

複数のGemini CLIが同時開発する狂気

Jujutsuが実現するAIエージェント協調の新世界

グンタ・ブルナー Günther Brunner

株式会社サイバーエージェント

⌚ 2025年7月22日 • 🗺️ Gemini CLI Meetup Japan #1

自己紹介

 開発生産性 & パフォーマンス専門

 2012年サイバーエージェント入社

 OSS OpenSTF創設者
GitHub 13k+ スター

 AI Code Agents祭り主催
2万人以上視聴

 X でフォロー
@gunta85



問題提起



現在の限界

- 1つのGemini CLIで順番に作業
 - ☒ 一つずつしか処理できない
- ファイル競合の恐怖
 - ☒ マージコンフリクトで作業停止

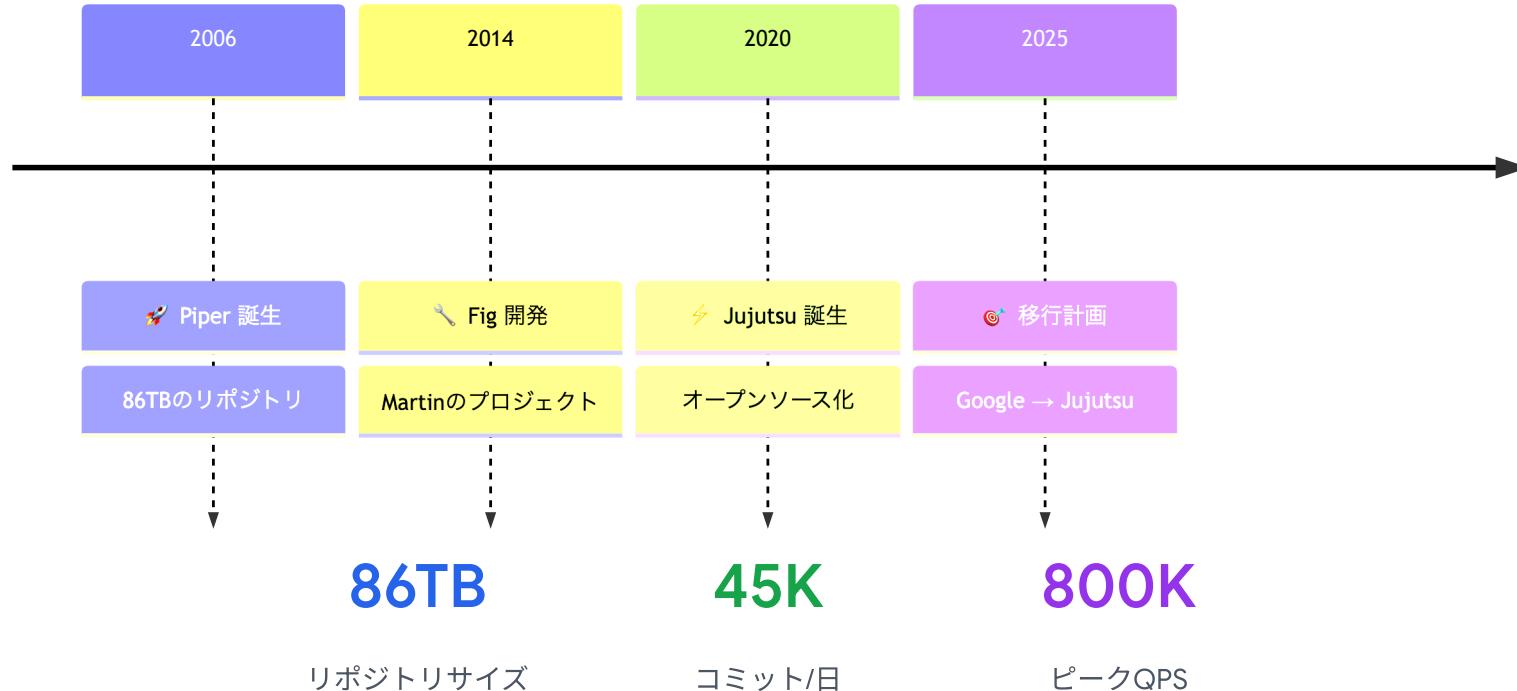


理想の世界

- 複数のGeminiが同時開発
 - ⌚ 真の並列開発
- コンフリクトフリー
 - ⚡ シームレスなマージ

欠けていたピース：Googleの物語

Googleが大規模なバージョン管理をどう解決したか



Jujutsu: Googleの経験から生まれた

Googleの経験から生まれた次世代VCS



Martin von Zweigbergk

Jujutsuの創作者

- ✓ Git rebase 共同作者 (2008)
- ✓ Google Fig VCS アーキテクト
- ✓ 20年以上のVCS経験
- ✓ Googleがフルタイム開発をスポンサー



Jujutsuが重要な理由

- ❤ コンフリクトをファーストクラスに - コンフリクトのままコミット可能
- ↶ 完全なundo - あらゆる操作が取り消し可能
- ↳ 真のワークスペース - 並列開発
- 🚀 ステージングエリアなし - AI向き



