「JSS3 を用いた科学衛星データ処理とデータ解析」

中平 聡志(JAXA)
海老沢 研(JAXA) 栗原 明稀(東大・院) 高木 亮治(JAXA)

0. 背景と本発表のスコープ

科学衛星の観測データ量は、装置や通信の進化および観測期間の長期化により増加を続けている。その状況下で、、、

- 較正パラメータの更新に伴う高次データの一括再処理
- 長期観測データ(10年など)から特定ターゲットに対する観測結果を取り出す科学解析処理

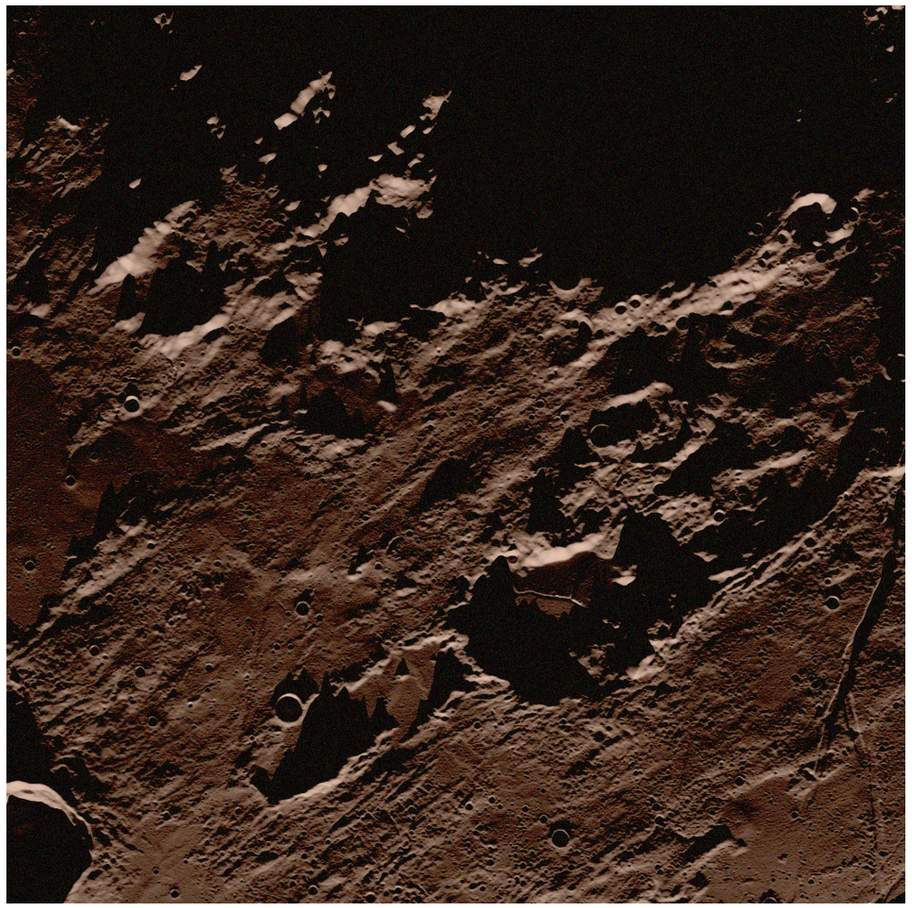
等を行うために、研究室/部署単位で所有するサーバーは数ヶ月単位の時間を要する場合がある

→その実施のためJAXAのスパコン(**JSS3**) でどれだけ有用か試した

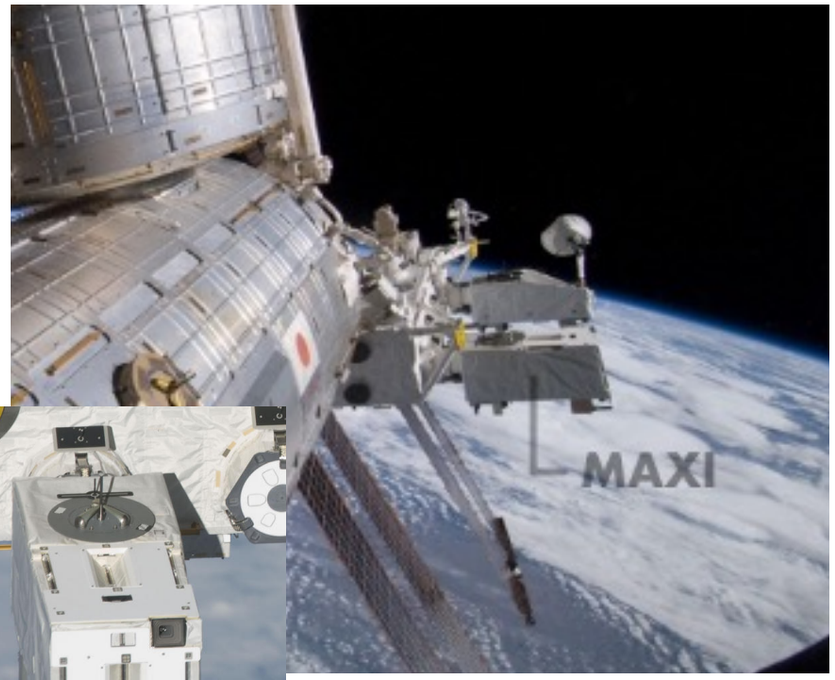
0. 背景と本発表のスコープ

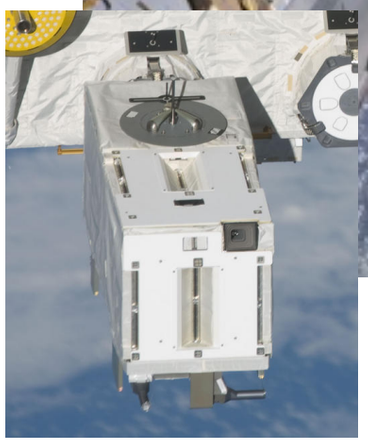
実施した内容

1. MAXIの約12年間の公開アーカイブデータの再作成処理
2. Swift衛星のBAT検出器の長期観測データ(2004.11-現在)から、着目した天体の硬X線強度データを抽出
   ・かぐやDEMデータの変換処理(午後の小林・梶浦発表と関連)
   →時間の都合で省略
3. 考察とまとめ



