンプレートページ
Template page (new template page?) misc
Template Name: fancybox テンプレート名
Title:
Seclev: 10000 セキュリティレベル
Description: Creates a nice box with title and contents
Template: [* IF center *]<CENTER>[* END *]
[* width = width || 200 *]
<TABLE WIDTH="{% width %}" BORDER="0" CELLSPACING="0" CELLPADDING="0">

1. テンプレートのセクション・ページ・名前を設定(新規作成でなければ自動)
セキュリティレベルを設定
2. テンプレート本文を編集
3. 「save」

テンプレートの書き方

例: テンプレート fancybox, misc, default

- ▶ スラッシュボックスを表示する部分

```
[% width = width || 200 %]
```

```
[% IF center %]<center>[% END %]
```

```
<table width="[% width %]" border="0" cellpadding="1" cellspacing="0">
```

```
<tr valign="top" bgcolor="[% user.bg.3 %]"><td>
```

```
<font size="+1" color="[% user.fg.0 %]"><strong>[% title %]</strong></font>
```

```
</td></tr>
```

```
<tr><td bgcolor="[% user.bg.4 %]">
```

```
<font size="-1">
```

```
[% contents %]
```

```
</font>
```

```
</td></tr>
```

```
</table>
```

```
[% IF center %]</center>[% END %]
```

```
<br>
```

テンプレートの文法

- ▶ [% %] で囲まれている部分がテンプレートのコードや変数
- ▶ それ以外は、そのまま表示される
- ▶ 文の区切りは「;」か [% %]

よく使われる構文

- ▶ [% vars %] ... 変数の内容をそのまま表示
- ▶ [% PROCESS template %] ...他のテンプレートを挿入
- ▶ [% IF ... ELSE ... END %] ... IF 文
- ▶ [% SWITCH ... CASE ... END %] ... SWITCH 文

デフォルトで使える変数

- ▶ user ... ユーザ情報
- ▶ form ... フォーム入力の内容
- ▶ constants ... 「変数」
- ▶ env ... 環境変数

template-tools

テンプレートを扱うコマンド

- ▶ テキストに書き出してエディタで編集できる

- ▶ データベースに入っているテンプレートをダンプ

```
# template-tools -u slash -d
:
# ls
admin;menu;default          listReadOnly;users;default
changePasswd;users;default  listSections;sections;default
comment_submit;comments;default listTopics;topics;default
:
```

- ▶ 編集したテキストをデータベースに入れる

```
# template-tools -u slash -s template;file;name
```

バックエンドデーモン slashd

rdfの取得やメッセージ配信メールなど、定期ジョブを起動するためのデーモン。

- ▶ 定期ジョブはタスクで管理
- ▶ /usr/local/slash/site/your.host.name/tasks 以下にタスク毎の実行スクリプトがある
- ▶ cron 風の起動時刻設定
 - ▶ `$task{$me}{timespec} = '13,43 * * * *';`

Slashcode の工夫

Slashcode の工夫

Slashcode では, 運用面での利便性・パフォーマンスを考えて, さまざまな工夫が入っている.

- ▶ 高速化
 - ▶ CGI で運用するのは無謀
 - ▶ mod_perl などを利用する
 - ▶ DB へのアクセスを減らすキャッシュ
- ▶ クラスタ構成を想定したシステム設計
 - ▶ コードとデータの分離
 - ▶ データの一括管理

高速化

CGIは実行ごとにプロセスを生成するので遅い

- ▶ プロセス生成は結構重い処理
- ▶ 頻繁にアクセスのあるページを CGI で書くのは自殺行為

プロセスを生成しない → デーモン化？

- ▶ httpd がデーモンで動いている
- ▶ httpd (apache) に組み込んでじゃえ!



mod_perl mod_ruby php

mod_perl について

<http://perl.apache.org/>

- ▶ Apache のモジュール
- ▶ Apache のプロセスとして動作するので高速
 - ▶ 10 倍以上のスピード?
- ▶ CGI と同様のインターフェースを備えているため、CGI からの移行が楽
- ▶ 逆にそれが命取りにもなる
 - ▶ プログラムが変だと Apache まで影響を受ける