Scott Chacon @chacon

GitHub 共同創業者

「CLIが好きなら、Jujutsuはすごく面白い。Gitがバックエンドだしね。」



Mitchell Hashimoto @mitchellh

HashiCorp 創業者

「このツイートから約2ヶ月、ずっとjjだけを使ってる。もっと長く書きたいけど、一言で言うと「jjは最高、もう元には戻れない」。唯一の例外は`git bisect`の時だけgitを使うこと。それだけ。」

2024年10月15日

ゲームチェンジャー：コンフリクト共有

従来のGitモデル

- ① コンフリクトは作業を止める

解決するまで進めない

- ② コミット前に解決必須

コンフリクト状態は保存不可

- ③ 各ワークツリーは独立

共有された理解なし

Jujutsu革命

- ① コンフリクトはデータ

ファーストクラスオブジェクトとして保存

- ② コンフリクトのままコミット

すぐに作業を継続

- ③ ワークスペース間で状態共有

全エージェントが同じコンフリクトを見る

複数のAIエージェントが同じコンフリクトのあるコミットで同時に作業可能

並列開発のサンプル

3つのGemini CLIが同時に動作

A バックエンド

backend-workspace

- REST API開発
- データベース連携

B フロントエンド

frontend-workspace

- Reactコンポーネント
- UI/UX実装

C テスト

test-workspace

- ユニットテスト
- 統合テスト

```
# 3つのワークスペースを作成
jj workspace add backend
jj workspace add frontend
jj workspace add test

# 各ワークスペースでGemini CLIを起動
cd backend && gemini -p "REST APIを構築" &
cd ../frontend && gemini -p "React UIを作成" &
cd ../test && gemini -p "テストを作成" &
```

コンフリクト解決の実例

✖️ コンフリクトシナリオ

3つのAIエージェントがserver.jsを異なる方法で修正:

認証チーム

app.use(auth)

データベースチーム

app.use(database)

ロギングチーム

app.use(logger)

❖ Git: 順次解決

1 エージェントAがコンフリクトに遭遇 → ブロック

1 全エージェントがコンフリクトを確認

2 エージェントBがAを待つ → ブロック

2 それぞれが自分の部分を作業

☒ 合計: 3倍の解決時間

⚡ 合計: 1倍の解決時間

各ワークスペースのエージェントは:

jj edit @ # 同じコンフリクトのあるコミットで作業可能 !

実例：機能開発

Gitワークフロー

```
# 機能ブランチを作成  
git checkout -b feature-payment  
  
# バックエンドで作業  
git add backend/*  
git commit -m "支払いAPIを追加"  
  
# コンテキストを切り替え  
git stash  
  
git checkout -b feature-payment-ui  
  
# マージコンフリクト！  
git merge feature-payment  
  
# CONFLICT (content): マージコンフリクト
```



Jujutsuワークフロー

```
# バックエンド用ワークスペースを作成  
jj workspace add backend  
  
# フロントエンド用ワークスペースを作成  
jj workspace add frontend  
  
# 同時に作業  
cd backend && jj new  
cd ../frontend && jj new  
  
# いつでもマージ  
jj new @backend @frontend  
  
# コンフリクトは保存され、作業は継続！
```

結果

並列開発と協調が出会うとき



5分で完全なアプリケーション

Gemini CLI + Jujutsu: 完璧なマッチ

AI向けの機能

- ✓ ステージングの混乱なし

AIが git add を忘れない

- ✓ 自動スナップショット

全変更が自動保存

- ✓ コンフリクト耐性

コンフリクトがあっても作業継続

生産性の向上

セットアップ時間

-90%

コンフリクト解決

-95%

全体的な速度

+1000%

高度な活用：マルチエージェント協調

💻 エージェントネットワーク構成

- コーディネーターエージェント タスク分配
- バックエンドエージェント (x3) API開発
- フロントエンドエージェント (x3) UI開発
- テストエージェント (x2) 品質保証

↔ オーケストレーションサンプル

```
# コーディネーターがタスクを作成
jj workspace add coordinator
jj new -m "プロジェクト構造"

# 専門エージェントを生成
for i in {1..3}; do
    jj workspace add backend-$i
    gemini -p "サービス$iを構築" &
done

# 全作業を自動マージ
jj new @backend-* @frontend-*
```

スケールでの未来

100のGemini CLIが調和して動作する世界を想像してください

100+ AIエージェント



マイクロサービス

各サービスに1つのAI

専門知識の活用



多言語対応

言語専門家

Python、Go、Rust、JS



専門役割

セキュリティ、性能、UX

ドメインエキスパート

今すぐ始めよう

```
# Jujutsuをインストール  
brew install jj  
# リポジトリを初期化  
jj init --git  
# ワークスペースを作成  
jj workspace add feature-a  
jj workspace add feature-b  
# Gemini CLIを起動  
gemini -p "素晴らしい機能を構築"
```

ドキュメント



Jujutsu公式ドキュメント

github.com/jj-vcs/jj

チュートリアル



インタラクティブ学習

jj-vcs.github.io

ありがとうございました

グンタ・ブルナー

株式会社サイバーエージェント

X @gunta85

一緒にAIの未来を創りましょう