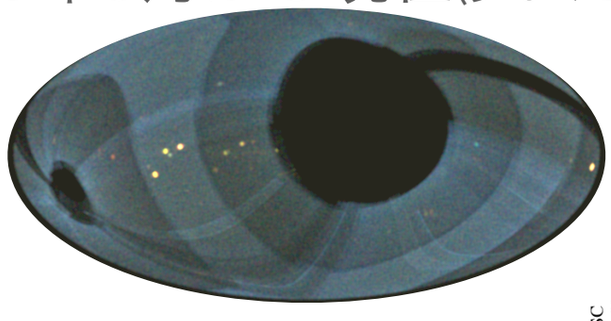
1. MAXIアーカイブデータの処理: 概要

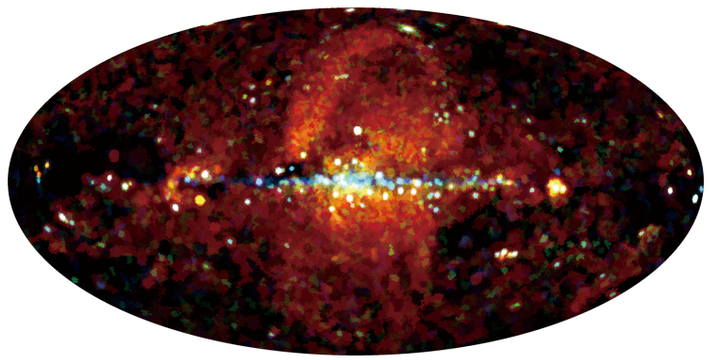




Monitor of **A**ll-sky **X**-ray **I**mage

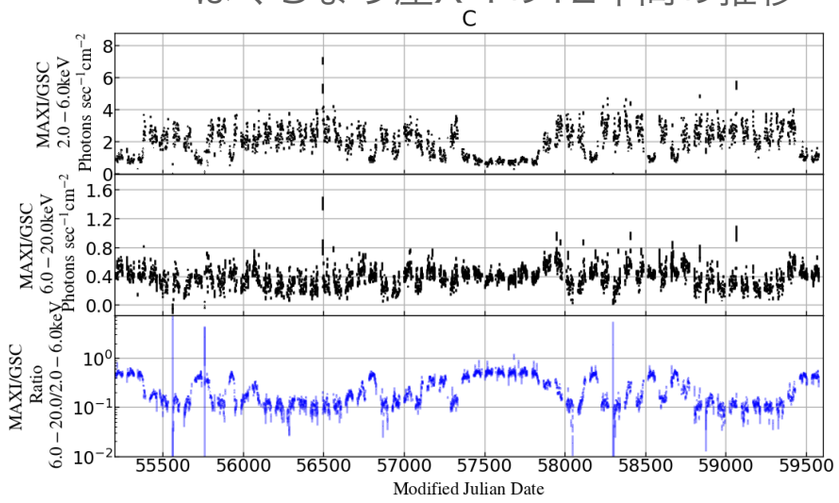
2009年8月から現在(少なくとも2023年まで)





