手軽で柔軟な ML アーカイバ msgcab の設計と実装

上野乃毅　田中哲

独立行政法人 産業技術総合研究所 情報技術研究部門

フリーソフトウェアの開発と ML

- 電子メール
  - 昔も今もコミュニケーションの要
- メーリングリスト(ML)
  - 開発上の議論や意思決定の場
  - 過去の議論を追うのが困難な場合がある

ML アーカイバ

ML のメールをウェブで
閲覧可能にするツール

従来の ML アーカイバ

- サービス
  - Gmane, MARC, blade (Ruby)
- ツール
  - Hypermail, MhonArc, Pipermail, w3ml

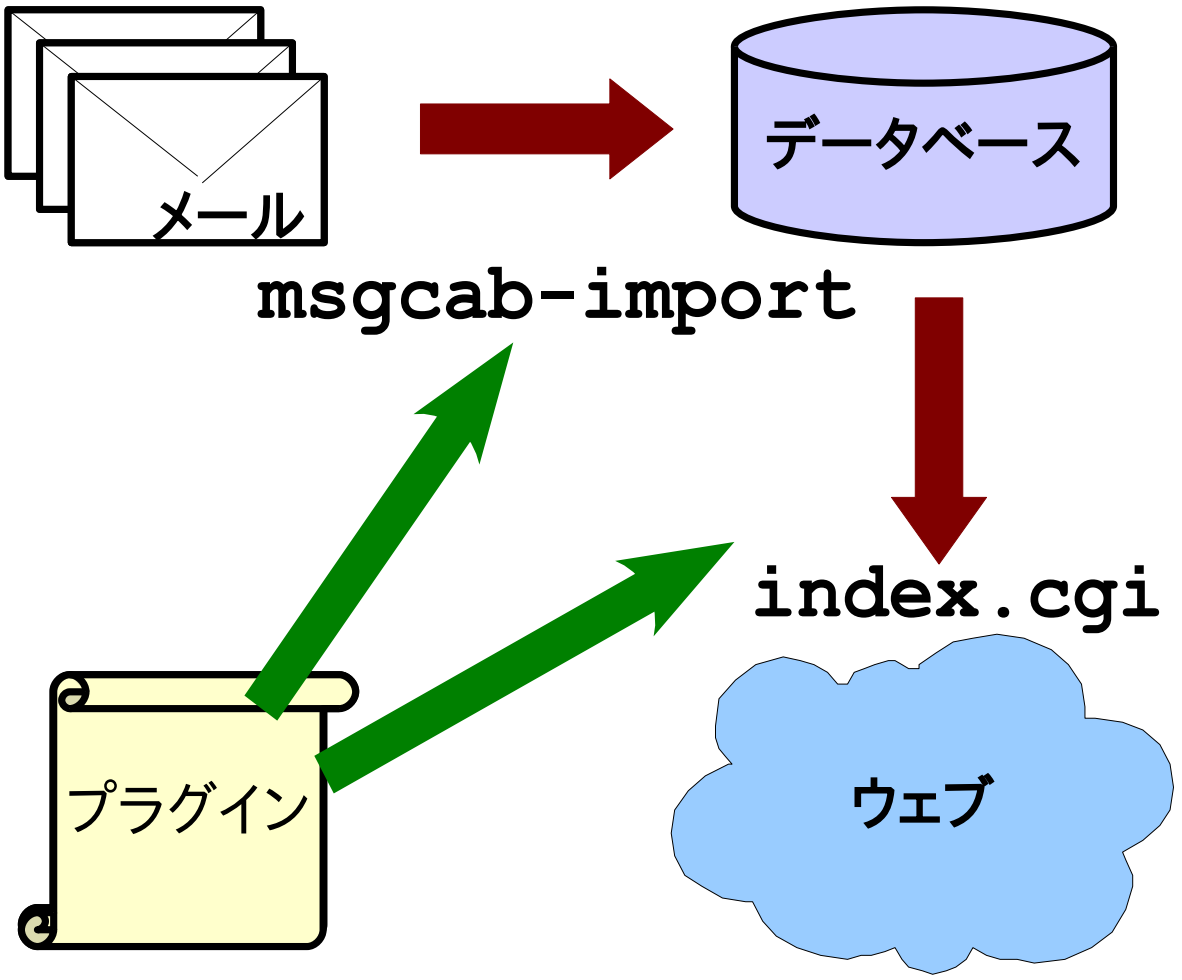
従来の ML アーカイバ

- サービス
  - ソースコードが非公開
  - 導入の敷居が高い
- ツール
  - 機能が乏しい
  - 拡張の余地が少ない

そこで msgcab

- 導入が簡単
  - 設定ファイル不要
- ML の自動識別・分類
- 典型的なメールボックス形式に対応
  - MH, mbox, Maildir から自動識別
- プラグインで拡張可能

