

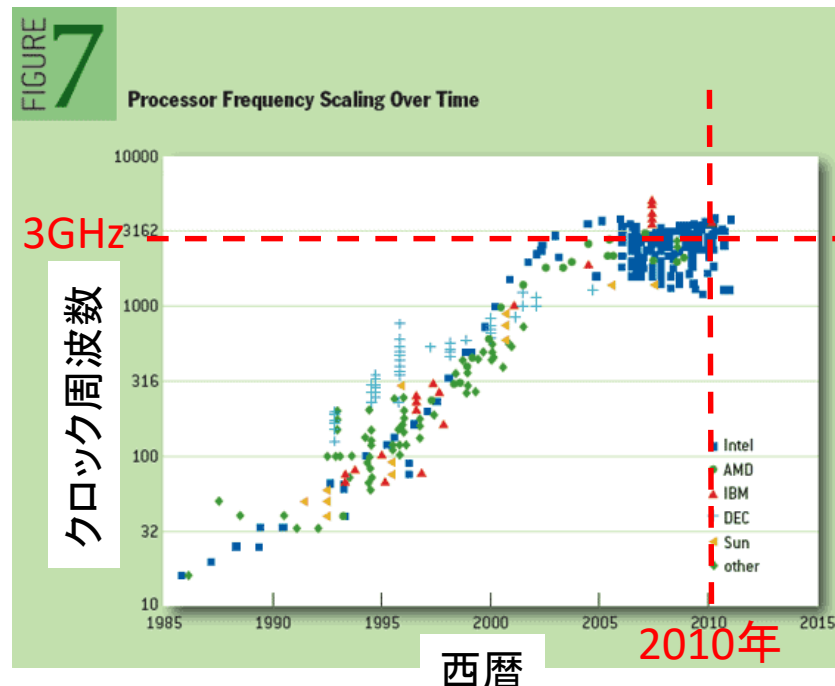
ALMA 用新高速データ解析 ライフライン Sakura の開発

中里 剛、杉本香菜子、川崎 渉、
川上申之介、中村光志、小杉城治
(国立天文台)

Sakuraライブラリ: 概要

- データ解析のための基礎的な機能を提供する
- マルチコアCPUの性能を最大限に引き出す
 - ベクトル化の徹底活用
 - スレッドセーフな実装
 - CPU世代に応じた最適化
- 既存の電波望遠鏡データ解析ソフトウェアに対して**10~20倍処理を高速化**

CPUのクロック周波数は頭打ち
マルチコアCPUの時代



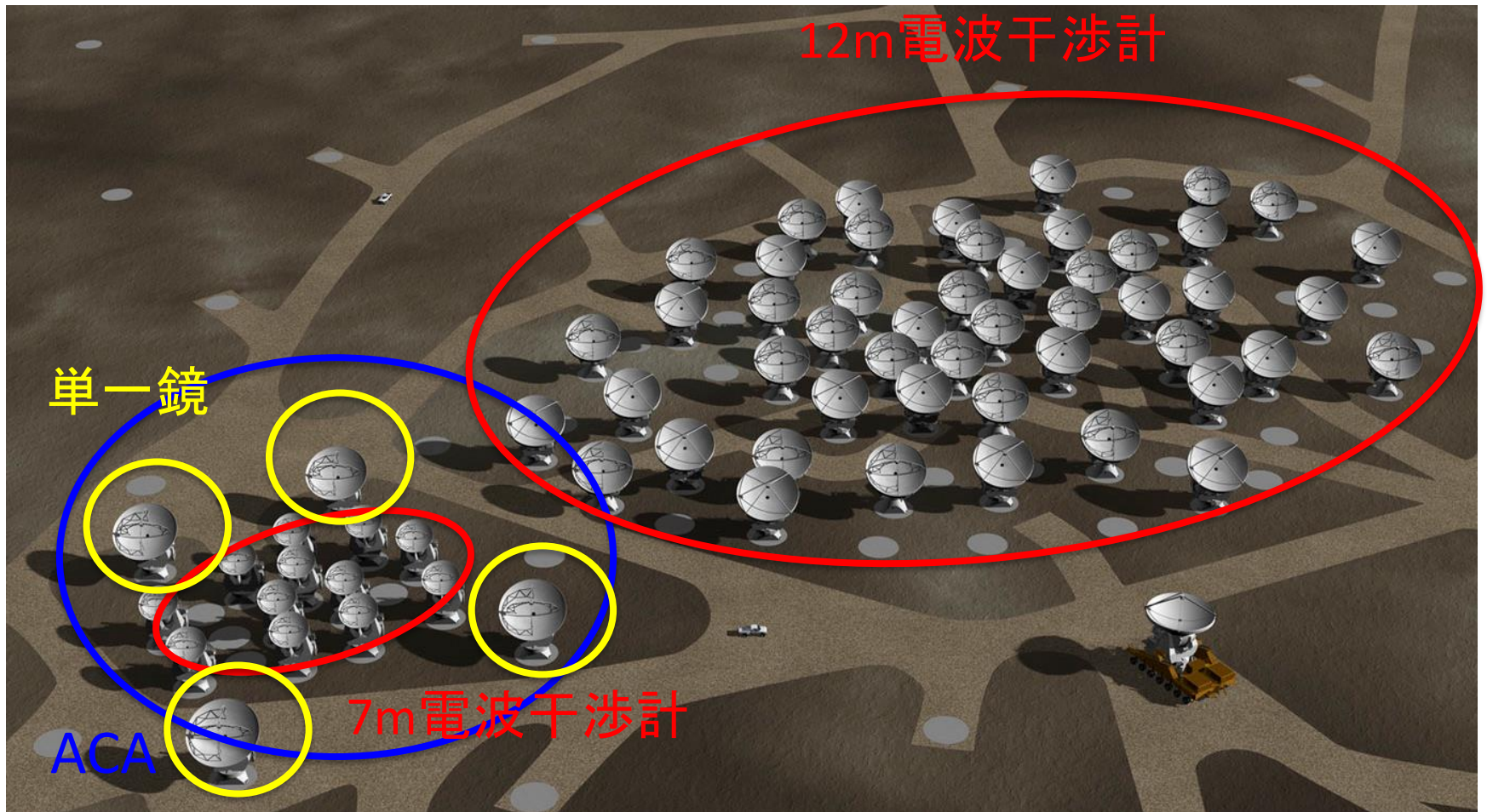
<http://gigazine.net/news/20130725-40-year-cpu-history/>