積算した軟X線の全天画像

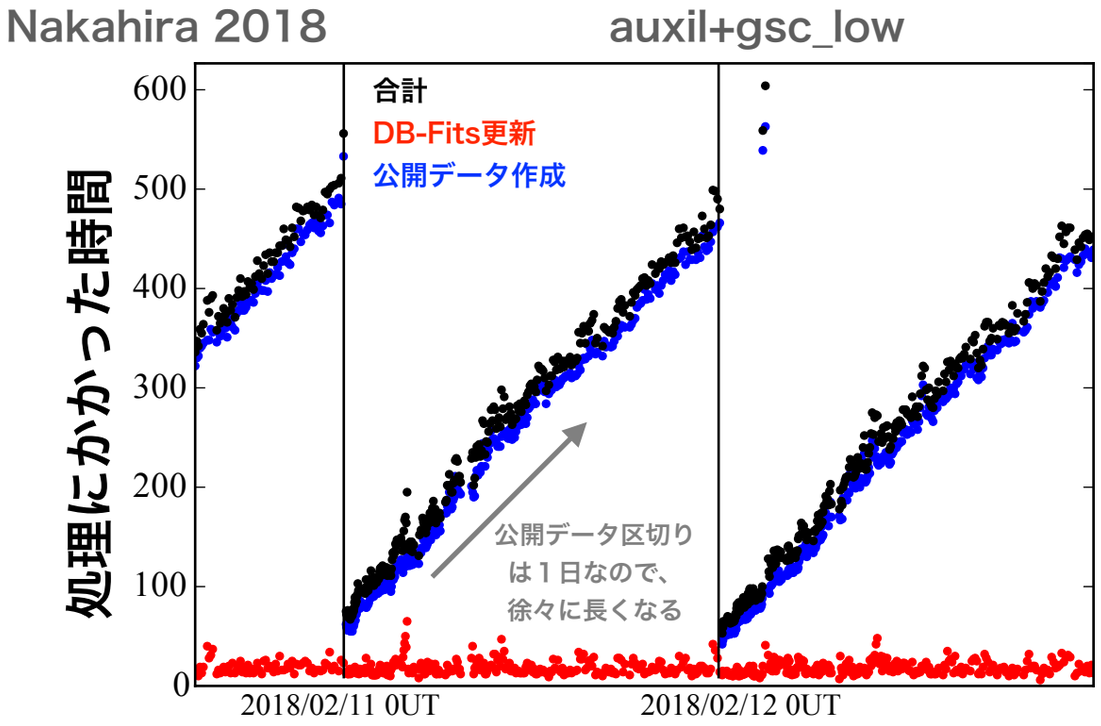
はくちょう座X-1の12年間の推移



- 軟X線で全天を監視(1日あたり>90%,92分毎)して天体変動現象を発見して通報して、観測を促すのが役割
- これまでに数百例の天体変動現象の速報を提供して、30程度のMAXIの名前がついた天体を発見した(ブラックホールや中性子星)
- ISS軌道上のラップトップにデータ解析ツールを移植して、軌道上でデータ解析してNICER望遠鏡と連携観測するOHMAN(**O**n-orbit **H**ook-up of **MA**XI and **N**ICER)が来月開始

1. MAXIアーカイブデータの処理: データの概要

- ~**70%はリアル**通信で、観測から数秒で地上のDB(PostgrasQL)にデータが登録される
- 宇宙研のDARTS(https://darts.isas.jaxa.jp/)からアーカイブデータを公開しており、データの速報的な価値を活かすために、公開データを逐次更新している



- 観測から2~8分遅れでデータ更新して公開領域に転送完了
- 更新ファイルは同じ名前で上書きし、Fitsのキーワードで識別
- 1日に300回程度更新される

- 当初のミッション期間は2年間だったが、既に12年超
- 観測期間の長期化によりプロセスの修正や較正の更新に伴うデータ更新が困難
  →10年経過時点の再処理には3ヶ月程度を要した

蓄積されたX線イベントデータ合計数
(約12年時点)

GSC低速 28.3×10^9
GSC中速 63.5×10^9
SSC中速 97.4×10^9

~2000億個

1. MAXIアーカイブデータの処理: 処理ツールについて

処理全体を駆動するのはPythonとperlのパイプラインスクリプト
+ Cで書かれた独自ツール、python(numpy)、GSFC/NASAのFtool等を組み合わせ

1. 補助データ作成

time(時刻補正用)
att (ads, iss, iss')
orbit
ISS太陽パネルの回転角
HK

ヒステリシスがあり時間で分割した
並列実行不可のためシリアルに処理
(処理は軽い)

2. X線イベントデータ作成

3種別のデータを作成

GSC低速(MIL-1553B) ←→ 元は同じデータだけど経路ごとの帯域に合わせて軌道上リダクション
GSC中速(Ethernet)
SSC中速(Ethernet)

X線イベント(10^7-10^8/day)1つ1つに対して

検出器座標→X線入射角度→空の方向(time, att)
PHAからエネルギー計算(pos, time, temp etc.)

観測条件が良好な時間を計算

→条件の悪い時刻のデータを除去

到来方向ごとにファイルを分割
(データ抽出の効率化のため)

