



ご自宅用ツール間同期メカニ ズム兼ストレージのご提案

Pusherの中の人も絶賛（予定）

seki@druby.org



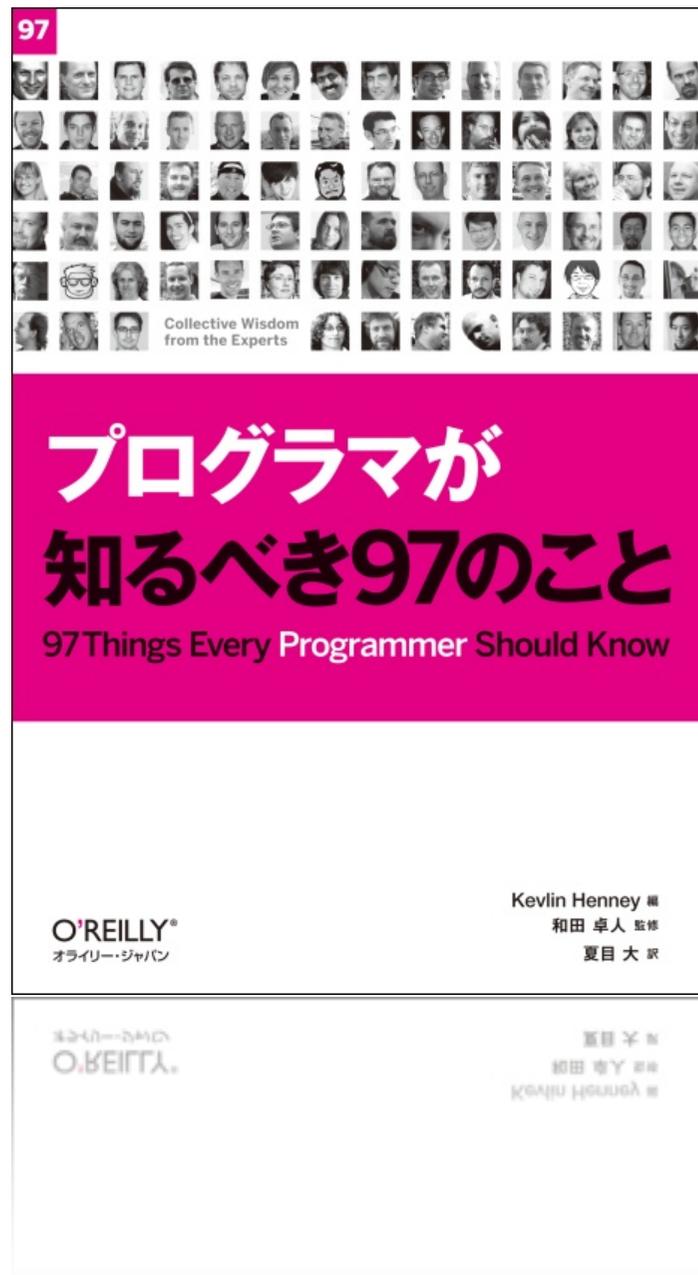
今日の話

- 重要なこと
- とても重要なこと
- 私について
- Dropのこと



重要

- 日本語版に寄稿しました
- 宣伝しなくても売れそう



とても重要

- 2005年夏
- まだ初刷買えます!
- 初刷5周年!!



さらに重要

- 2010年秋
- 電子書籍版
- 永遠に初刷？



● ● ● 私について

● ちゃんとした会社員

● toRuby

○●● ちゃんとした会社員

- わりと大きな会社
- プロの無職

●●● ちゃんとした会社員

- わりと大きな会社
- プロの無職



○ ● ● toRuby

- 西那須野公民館を中心に活動
 - とちぎRubyの勉強会
 - とちぎRuby会議01,02
 - とちぎテストの会議（連番なし）

● ● ● メモのご用意を

● とちぎRuby会議03

○ 2011-2-26

○ 50回達成記念

○ スピーカー募集中（自薦歓迎）

○ ● ● RubyKaigi x 5

● 2006 dRuby, Again.