内容

- ALMA望遠鏡とは
- 単一電波望遠鏡の観測について
 - 観測データと観測手法
 - 解析手法
- Sakuraライブラリ
 - 3つの特長
- 性能評価
 - 性能評価用アプリケーション
 - 性能測定結果
 - 考察
- まとめと今後の展望

ALMA望遠鏡とは

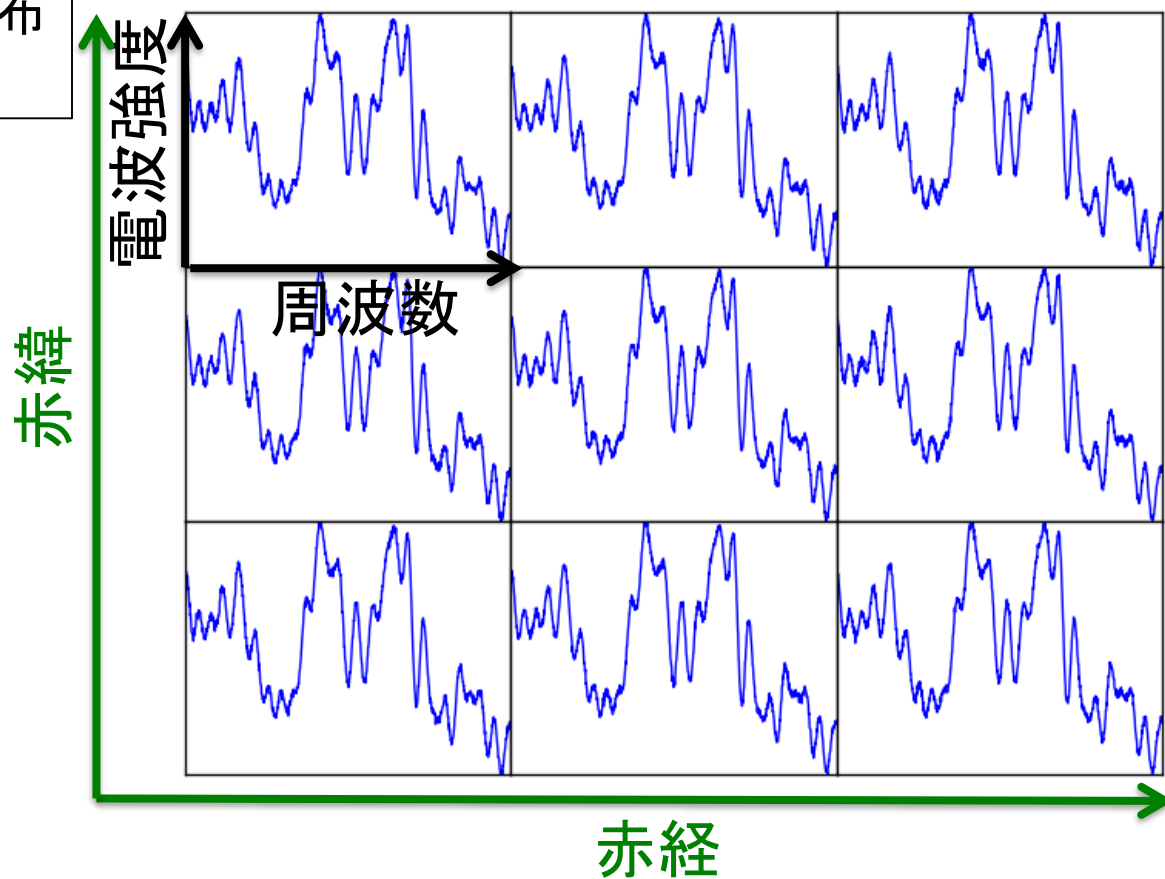
(Atacama Large Millimeter/Submillimeter Array)



ALMA(ESO/NAOJ/NRAO)