3. Optional (後付のプロダクト)

VSC (スターセンサ)ダンプの可視光画像

姿勢と星像のカタログマッチングからWCSを計算

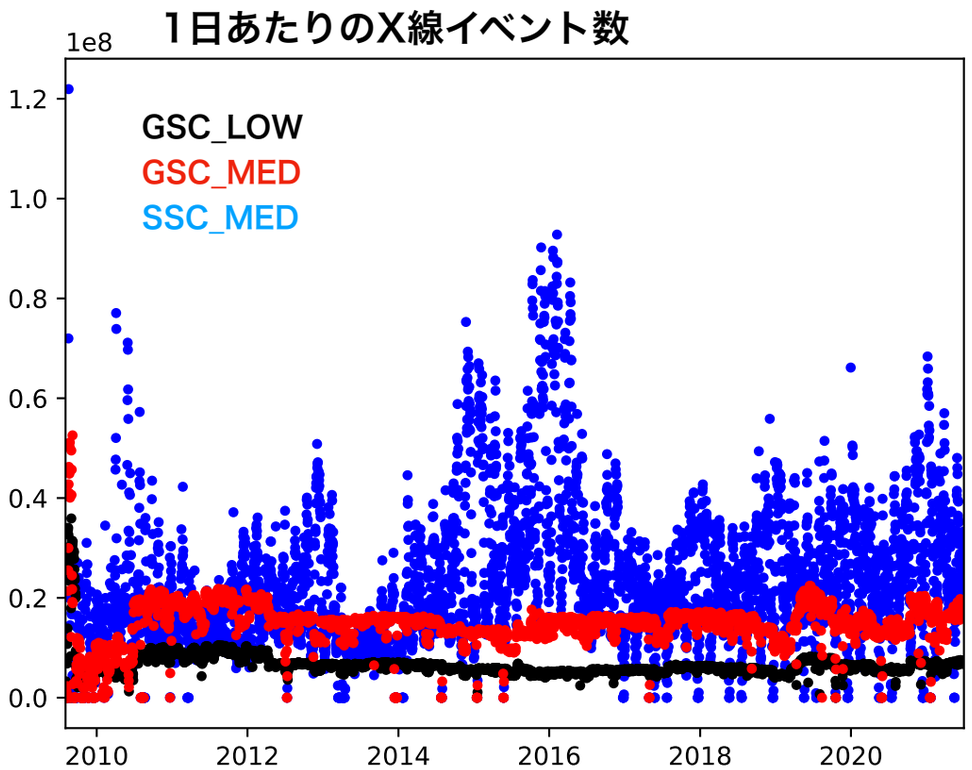
HiPS画像

RBM (荷電粒子モニタ)、地球物理向け

FITS、CDF

1. MAXIアーカイブデータの処理: スパコン実行環境構築

Singularityのsandboxで構築してそのまま実行:
基本的に元の実行環境で利用しているバイナリパッケージをまるごとコピー

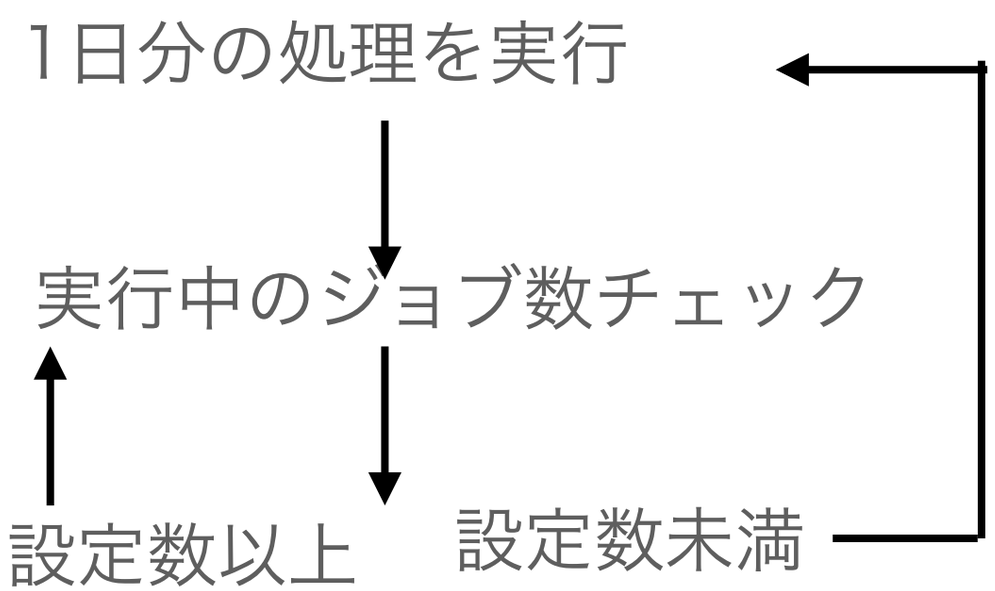


実行時間に与える影響はX線イベントの数が支配的。結果的に日によって数倍変わる

ジョブ投入: 主に2つの方法