● 2007 Answering dRuby and Rinda

● 2008 erbを偲んで

● 2009 偉大なBigTableとぼくのおもちゃ

● 2010 RWikiと怠惰な私の10年間

○ 遺産で登壇するのはそろそろ限界か。 ◀いまここ



今日の話



Drop。



Rindaの実用的？な応用



いろんなモノに見えるライブラリ



Drop

Pusherの中の人も絶賛!!



○ ● ● いろいろなものに見える

- プロセス間の同期・変更通知の機構
- オブジェクトの保存場所
- バッチ処理のミドルウェア
- インタラクティブ処理のミドルウェア
- Twitterのタイムライン

● ● ● どこからきたの？

- PRb, KoyaなどのOODBへの挑戦と失敗
- Rindaのアプリケーションの経験
- 分散ファイルシステムへの憧れ
- 2009 RubyKaigiでの講演のテーマ
 - 世界をコントロールブレイク!

○ ● ● できること

● 覚える

● 思い出す

● 待つ

○ 忘れられない

覚える

- キーを指定しないで保存するKVS_{www}
- 受け付けた時刻がキー
 - 保存操作の戻り値がキー
- 思い出す補助となるタグを付加できる



キ一

- 時刻から変換された整数
- Dropの中でユニーク
- Drop内部ではオブジェクトは時系列に並ぶ

思い出す

- 時刻を指定して思い出す
- あの時よりも後に追加されたものを思い出す
- 時間軸にそって前後にブラウズ
- タグでフィルタできる

○ ● ● 待つ

- 指定した時刻よりも後に追加されたオブジェクトがないときはブロック
- 新しいオブジェクトが追加されたら再開する
- タグでフィルタできる

○ ● ● タグ

- Stringです
- いくつでもつけられるけど、覚えるときにしか指定できない

○ ● ● 忘れられない

- オブジェクトの更新や削除はできない
- 過去は変更できないので、並行処理のイヤなことを気にしないですむ
- トランザクション？

●●● 使い方のヒント

- あまりにも唐突な仕組みなので

● ● ● なにに使うの？

- 某所向けRWiki全文検索
- 某所向けメッセージングシステム
- 自分用メモ
- irbで作ったオブジェクトをメモとか



メモの例

```
irb(main):041:0> pp drop.head(2)[0]
[1289928806763125,
 {"メタグロス"=>[80, 135, 130, 95, 90, 70],
 "ベロベルト"=>[110, 85, 95, 80, 95, 50],
 "パッチール"=>[60, 60, 60, 60, 60, 60],
 "ココドラ"=>[50, 70, 100, 40, 40, 30],
 "エテボース"=>[75, 100, 66, 60, 66, 115],
 "ピカチュウ"=>[35, 55, 30, 50, 40, 90],
 "ナックラー"=>[45, 100, 45, 45, 45, 10],
 :
 :
 "オタチ"=>[35, 46, 34, 35, 45, 20],
 "マッギョ"=>[109, 66, 84, 81, 99, 32],
 "ドクケイル"=>[60, 50, 70, 50, 90, 65],
 "ムクホーク"=>[85, 120, 70, 50, 50, 100],
 "ブラッキー"=>[95, 65, 110, 60, 130, 65],
 "ハンテール"=>[55, 104, 105, 94, 75, 52]},
 "種族値"]
=> nil
```

とりあえず保存する

- 作業中のオブジェクトをとりあえずメモしておく
- キーとか気にせず気軽に保存
- OODBでObjectをnewする感覚
- おおよその時刻がわかればなんとかなる
- ロディアにメモする感じ

write

- write(obj, *tags)
 - オブジェクトを追加する
 - 分類、参照のためのタグを指定できる
 - 戻り値はkey

ユニークなキー

- 重複の無い値を安価に作るのは大変
 - 考えるのめんどうだった > Koya, PRb
- ある瞬間にDropが気づける事象は一つ
- 時刻は単調増加でいるいる便利

○●● ストリームとして使う

- ある時刻より後のオブジェクトを取得するAPI
- アプリケーション側で外部イテレータを簡単に書ける
- Dropからコールバックする必要がない
- RMIを減らす工夫

read

- `read(key, n=1, at_least=1, timeout=nil)`
 - 俺が知っていることより新しいことを
n個教える
 - `at_least`個揃うまで報告するな