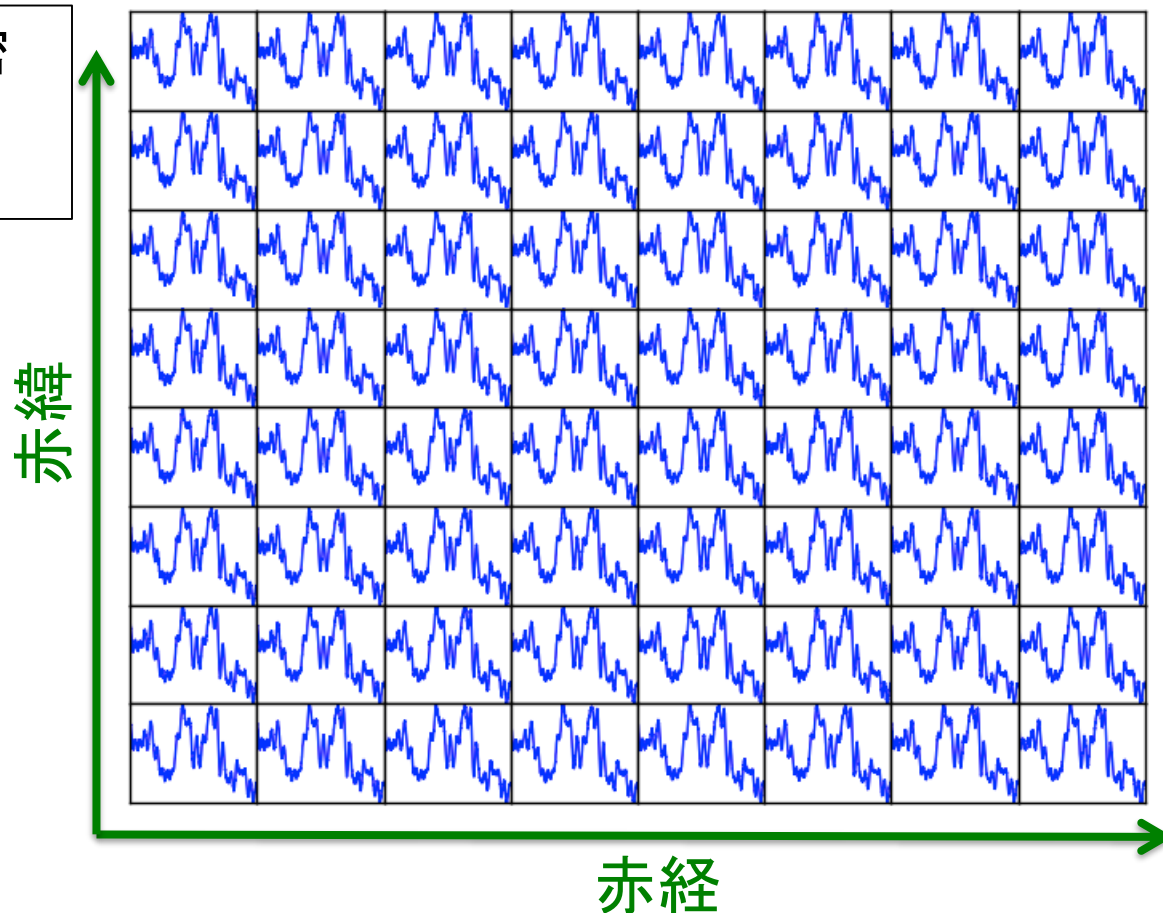
単一電波望遠鏡の観測データ

- 天球面上の電波強度の分布
- スペクトル(分光)観測



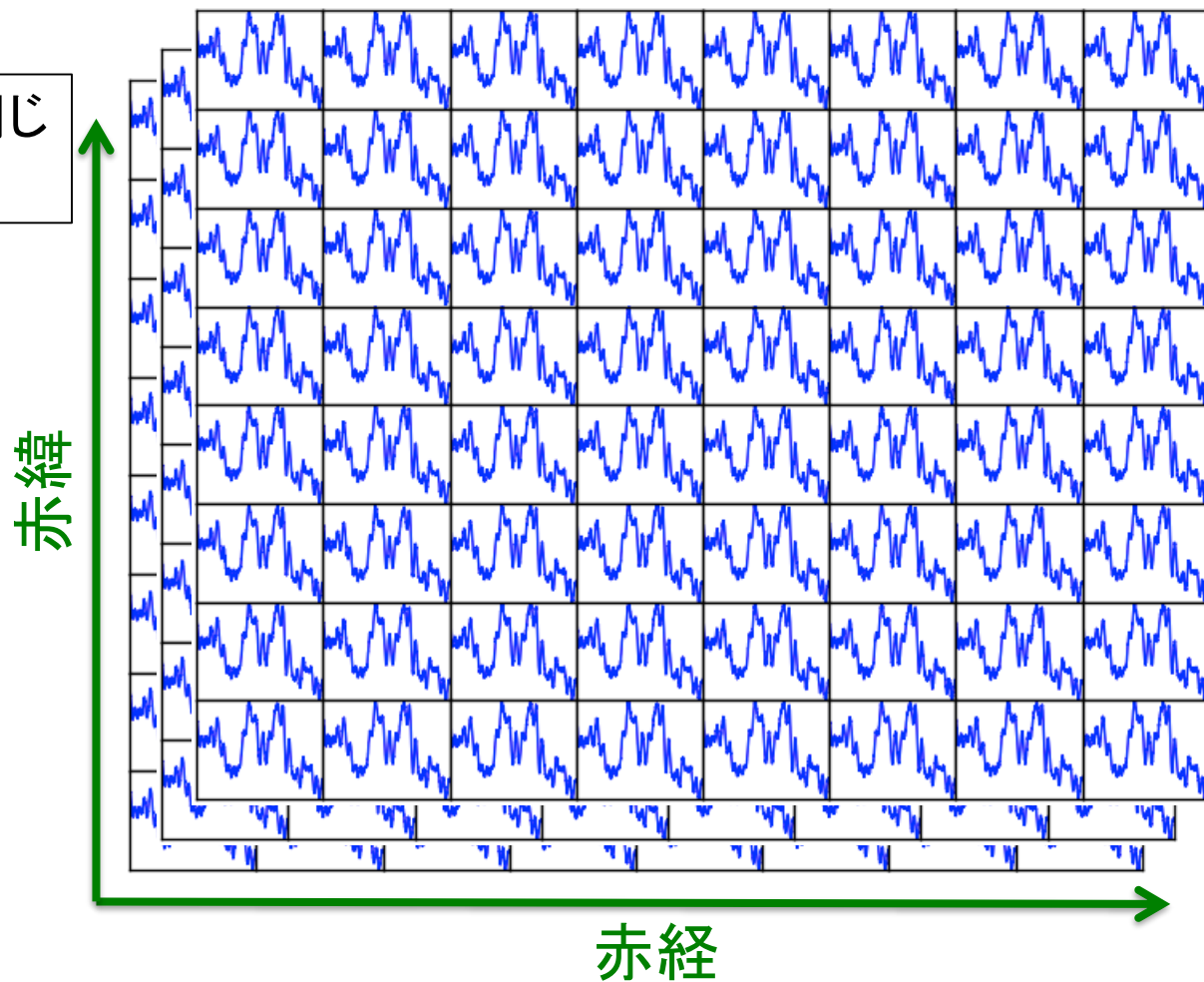
単一電波望遠鏡の観測データ

- 空間サンプリングは十分密でなければならない (cf. サンプルング定理)