データのキャッシュ

- ▶ `mod_perl` のスクリプトは基本的に再ロードしない
- ▶ 変数も初期化しない
- ▶ それを逆手にとって、グローバル変数内にデータを保持
 - ▶ データベースへのアクセスが減る
 - ▶ 高速に表示ができる
 - ▶ データベースサーバへのトラフィックが減る
 - ▶ `fork` したプロセスが死ぬまで有効
 - ▶ ときどき内容が消える?
 - ▶ 内容をチェックして抜けていたら取りに行く

なにをキャッシングしておくか

あまり内容が変わらないもの
頻繁にアクセスするもの
あまりアクセスがなかったり、内容が変わったりするものは NG

- ▶ `Slashcode` では、ストーリーやテンプレート

キャッシュの有無でサーバの負荷がだいぶ違う(5倍以上?)

キャッシュを有効に活用するために

- ▶ サーバにたくさんメモリを搭載する
 - ▶ 各プロセスごとにキャッシュを保持するので、プロセスのサイズが大きくなる
 - ▶ 高速に動作するので同時に複数のプロセスが稼働する
 - ▶ そのぶんだけメモリが必要
- ▶ Apache の設定で最適化
 - ▶ **MaxClients ... 最大同時接続クライアント数**
 - ▶ 上げすぎて失敗 デフォルト値(150)のまま
 - ▶ **MaxRequestPerChild ... forkしたプロセスがこの数だけ処理して死ぬ**
 - ▶ キャッシュを生かすために上げる(500-1000)
 - ▶ メモリリークなどの心配があるのであまり上げすぎない
 - ▶ **KeepAliveTimeout ... キープアライブのタイムアウト時間**
 - ▶ 短く(3秒)して、次の要求をすぐに受けられるようにする

セキュリティ対策

シンプルに 確実に

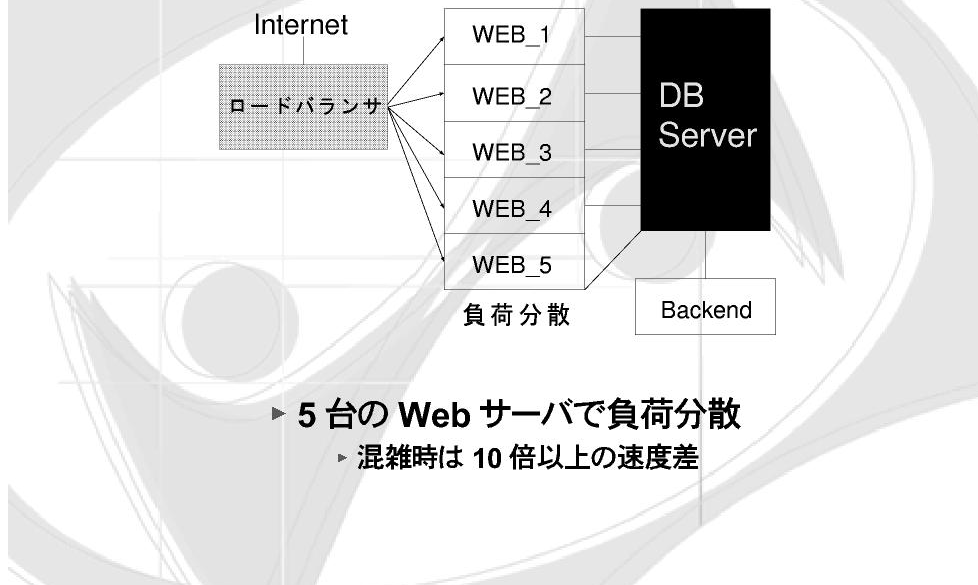
- ▶ 複雑なことはしない
 - ▶ 複雑にすると、管理が煩雑になる
 - ▶ 不具合の原因追及がしにくくなる
- ▶ 不要なポートは開けない
 - ▶ 必要なサービスだけ立ち上げる
- ▶ セキュリティ情報を常にチェックする
 - ▶ 使用しているソフトに穴が見つかったらすぐにパッチをあてる
- ▶ Firewall やロードバランサを入れてプライベートに押し込む
 - ▶ 不要なホストの情報を外に出さない