msgcab の動作



デモ

設計方針

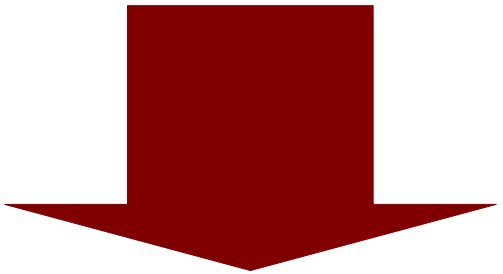
1. 議論の追跡可能性
2. 拡張可能性
3. スケーラビリティ

把握が困難な議論

- ２つの ML で並行に進む議論
  - 全文検索が有効だが、不十分
- 長期に渡る議論
  - 月毎・100通毎にスレッドが途切れる

msgcab では、

- ML を横断的に扱う
- 過去の全メールからスレッドを復元



頑健で高速なスレッド復元
アルゴリズムが必要

スレッド

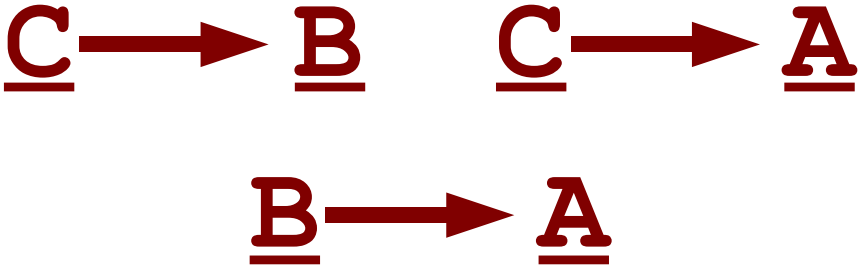
メールの参照関係を
木構造で表わしたもの

全メールの親が特定できれば
スレッドが復元できる

メールの参照関係

From: bob@example.com
To: foo-devel@example.com
References: A B
Message-ID: C

ボブです。どうみてもバグです。
本当にありがとうございました。



スレッドの復元の難しさ

- MUA のバグ
  - ヘッダが正しく付加されない
    - **`References:`** の順序が変, etc.
- ML ソフトウェアや MTA のおせっかい
  - ヘッダ部の書き換え
- メールの配送順序が前後

既存のアルゴリズム

References:の後ろから順に親を探す

- 頑健性×
  - はぐれたメールを探すのが困難
  - **References:**の要素の順序に依存
- 速度◎

既存のアルゴリズム(2)

Zawinski のアルゴリズム

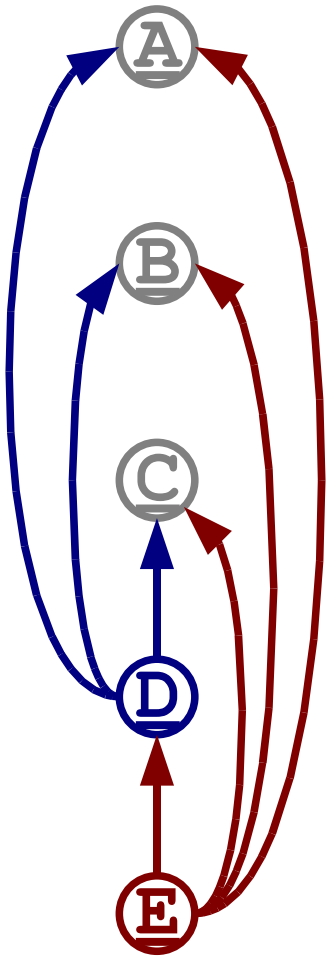
- 頑健性◎
  - 実メールの統計に基づいたアルゴリズム
  - さまざまなケースに対応
- 速度△
  - インクリメンタルに更新できない
    - 10000通のメールを処理したあとに10001通目のメールが届いた場合、やり直し

新たなアルゴリズム

akr のアルゴリズム

- 頑健性○
  - **`References:`** の順序に依存しない
- 速度◎
  - インクリメンタルな更新に対応

akr のアルゴリズム



$R(m)$: m の直接の参照
$P(m)$: m の親候補

$$R(\mathtt{E}) = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}$$
$$P(\mathtt{E}) = R(\mathtt{E}) - R(R(\mathtt{E}))$$

親の親は自分の親ではない

※論文のアルゴリズムを若干改良しました。

akr のアルゴリズム(2)

- メールが未到着な場合
  - $P(\boldsymbol{m})$ の計算過程で出会った未到着なメールの集合$N(\boldsymbol{m})$ をデータベースに保存
  - 届いたら再計算
- インクリメンタルな更新
  - $P(\boldsymbol{m})$ をデータベースに保存