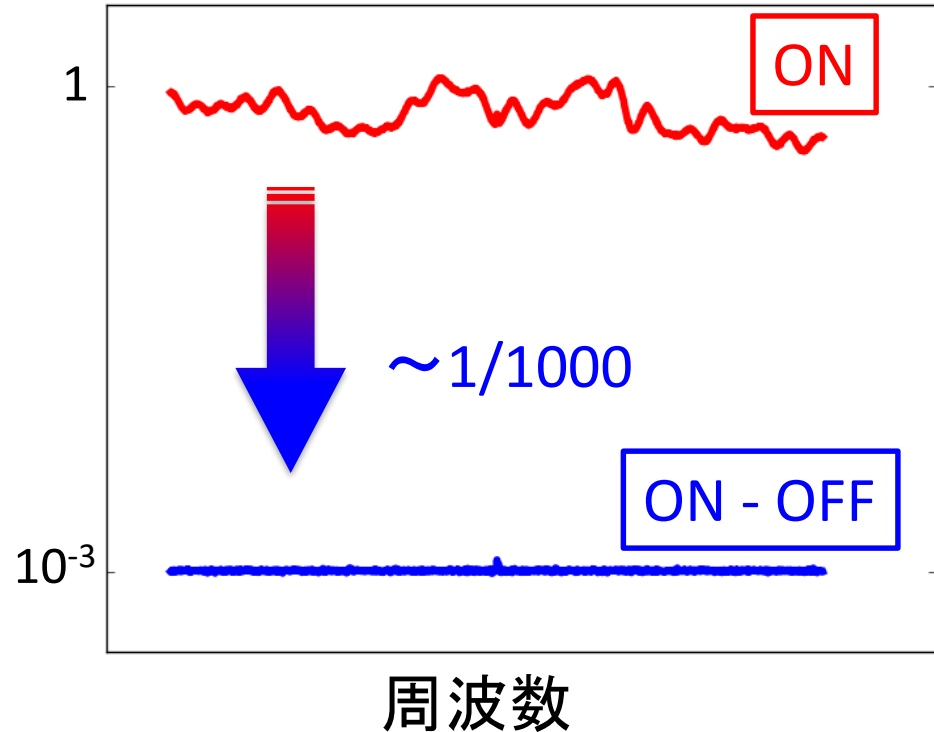
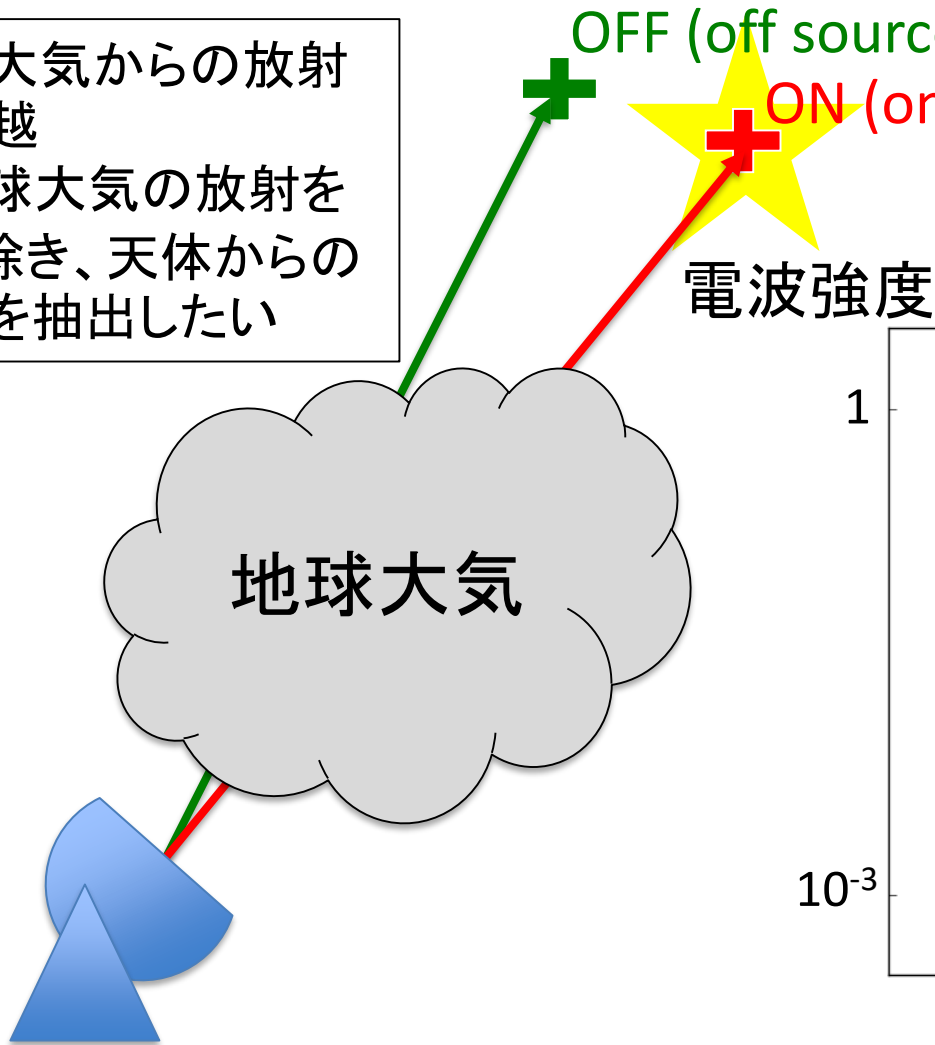
単一電波望遠鏡の観測データ

- S/N比が十分でなければ同じ場所を何度も観測する



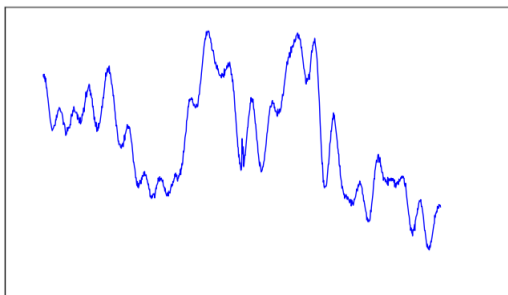
単一電波望遠鏡の観測手法 (ポジションスイッチ法)

