# 2017年 日本のマストドン

2017/7/27 JANOG40 Meeting 発表資料

DMM.comラボ 佐々木 健



# インターネットの価値





# JANOGの価値



# 今回も盛り上がっていきましょう

Twitterハッシュタグ

#janog

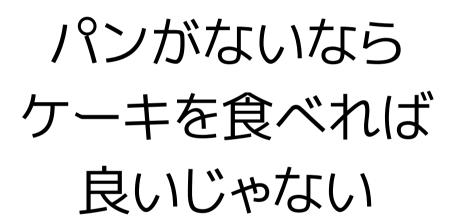
## Twitter???





## Twitter???









#### JANOG40のテーマ

「今までを振り返る、これからを知る」

## 本日の内容

- 枕
- 今までの分散SNSの振り返り
- Mastodonの特徴
- 今後の展開予想
- ディスカッション



# Mastodonブームに便乗

Mastodonがメディアに取り上げられ突然ブレイク
JANOG40プログラム募集締切日
エイヤでMastodonプログラム応募
プログラム選考会議
janogdon.net のインスタンス立ち上げ
内輪向けに細々とアナウンス
JANOGのメーリングリストでアナウンス
IPv6対応
バージョンアップ(1.3.2→1.4.1)
バージョンアップ(1.4.1→1.4.6)
バージョンアップ(1.4.6→1.4.7)
JANOG40ミーティング ← Now!!



# janogdon.net とは

https://janogdon.net/about の内容↓

#### 注意書き↓

とりあえず立ててみただけ。気まぐれで落とすかもしれない。

JANOGer っぽい人用?、と思ってるけど、JANOG運営委員とは連係してないよ。

janogdon ≠ JANOG だよ、と。

自由に遊んでくれてOK。

でも mastodon はアカウント削除機能が未実装なので覚悟してアカウント登録してね。(※バージョン1.4.2で実装されました)

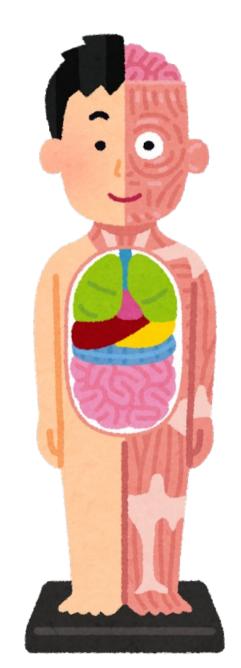
裏側のこととかは、どこかのJANOGの発表で使わせてもらうかもしれない。

サーバはさくらのVPS、非力なスペックなので、あんまりいじめないでね。



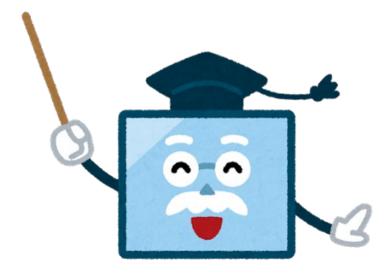
2017/7/24時点

# Mastodonの技術的概観



### 多機能なMastodon

- Twitterに似たモダンなウェブインターフェイス
- Twitterに似たAPI
- バックエンドにデータベースを持つ
- インスタンス間通信の仕組み(OStatus)



#### clocコマンド

- cloc(Count Lines of Code)
  - https://github.com/AlDanial/cloc

ソースコード中の行数を数えるツール 有効な行のみをカウント 多数の言語に対応



## clocコマンドでMastodonを調べる

% cloc mastodon-1.4.7.tar.gz

3275 text files.

3237 unique files.

1970 files ignored.

github.com/AlDanial/cloc v 1.70 T=9.37 s (139.9 files/s, 8238.7 lines/s)

Language files blank comment code

Language	files	blank	comment	code
Ruby	651	6092	1410	22216
YAML	148	196	144	16284
JavaScript	212	2375	141	12167
JSON	36	0	0	6832
SASS	23	936	5	5154
Haml	88	300	22	1466
ERB	117	374	0	627
Markdown	27	67	0	304
HTML	6	4	0	73
XML	2	0	0	11
Bourne Shell	1	4	3	7
SUM:	1311	10348	1725	65141