A. 1ジョブずつ実行
B. ジョブ投入最適化ツール

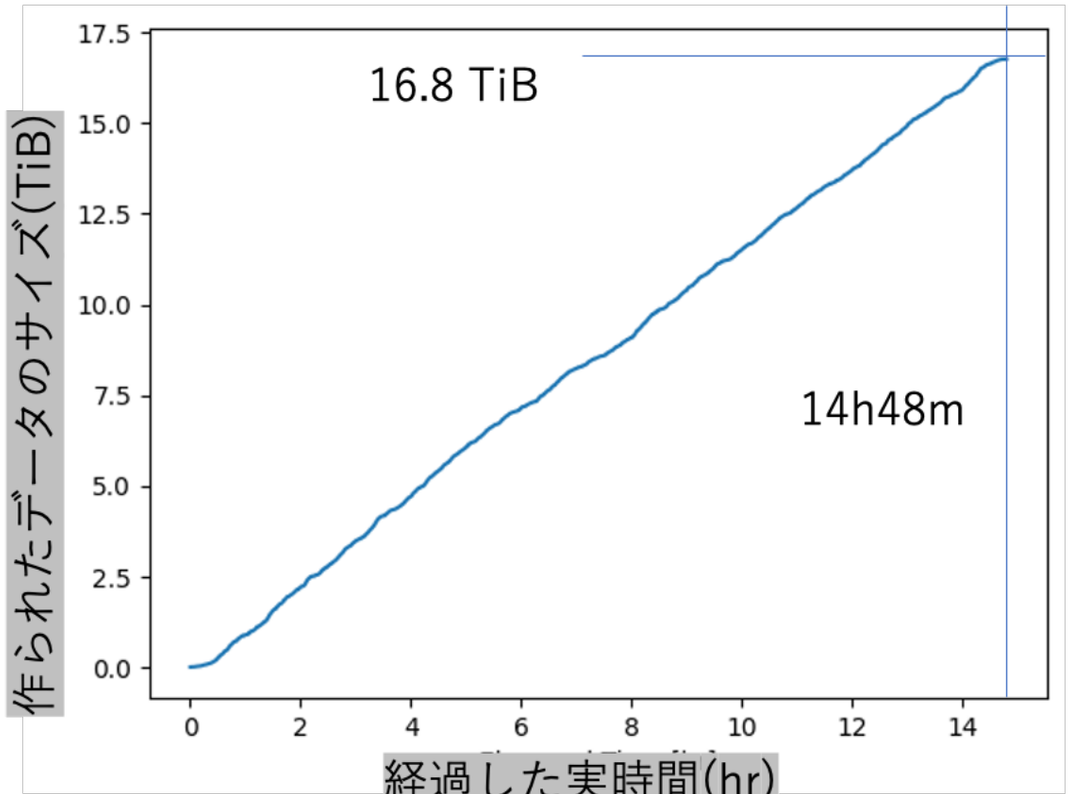
Bはまとまった単位ごとに投入し、終了を待つ
実行時間にばらつきがあると非効率

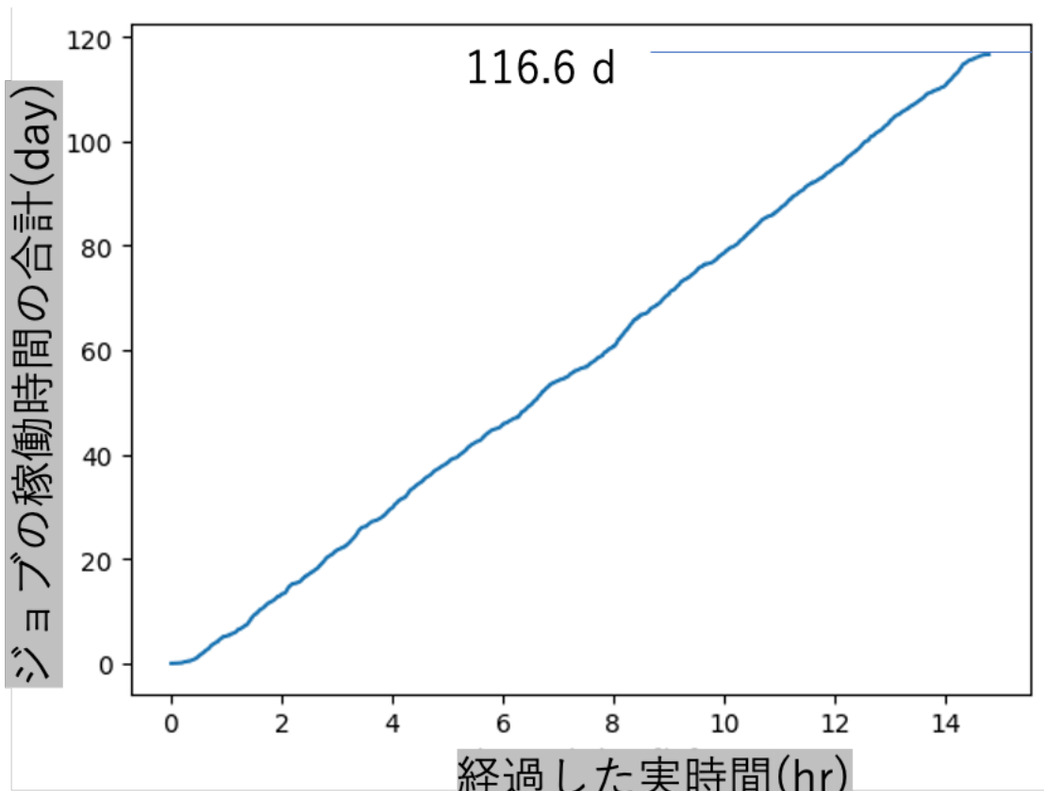


1. MAXIアーカイブデータの処理: スパコン実行結果

再処理対象: 2009年8月3日-2021年7月1日 (11年11ヶ月弱)

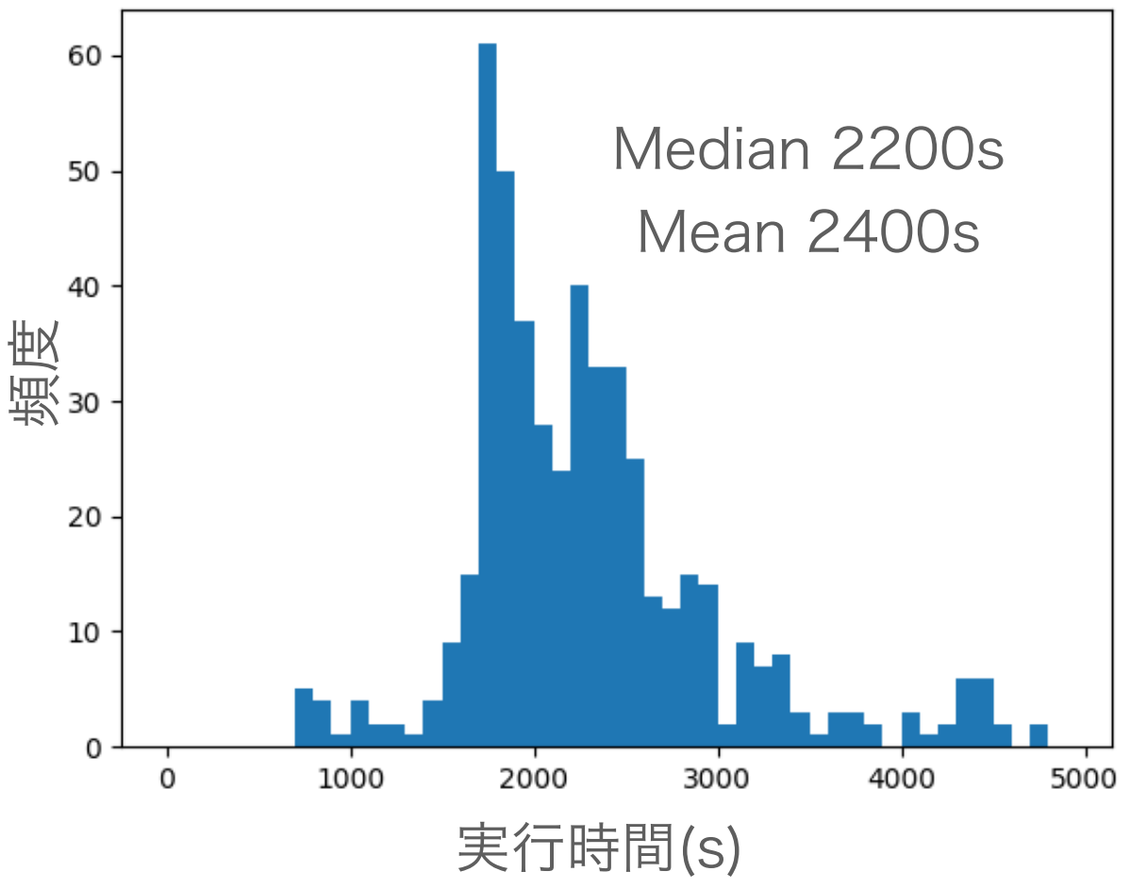
入力データとして9.5TBのFltsファイルをJSSのHDDにコピー
一時的データの書き込み左記にはSSDを指定
同時実行ジョブが200弱になるように調整