地球大気からの放射が卓越
→地球大気の放射を取り除き、天体からの成分を抽出したい



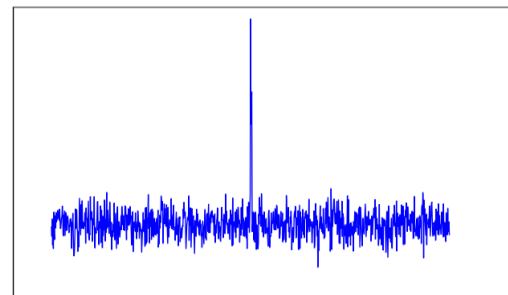
単一電波望遠鏡データの解析

電波強度



周波数

電波強度



周波数

データ校正

- ON – OFF
- 電波強度
スケーリング

フラグging

- 性質の悪い
データの無効化

ベースライン 除去

- データ校正の残差
- 連続波成分

スムージング

- 周波数方向に
データを滑らかに
する

Sakuraライブラリ

- 基礎的な解析機能を提供する汎用ライブラリ
 - 現在は単一電波望遠鏡データの解析にターゲットを絞って開発
- 高速な処理
 - 使えるリソースは最大限利用する
 - 「マルチスレッド × ベクトル化」の相乗効果
- 実行環境を選ばない
 - ノートPCからクラスタ計算機まで
 - C API

Sakuraライブラリ: 特長①

ベクトル化を徹底的に活用

通常の命令とSIMDの比較

※ SIMD: Single Instruction Multiple Data

通常の命令

$$\begin{array}{l} A_0 + B_0 = C_0 \\ A_1 + B_1 = C_1 \\ A_2 + B_2 = C_2 \\ A_3 + B_3 = C_3 \end{array}$$

SIMDの命令

$$\begin{array}{l} A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} + \begin{array}{l} B_0 \\ B_1 \\ B_2 \\ B_3 \end{array} = \begin{array}{l} C_0 \\ C_1 \\ C_2 \\ C_3 \end{array}$$

<http://ascii.jp/elem/000/000/172/172886/> より引用

- コンパイラによるベクトル化
 - ベクトル化されやすいようにコーディング
- 必要に応じてIntrinsicsを使う

データ較正

$$\begin{array}{l} A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} + \begin{array}{l} B_0 \\ B_1 \\ B_2 \\ B_3 \end{array} = \begin{array}{l} C_0 \\ C_1 \\ C_2 \\ C_3 \end{array}$$

フラグging

$$\begin{array}{l} A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} + \begin{array}{l} B_0 \\ B_1 \\ B_2 \\ B_3 \end{array} = \begin{array}{l} C_0 \\ C_1 \\ C_2 \\ C_3 \end{array}$$

ベースライン

$$\begin{array}{l} A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} + \begin{array}{l} B_0 \\ B_1 \\ B_2 \\ B_3 \end{array} = \begin{array}{l} C_0 \\ C_1 \\ C_2 \\ C_3 \end{array}$$

スムージング

$$\begin{array}{l} A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} + \begin{array}{l} B_0 \\ B_1 \\ B_2 \\ B_3 \end{array} = \begin{array}{l} C_0 \\ C_1 \\ C_2 \\ C_3 \end{array}$$

Sakuraライブラリ: 特長②

スレッドセーフな実装

- アプリケーションレベルでのマルチスレッド処理をサポート
 - Sakura自身はマルチスレッドでは動作しない
 - Sakuraを基盤とするアプリケーションのマルチスレッド処理を妨げない

データ較正

$$\begin{array}{|c|} \hline A_0 \\ \hline A_1 \\ \hline A_2 \\ \hline A_3 \\ \hline \end{array} + \begin{array}{|c|} \hline B_0 \\ \hline B_1 \\ \hline B_2 \\ \hline B_3 \\ \hline \end{array} = \begin{array}{|c|} \hline C_0 \\ \hline C_1 \\ \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

フラッキング

$$\begin{array}{|c|} \hline A_0 \\ \hline A_1 \\ \hline A_2 \\ \hline A_3 \\ \hline \end{array} + \begin{array}{|c|} \hline B_0 \\ \hline B_1 \\ \hline B_2 \\ \hline B_3 \\ \hline \end{array} = \begin{array}{|c|} \hline C_0 \\ \hline C_1 \\ \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

ベースライン

$$\begin{array}{|c|} \hline A_0 \\ \hline A_1 \\ \hline A_2 \\ \hline A_3 \\ \hline \end{array} + \begin{array}{|c|} \hline B_0 \\ \hline B_1 \\ \hline B_2 \\ \hline B_3 \\ \hline \end{array} = \begin{array}{|c|} \hline C_0 \\ \hline C_1 \\ \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

スムージング

$$\begin{array}{|c|} \hline A_0 \\ \hline A_1 \\ \hline A_2 \\ \hline A_3 \\ \hline \end{array} + \begin{array}{|c|} \hline B_0 \\ \hline B_1 \\ \hline B_2 \\ \hline B_3 \\ \hline \end{array} = \begin{array}{|c|} \hline C_0 \\ \hline C_1 \\ \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