65,141

#### Mastodonのコード

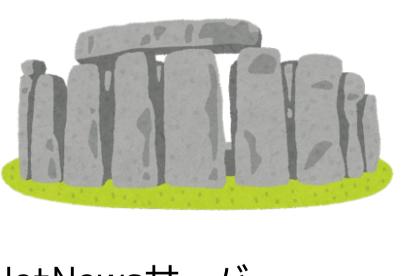
- 6万行程度のコンパクトなコード
- Ruby on Rails というモダンなウェブフレームワークで書かれている
- JavaScriptのコードも多い
- モダンな拡張言語も活用
  - SASS:スタイルシート言語
  - Haml:HTMLテンプレート言語(Railsで良く使われる)
  - ERB:HTMLへのRubyスクリプト埋めこみ技術
- 教科書的な綺麗なコード



# 他のソフトウェアと比較

- bind-9.11.0-P5
  - みんな大好きDNS
- inn-2.6.1
  - だいぶ前にお世話になったNetNewsサーバ
- IRCD-Hybrid
  - だいぶ前にお世話になったIRCサーバ







#### bind-9.11.0-P5

% cloc bind-9.11.0-P5
4722 text files.
4434 unique files.
2568 files ignored.

github.com/AlDanial/cloc v 1.70 T=8.49 s (255.2 files/s, 102860.2 lines/s)

code	comment	blank	files	Language
363251	56380	66345	768	 :
133091	20143	19049	444	Bourne Shell
44516	795	548	116	HTML
31248	42156	10477	609	C/C++ Header
21412	138	1258	13	KML
17860	818	1989	32	n4
17191	3129	3487	74	C++
8472	1771	1571	61	Perl
1588	138	146	7	KSLT
842	149	174	2	Tcl/Tk
513	52	20	9	Windows Module Definition
366	63	64	5	Python
319	70	112	13	make
249	47	41	2	Windows Resource File
240	74	31	1	diff
236	145	69	5	awk
140	28	31	1	css
64	0	22	1	Markdown
49	170	49	1	Protocol Buffers
29	1	8	1	Windows Message File
20	33	8	1	OTO
8	6	3	1	ГеX
641704			2167	 SUM:



641,704

Mastodonの 10倍

#### inn-2.6.1

% cloc inn-2.6.1

894 text files.

885 unique files.

406 files ignored.

github.com/AlDanial/cloc v 1.70 T=1.55 s (314.1 files/s, 138891.9 lines/s)

Language		blank	comment	code
С	262	16751	20114	94598
Bourne Shell	28	6071	6476	35541
m4	37	1091	986	10219
C/C++ Header	94	1390	3311	4503
make	24	579	230	3556
Perl	22	343	1355	3107
Visual Basic	5	341	0	2588
yacc	1	212	19	842
Python	10	137	445	343
lex	1	46	12	197
CSS	1	3	7	79
XSLT	1	8	42	78
Korn Shell	1	26	81	18
Oracle Forms	1	3	0	15
SUM:	488	27001	33078	155684



155,684

Mastodonの 2.4倍

# IRCD-Hybrid

- % cloc hopm-1.1.4
  - 101 text files.
  - 94 unique files.
  - 14 files ignored.

github.com/AlDanial/cloc v 1.70 T=0.26 s (331.9 files/s, 230088.4 lines/s)

Language	files	blank	comment	code
Bourne Shell	10	4133	4547	24884
m4	16	1009	131	9678
С	23	1903	2632	8746
C/C++ Header	30	195	443	814
yacc	1	147	28	552
lex	1	60	26	285
make	5	6	0	70
YAML	1	5	1	19
SUM:	87	7458	7808	45048



45,048

## 比較結果

ソフトウェア	総コード数	主な言語
Mastodon	65,141	Ruby, Java Script
bind	641,704	C
inn	155,684	C
IRCD-Hybrid	45,048	Shell,C