1. MAXIアーカイブデータの処理: スパコン実行結果2

MAXI側処理サーバーでの定常daily処理で
1日分データに要した処理時間(最近約1年)



定常的にはデータが増えるごとのリアルタイム生成(realtime)と別に、1日毎に過去の数日分を再作成(daily)している

自サーバーでのDailyの処理時間の概算見積り
データは~4330日分ある
~4330日×2200s=110d

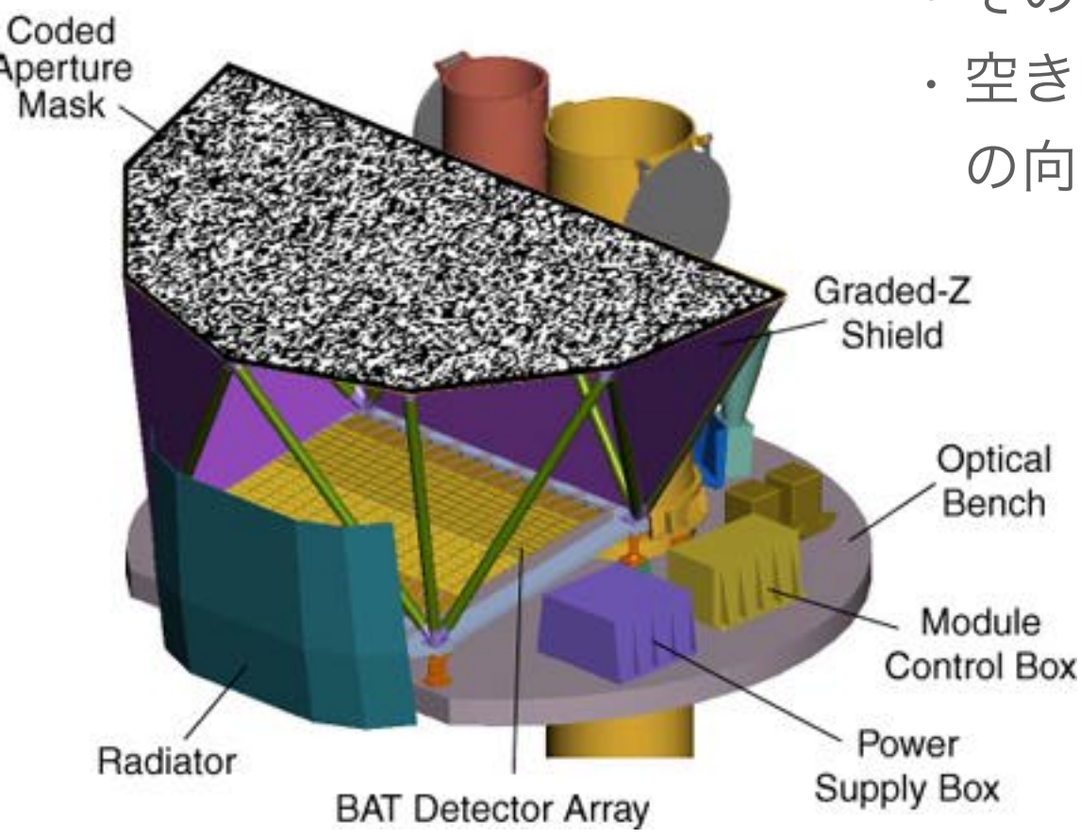
JSS実行結果の~116dayと大体等価

200ジョブ並列実行しても、1日分のデータの処理時間は、自サーバーと比較して大きな遜色はなかった!

2. Swift.BAT硬X線長期データ抽出

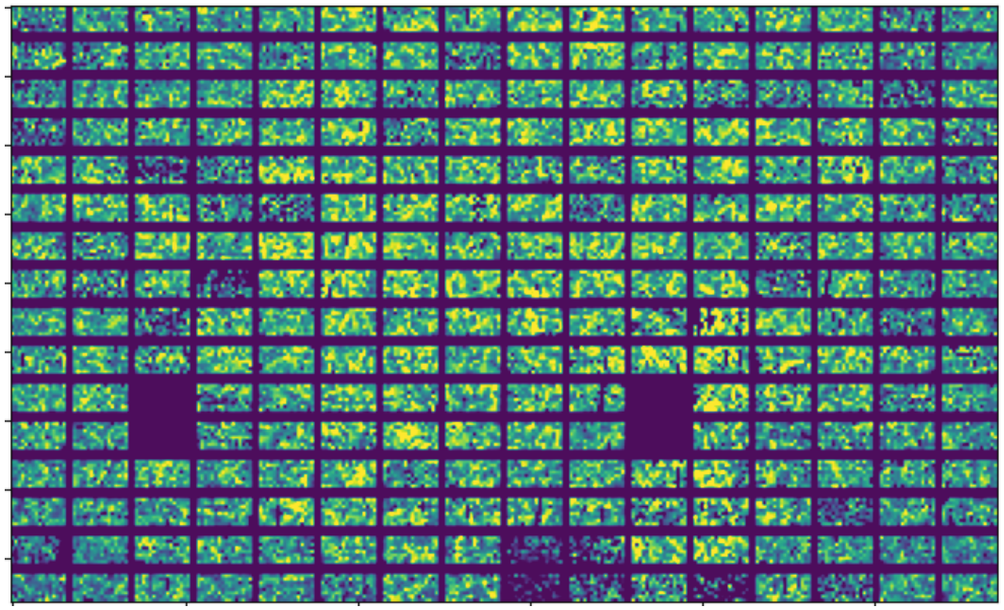
Neil Gehrels Swift Observatory

- 「ガンマ線バースト」観測を主目的とした衛星
- そのための装置が **BAT** (Burst Alert Telescope)
- 空き時間は望遠鏡(X,UV)で既知天体を観測しており、望遠鏡の向きに追随してBATが広い空(~1.4 Sr.)をカバー