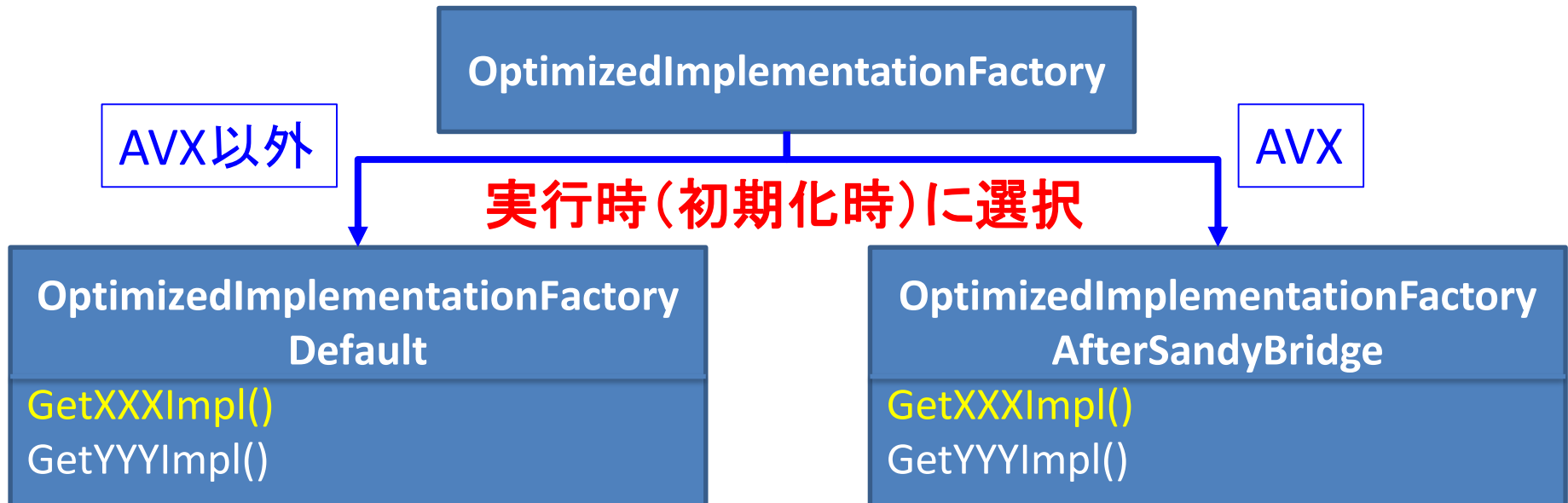
$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline A_2 \\ \hline A_3 \\ \hline \end{array} \begin{array}{|c|} \hline B_2 \\ \hline B_3 \\ \hline \end{array} \begin{array}{|c|} \hline C_2 \\ \hline C_3 \\ \hline \end{array}$$

Sakuraライブラリ: 特長③

CPU世代に応じた最適化

- CPU世代毎に最適化されたオブジェクトコードを持ち、実行時（SAKURAの初期化時）に適切なコードが選択される
- 現在はMMX/SSE (128ビット幅)とIntel AVX(256ビット幅)をサポート

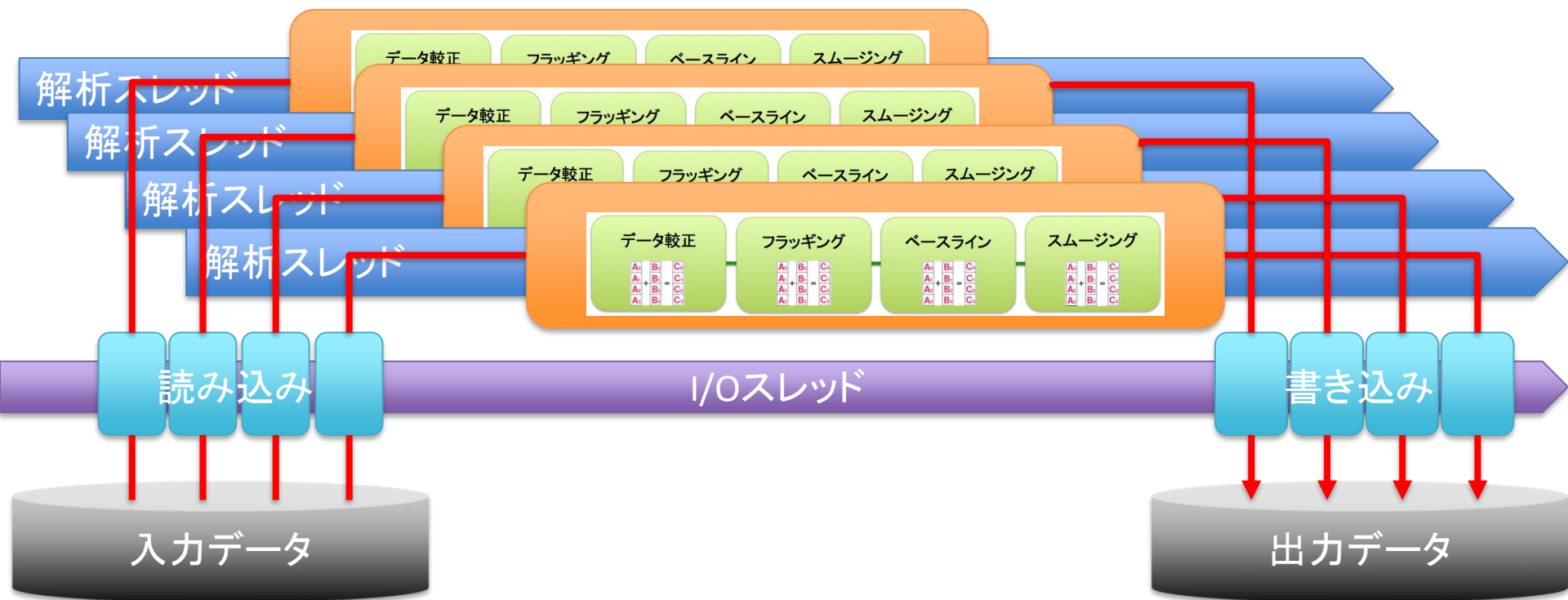


性能評価

- SAKURAベースの性能評価用アプリケーションを作成
- CASAで同内容の解析スクリプトを作成して比較
 - Common Astronomy Software Applications
 - ALMAのデータ解析ソフトウェア
 - シリアル処理主流の典型的アプリケーションとして
- 測定環境
 - CPU: Intel® Xeon® CPU E5-1650 @ 3.20GHz
6 cores x 1CPU (12 Threads)
 - Memory: 64GB
 - Disk: 3TB (512GB SSD × 6, RAID0)