設計方針

1.議論の追跡可能性
2.拡張可能性
3.スケーラビリティ

プラグイン機構

- msgcab に機能を追加する仕組み
- 標準的な拡張機能の実装

cite	メールの引用部分の強調表示
estraier	メールの全文検索
face	送信者のアイコンを表示
link	メールの本文中の URL の置換
mask_mail_addre	メールアドレスの隠蔽
rss	RSS によるサマリの生成

プラグインの構成

- **`foo.webapp.rb`**
  - CGI スクリプトから呼ばれた場合の処理
- **`foo.cli.rb`**
  - コマンドラインから呼ばれた場合の処理
- 追加の画像ファイルなど

テンプレート ライブラリ htree

- デザインとロジックの分離
- XSS (Cross Site Scripting) 対策
  - HTML の構文木を対象に展開処理

テンプレート ライブラリ htree (2)

- HTML 要素の属性に処理を記述

```
<elem _attr_name="expr">content</elem>

<elem _text="expr">dummy-content</elem>

<elem _if="expr" _else="mod.name(args)">
  then-content</elem>

<elem _iter="expr.meth(args)//vars">content</elem>

<elem _call="mod.name(args)">dummy-content</elem>

<elem _template="name(vars)">body</elem>
```

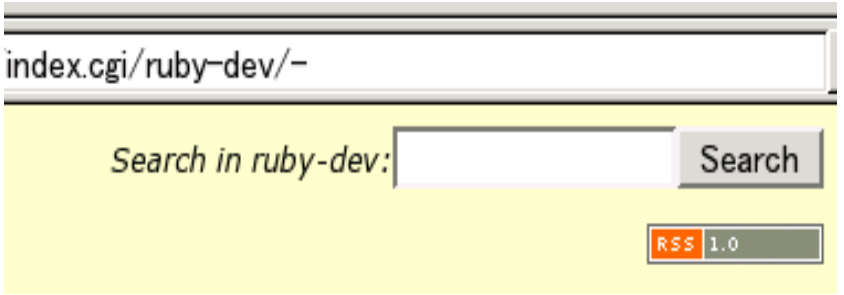
プラグインと htree

- 本体のテンプレートを操作したい
  - 例:検索フォーム・RSS のアイコン
- 本体のテンプレートをコピペ・変更
  - メンテナンスが大変
  - 他のプラグインと共存できない

アンカー

テンプレートにテンプレートを
埋め込む仕組み

アンカーの定義




```
<div _template="replace">
  <form _attr_action='view.uri("/folder")'>
    <span _text='"Search in #{folder.name}:"/>
    <input type="text" size="16" name="q"/>
    <input type="submit" value="Search"/>
  </form>
  <span _call="next_anchor.replace"></span>
</div>
```

アンカーの呼び出し

```
<html>
  <head>
   <title>msgcab: summary</title>
  </head>
  <body>
    <div class="header">
      <span _call="replace(:header, self)"/>
    </div>
    ...
  </body>
</html>
```

設計方針

1. 議論の追跡可能性
2. 拡張可能性
3. スケーラビリティ

メールの格納方式

- 単一ファイル (mbox, mbx)
  - 壊れやすい
  - 復旧が困難
- フラットなディレクトリ (MH, Maildir)
  - 大量のファイルを格納すると、open(2)の呼び出しに時間がかかる

新しい格納方式 MailTree

- メールに通し番号を付け、桁数で分類
  - 0 ... 99 番目のメール

    1/{00...99}
  - 100 ... 9999 番目のメール

    2/{01...99}/{00...99}
  - 10000 ... 999999 番目のメール

    3/{01...99}/{00...99}/{00...99}

ディレクトリの深さの上限

$$\lfloor \log_{100}(n) \rfloor + 1$$

実装

- Ruby で実装 (約4000行)
- ライブラリ
  - Ruby/DBI または SQLite/Ruby
  - RubyMail
  - htree
  - webapp

評価

1.基本的な性能

10,000通のメールを一括で取り込む

200通のメールをスレッドを表示

2.インクリメンタルな更新の性能

100通単位でメールを取り込む

総メール数と所要時間の関係を調べる

評価に使用した環境

CPU	Mobile Athlon(tm) 64 3400+
メモリ	2GB
OS	Linux 2.6.8
RDBM	SQLite 2.8.16
Ruby	1.8.4