Mastodonは高機能なのにコンパクト モダンな開発フレームワークの威力

### Mastodonが利用するソフトウェア

• Ruby on Rails →ウェブ開発フレームワーク





• Redis → KVS

Node.js →サーバサイドJavaScriptエンジン

• Docker →コンテナ管理フレームワーク

## 運用上必要になるソフトウェア

- Linux OS基盤
- リバースプロキシ(Nginx等)
- メールサーバ(Postfix等)
- 監視ツール



## 技術的概観まとめ

- 多機能を少ないコードで実現
- 新しめの技術を利用
- 複数の外部ソフトウェアを活用
- コンテナ技術を活用したパッケージング



# Mastodonの技術的課題



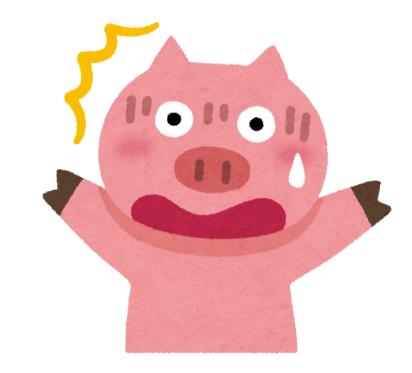
## 運用が大変

- インストールが簡単ではない
- アップデートも簡単ではない
- 理解すべき技術要素が多い
- トラブルシュートが難しい



## 商用サービスとしては機能不足?

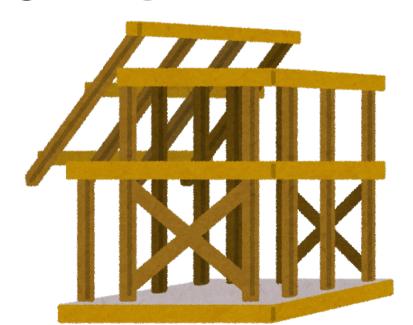
- アカウント削除機能がない (※バージョン1.4.2で実装された)
- 管理機能が弱い (※バージョンアップの度に強化されている)
- 外部認証が実装されていない (※開発者が実装予定と言っている)



- 高負荷時にボトルネックポイントがある
  - (※バージョンアップ時に改良されている)

# プロトコル(OStatus)が発展途上

- まとまった仕様書がない
- インスタンスをまたいだメッセージ削除ができない
- URIをキーに用いてしまっている
- 安全なインターネットを前提にしている



### 耳にしたことがあるクレーム

- PosgreSQLじゃなくてMySQLを使って欲しい
- React.jsを使ってるのはダサい
- UIが絶望的に良くない
- APIとUIは分離させるべき

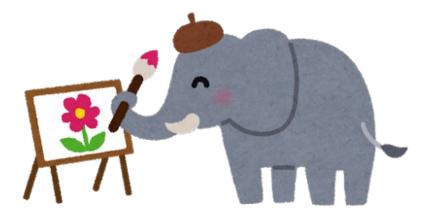


# Mastodonの今後の展望



## 継続的なソフトウェア開発

- GitHub上で開発が続けられている
- 開発ペースは速い
- 開発者のモチベーションも高い
- コントリビューションも多い
- 不足機能は継続的に追加されている



## プロトコルの進化

- OStatus → ActivePub
- Pubsubhubbub → WebSub
- OpenID → WebFinger
- Atom → Activity Streams



## 別実装の出現

- Mastodonの拡張
  - Pawoo
  - friend.nico
- OStatus通信可能な別実装
  - stat.ink



# Mastodonをいじってみた感想



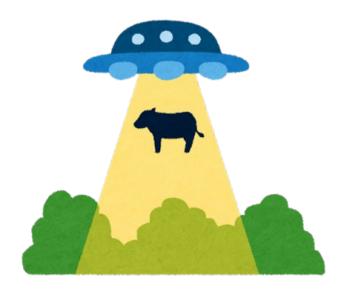
## なつかしい気分になった

- インスタンス間でメッセージをやりとりするのは NetNewsやIRCを彷彿とさせる
- 昔のノウハウがそのまま生きる
  - ストレージを圧迫するのは画像
  - 繋がる相手を選ぶことでストレージ圧迫を減らせる
- 小さいコミュニティの面白さ



## 新しい技術は楽しい

- 少ないコード量で超高機能なサービスを実現
- 強力なモダンフレームワーク
- 複数のソフトウェアを統合的に管理できるDocker



### こうなると嬉しい

- 簡単にインストール可能
  - apt install mastodon
  - kubectl run mastodon --option...
- 運用が楽な別実装の出現
- 既存のSNSのOStatus(ActivePub)サポート
  - mixiの復活、Twitterという巨大インスタンスの出現
- クラウドサービスの出現(SaaS、BaaS)
  - 一般人はサーバ管理はしたくない



## 最後に言いたいこと

- 人と人とを繋ぐのはインターネットの大きな価値、 それを実現するツールはなくならない
- 新しいものは面白い
- 古い知識も意外と役に立つ
- Mastodonがなくなっても知見は後で役に立つ
- 一緒にMastodonで遊びましょう



# インターネットの価値



# おしまい

Special thanks to いらすとや