性能評価用アプリケーション

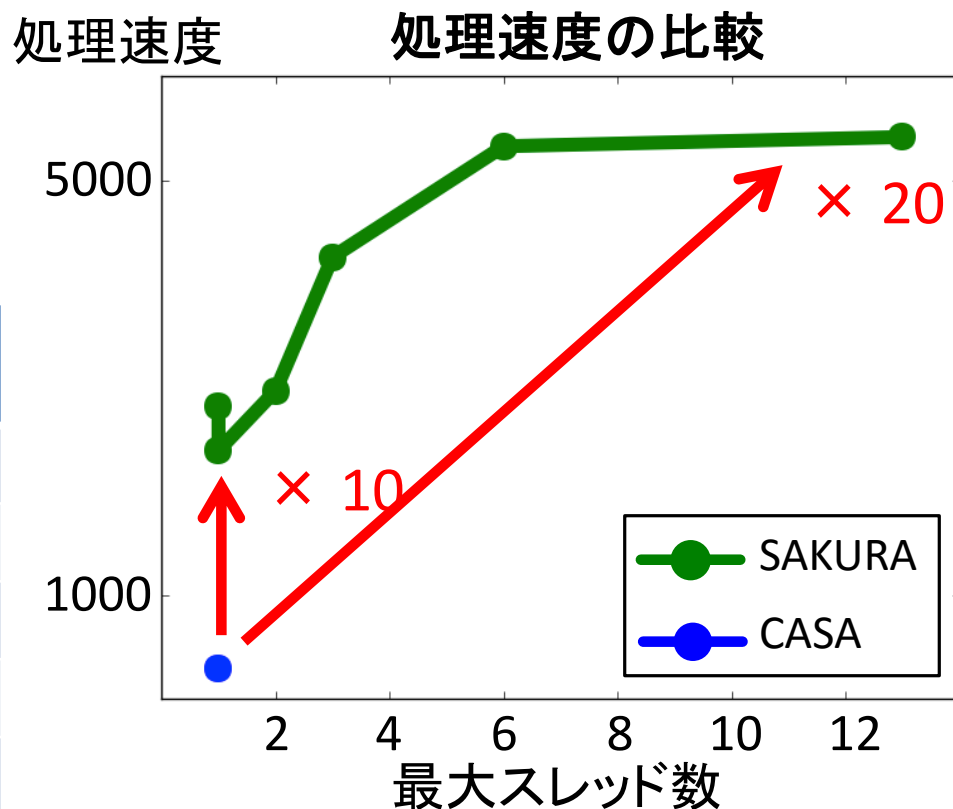
- 解析スレッドで解析処理を並列に行う
 - 処理単位はスペクトル1本に対する一連の解析処理
 - 処理結果の書き込みをI/Oスレッドに依頼したら次の解析処理に移る
- 入出力はI/Oスレッドでシリアルに行う



測定結果

- スペクトル数4万本
 - 1本は長さ3840のfloat配列
- 処理速度 $\equiv \frac{\text{スペクトル数}}{\text{処理時間(秒)}}$

アプリケーション	最大スレッド数	処理時間(秒)	処理速度
SAKURA	シリアル	14.3	2826.7
	1	16.8	2395.2
	2	13.6	2970.8
	3	9.5	4258.7
	6	7.6	5326.4
	13	7.4	5415.3
CASA	シリアル	139.7	288.7



※処理時間にはI/Oも含む
 ※最大スレッド数はI/Oスレッドを含まない

考察

- SAKURAはCASAより10~20倍速い
- SAKURA(シリアル)はCASAのおよそ10倍速い

- ベクトル化の効果は最大4倍
(256ビット = double × 4)
- 余計なデータコピーの低減も効いている

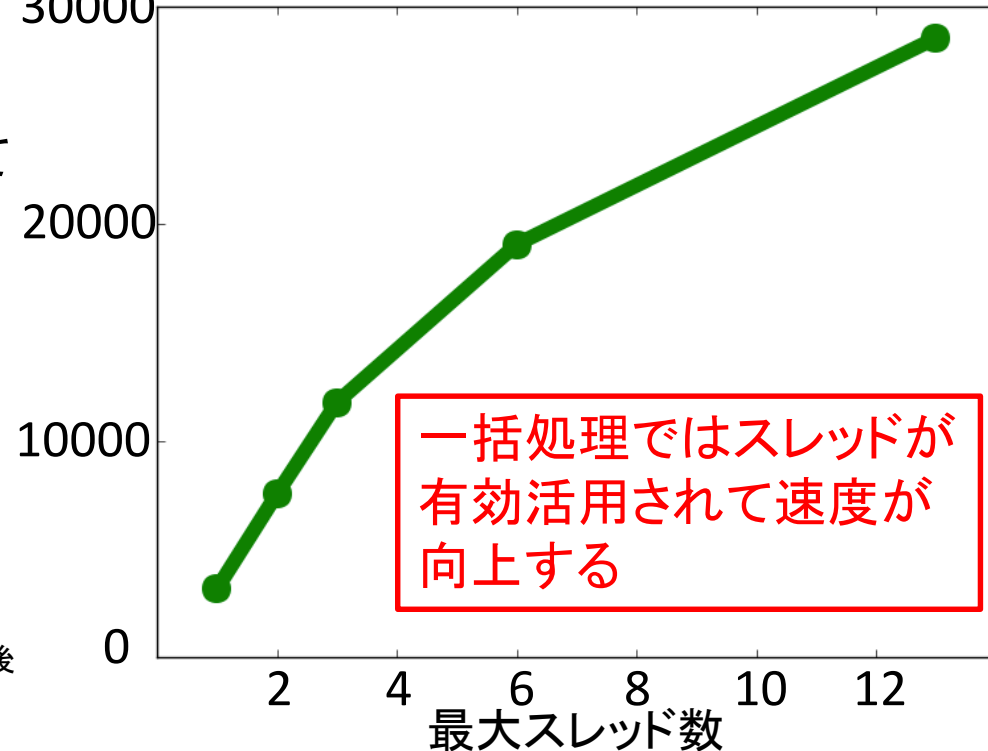
- マルチスレッド化がそれほど効いていないように見える

- スレッド数を増やしても実質6並列程度
- I/Oネックである
- 「一括処理」モードでは速度が向上する(右図)