● ● ● 典型的なブラウザ

```
def drop_each(drop, key=0, &blk)
  while true
    ary = drop.read(key, 10, 1)
    ary.each(&blk)
    key = ary[-1][0]
  end
end
```

● ↑実験してないので注意

●●● バッチ処理

- バッチ処理はたいてい失敗する
 - とういかやりなおしたいことがある
- ブラウズの中断や再開が容易
- 消えないQueueの利点

○●● 待ち合わせ

- 全て読み尽くしたらオブジェクトが増えるまでブロックできる
- 新しいイベントが発生するまで止まる
- やることないときはぶらぶらしない

一度に教える

- 一度にN個取り出せる
- 長い処理の間にイベントがたまってきたとかね
- 途中経過に興味がなく最新の状態に興味があるケースが多々ある
- RMI回数を最小に

○●● 通知に関すること

- オブザーバーパターンじゃない
- アプリケーション側の速度に律速しがち
- イベントが起きてもコールバックしない
- アプリケーションがひまになったらイベントを調べる

● ● ● 世界をログから編み上げる

- Koya (OODBの習作) でやってた
- Undo、タイムとラベルなど
- GitはスナップショットのCVSは変更のログでできてる
- そういえばRWikiもそうだ

○ ● ● RWiki全文検索の例

● クローラ

- 以前観測した版よりも新しいページ
を見つけてwriteする
- タグにページ名をつける

○ ● ● RWiki全文検索の例

● インデクサ

- ときどき動き出しては新しいオブジェクトをread
- 同じページ名の以前の版と一緒にSennaに更新させる



実装

- 基本的な作戦
 - インデックスとキャッシュをメモリに
 - オブジェクトはファイルに



一次記憶

- 主にインデックス
- RBTreeを二つ使う
 - 赤黒木の実装
- 時刻順のRBTree
- タグと時刻の組をキーとしたRBTree



二次記憶

- デフォルトではplainなファイル
- Marshal.dump/loadとseekの組合せ



限界

- 現在の実装での上限
- プロセスのメモリ空間
- ディスク容量
- Dropはご自宅用を想定している

●●● 巨大な問題は無視

- ふつうの人たちはみんなサイズの問題にぶちあたっているのだろうか
- BigTable族が必要な問題に私はめぐりあえない感じがする
- もっと個人的な情報に利用したい

○ ● ● in memory

- SSDはスワップが高速だってアルファブ
ロガが言ってたよ
- 新MacBook Airで限界に挑戦したいと
思うので誰か下さい
- スポンサー募集中!!!!!!!



github.com/seki/Drop/

- Githubから持ってこれます
- 別途RBTreeが必要です
- setup.rbでインストール
- lib/drop.rbを実行するとデーモンに

○ ● ● Dropの名前

- DRbを連想させてGoogleステルス性能の高い名前で決めた
- 当時、Ruby界隈でDropというプロジェクトは見つからなかった

これはまずい

- つい最近RubyDropとかいうDropboxクローンが出現したっぽい
- dRubyがあるのにDRubyを出すのが許されるんだから気にしなくていい？

○ ● ● 改名すべき？

- Dropになる前はdrip dropが候補だった
- 今日からDripにしようかなあ

API

- write
- read, read_tag
 - newer, older, head

read_tag

- read_tag(key, tag, n=1, at_least=1)
 - タグで規制されたread
 - タグを持つ要素だけを報告する

newer

- newer(key, tag=nil)
 - keyの次の要素を一つ返す
 - ブロックしない
 - read(key, 1, 0), read_tag(key, tag, 1, 0)と同じ意味

older

- `older(key, tag=nil)`
 - 過去方向のread
 - `key`の前の要素を一つ返す
 - ブロックしない
 - 過去に新しい情報は増やせないからね

head

- head(n, tag=nil)
 - 最新から過去へ
 - older(nil) n個分

naming

- 名前難しい
 - 過去から未来へ - newer
 - 現在から過去へ - older
 - 現在から最も近い過去 - older(nil)
 - 太古の次の未来 - newer(0)

○ ● ● まとめ

- すごく若いライブラリDropを紹介
- 実績のないものを話すのは緊張する
- 新MacBook Airの提供をぜひ
- 個人・企業スポンサー募集



重要

- 2005年夏
- まだ初刷買えます!
- 初刷5周年!!





なにかご質問は