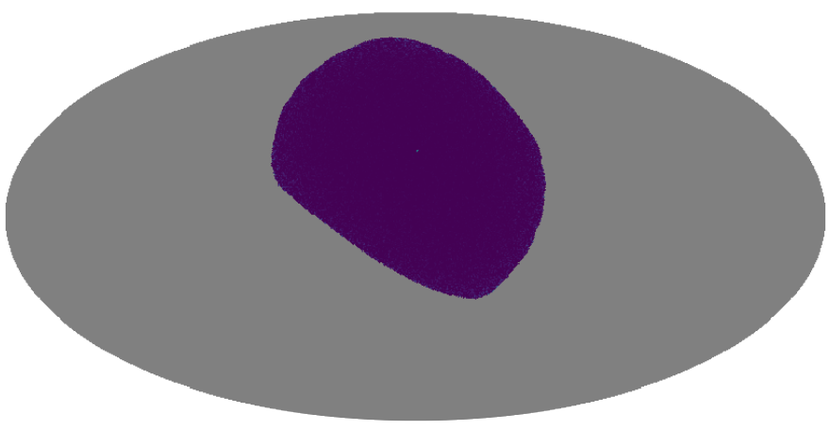


(c) GSFC/NASA

二次元符号化マスク+**C**d**Z**n**T**e array



Sco X-1に対するマスクパターン / 検出器面イメージ
(Sco X-1が正面にいるとき)

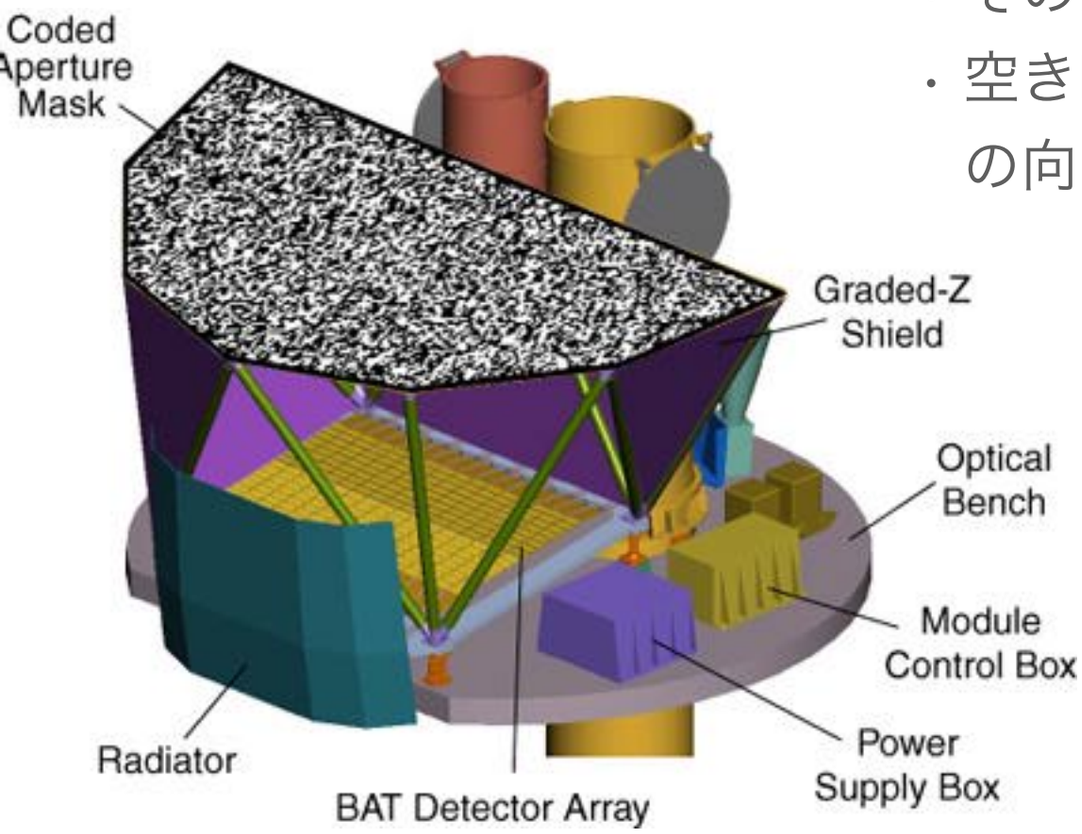


典型的に1-2ks観測して別の星に向きを変えるのでBATの「サーベイデータ」を観測すると多くの天体の硬X線強度変化を追うことができる
→新天体が見つかった際に過去の活動を知りたい!