- 処理開始時にデータを全て読み込み、処理終了後にまとめて書き出す

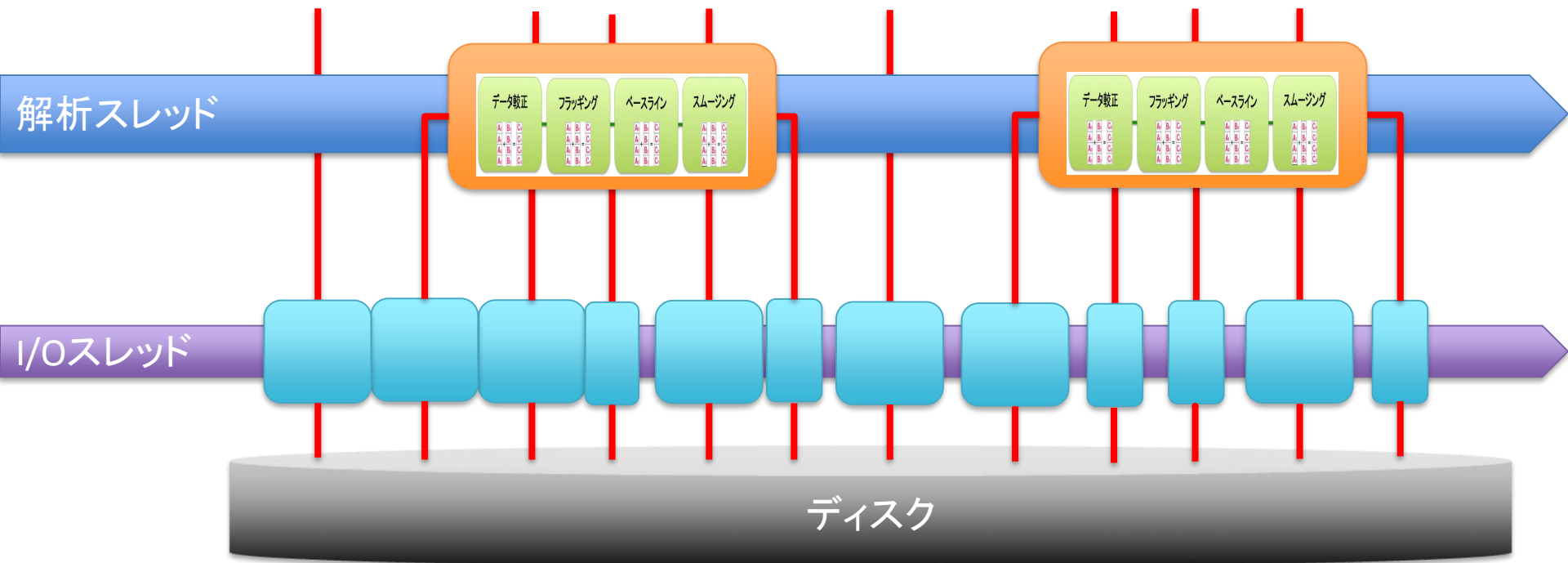
処理速度
(I/O除く)
30000

処理速度の推移(一括処理)



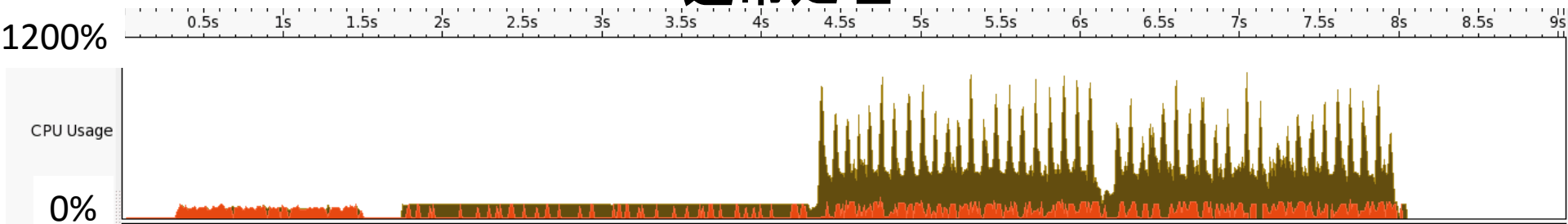
処理はI/Oネックである

- 解析スレッドの処理にI/Oの速度が追いついていない
- 解析スレッドは次のデータを受け取るまで遊んでいる

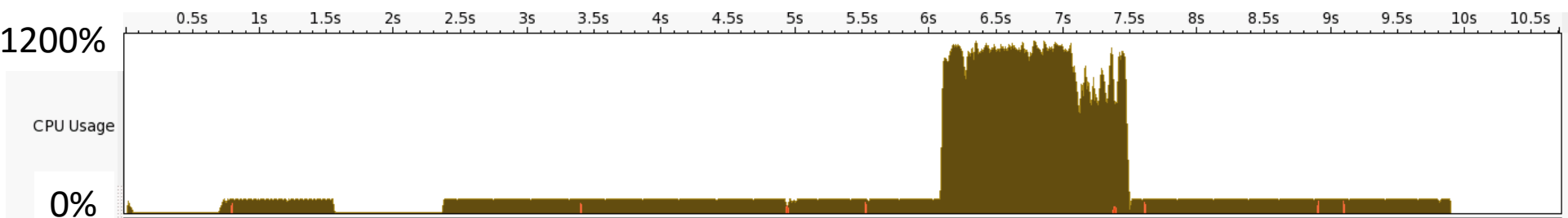


CPU使用率

通常処理



一括処理



まとめと今後の展望

- まとめ

- 高速データ解析ライブラリSakuraを開発した
 - ベクトル化の徹底活用
 - スレッドセーフな実装
 - CPU世代に応じた最適化
- ALMA単一鏡データの解析は、Sakuraを使えば**10～20倍の高速化**が見込める

- 今後の展望

- ALMAのデータ解析を高速化
- 観測データのクイックルックへの応用
 - おおざっぱに処理したデータを準リアルタイムで表示
- 機能のさらなる充実による汎用化