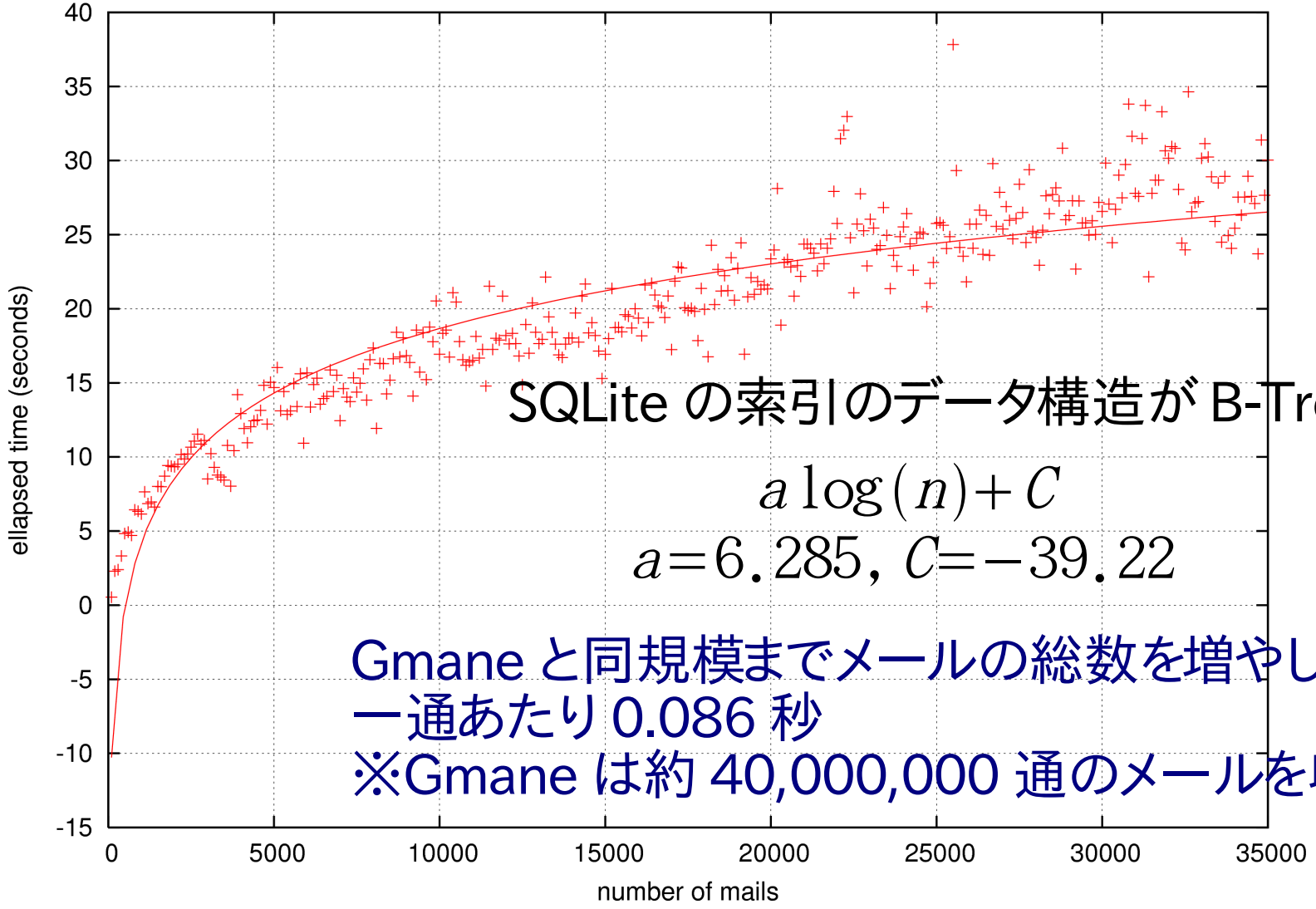
評価対象の ML

- Wanderlust 関連の ML
  - wl, wl-en
- Ruby 関連の ML
  - ruby-list, ruby-dev, ruby-talk

基本的な性能

- 10,000通を一括で取り込み
  - 約16分12秒
- 200 通のメールをスレッド表示
  - 約2秒

インクリメンタルな更新の性能



SQLite の索引のデータ構造が B-Tree

$$a\log(n)+C$$

$$a=6.285,\ C=-39.22$$

Gmane と同規模までメールの総数を増やした場合、
一通あたり 0.086 秒
※Gmane は約 40,000,000 通のメールを収録

応用

- 他のウェブアプリケーションと重ね合わせ (mushup)
  - JavaScript の CSI (Client Side Include) でスレッドを blog に引用
- 自動テストシステムと組み合わせたバグ追跡システム Couya

今後の課題

- 認証
  - 未読管理などに必要
  - OpenID などのシングルサインオン認証
- 専用クライアント
  - Emacs インターフェイス
- スレッドの復元アルゴリズムの改良
  - **`References:`** や **`Message-ID:`** がない場合にも、メールの本文の引用情報を利用して復元

まとめ

- 導入が簡単でプラグインによる拡張が可能な ML アーカイバ msgcab の内部構造を解説した
- http://msgcab.nongnu.org で配布中
  - 2006 5/29 に 0.0.3 をリリース