運用面での工夫

Slashdot Japan の運用から

- ▶ **Slashdot Japan は 2001 年 5 月にオープン**
 - ▶ slashcode 2.0 ベースでスタート
 - ▶ このときはデータベースサーバと Web サーバの 2 台体制
 - ▶ オープン時から多少重めだったが、徐々にアクセスが増えていった
 - ▶ ピーク時はロードアベレージが 30 を越えた
- ▶ **緊急にロードバランサを準備**
 - ▶ 8 月にクラスタ体制に移行
 - ▶ 負荷が負荷を呼ぶ状態を脱して、軽快動作
 - ▶ サーバ全体を private network に入れたため、安全

ロードバランスの状態



データのバックアップ体制

- ▶ データセンターにあるため、自分でテープバックアップするのは困難
 - ▶ ハードディスクにバックアップを取る
- ▶ データはすべてデータベースに入っているので、とりあえず `mysqldump` で一発
- ▶ それ以外(コードや設定など)は `rsync` や `vbackup` でバックアップ

mysqldump でバックアップ

- ▶ mysqldump は MySQL のデータを SQL 文で書き出す
- ▶ その SQL を mysql コマンドに通せば元に戻る

- ▶ **使い方:**

- ▶ 実行するユーザの ~/.my.cnf に以下のように書いておく

```
[mysqldump]
user = username
password = pass
host = host_of_mysql
```

- ▶ mysqldump を実行
mysqldump database_name > sqlfile.sql
- ▶ cron で定期的に行う

vbackup で差分バックアップ

ファイルシステムをそのまま HDD → HDD でコピーするバックアップツール

- ▶ <http://www.rommelwood.de/~martin/vbackup/>
- ▶ ハードリンクを利用した差分バックアップツール
- ▶ 普通のファイルシステムなので、ls cp などのツールがそのまま利用できる
- ▶ ハードリンクを利用するため、容量の節約になる
 - ▶ ハードリンクが利用できないファイルシステムでは利用不可
- ▶ つかいかた
 - ▶ vbackup [もとディレクトリ] [バックアップ先]

vbackup で差分バックアップ (2)

例:

```
% vbackup /var /backup/var  
% ls /backup/var  
2001-10-27@03:17:44 2001-10-28@03:16:28 2001-10-29@03:17:09
```

各日付・時間のディレクトリを作り, その中にファイルが格納される

```
% vbackup arege:/var /backup/var  
% ls /backup/var  
2001-10-27@03:17:44 2001-10-28@03:16:28 2001-10-29@03:17:09
```

rsh/ssh を利用して, リモートのバックアップもとれる

さいごに...
RDBM の選択

なぜ MySQL を利用するか

Slashcode は各種 RDBM をサポートしている

- ▶ MySQL
- ▶ PostgreSQL
- ▶ Oracle(?)

なぜそのなかから MySQL を選択したか

- ▶ Slashcode + MySQL の方が実績がある
- ▶ slashdot.org でも MySQL を利用している

MySQL は高速かつ使いやすいから

MySQL の特徴

▶ **高速**

- ▶ ベンチマークを見ると他を圧倒
- ▶ <http://www.mysql.com/information/benchmarks.html>

▶ **信頼性**

- ▶ Yahoo! など超巨大サイトでも利用されている

▶ **管理が複雑ではない**

- ▶ かんたんなコマンドで管理が可能
- ▶ GUI (MySQL GUI) もある
- ▶ ドキュメントが充実している

MySQL に適した場所は?

- ▶ 単純な問い合わせが多い
- ▶ とにかく高速で動作することが必要
- ▶ データの更新が頻繁ではない
- ▶ 多くの機能を必要としない

Web は上記の条件をみたしており
MySQL に最適な環境

END