2. Swift.BAT硬X線長期データ抽出

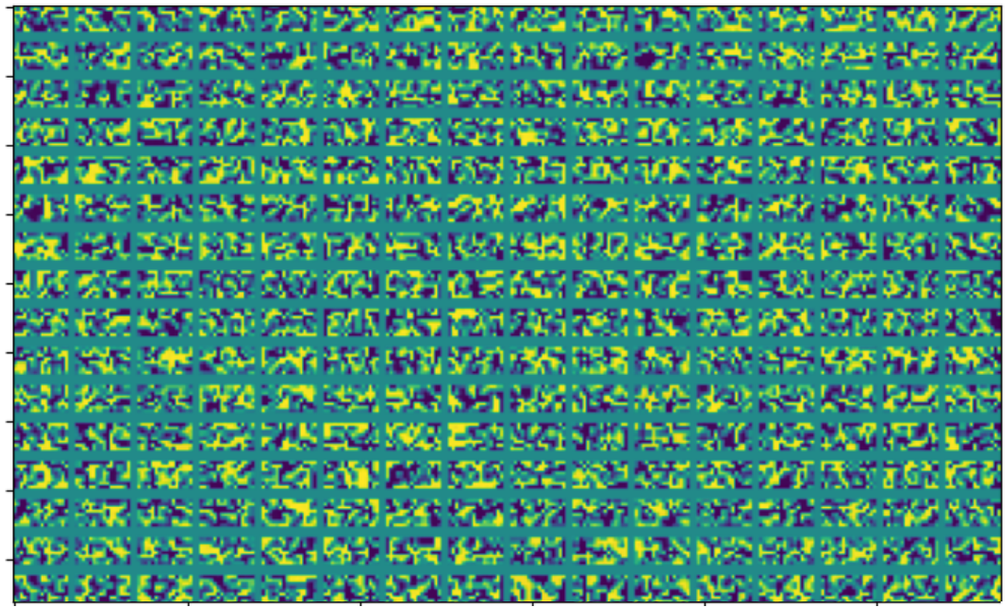
Neil Gehrels Swift Observatory

- 「ガンマ線バースト」観測を主目的とした衛星
- そのための装置が **BAT** (Burst Alert Telescope)
- 空き時間は望遠鏡(X,UV)で既知天体を観測しており、望遠鏡の向きに追随してBATが広い空(~1.4 Sr.)をカバー

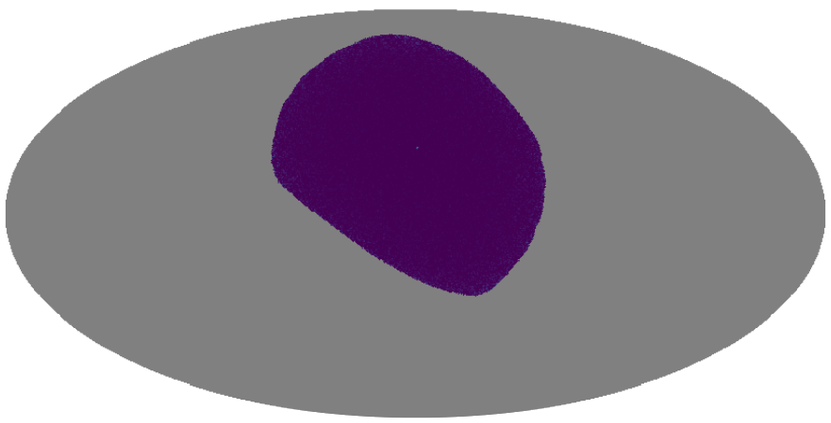


(c) GSFC/NASA

二次元符号化マスク+**C**d**Z**n**T**e array



Sco X-1に対するマスクパターン / 検出器面イメージ
(Sco X-1が正面にいるとき)

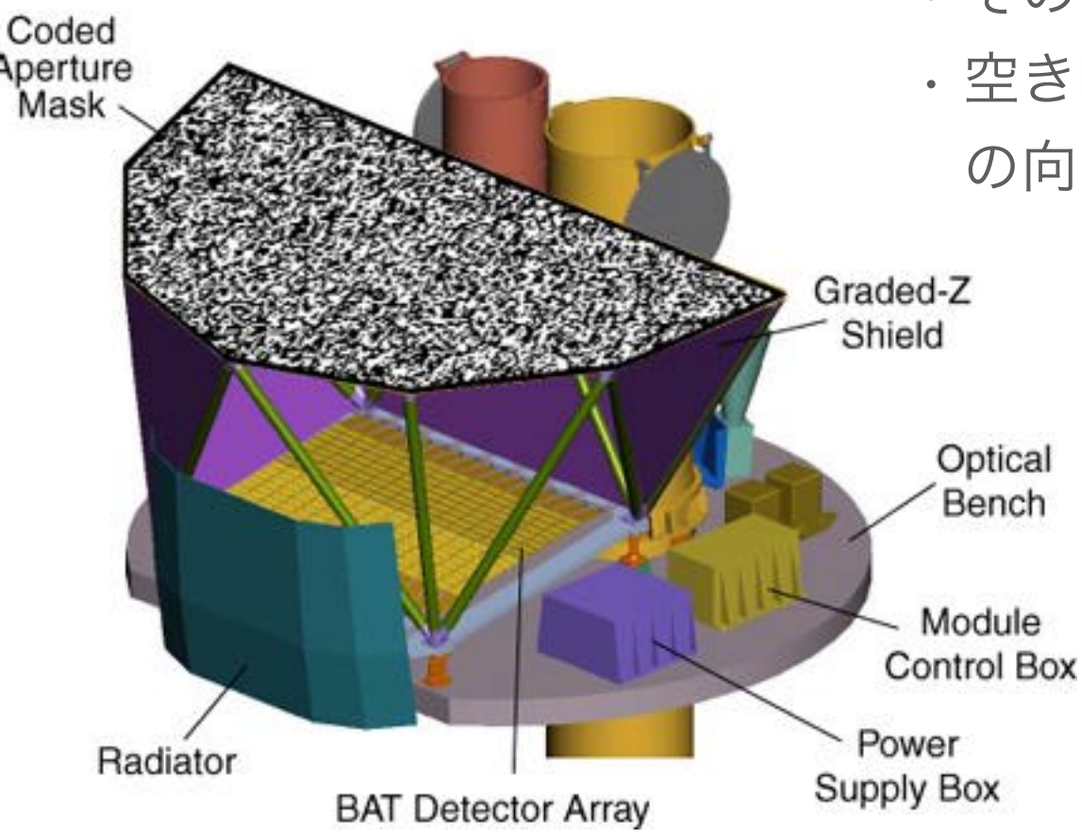


典型的に1-2ks観測して別の星に向きを変えるのでBATの「サーベイデータ」を観測すると多くの天体の硬X線強度変化を追うことができる
→新天体が見つかった際に過去の活動を知りたい!

2. Swift.BAT硬X線長期データ抽出

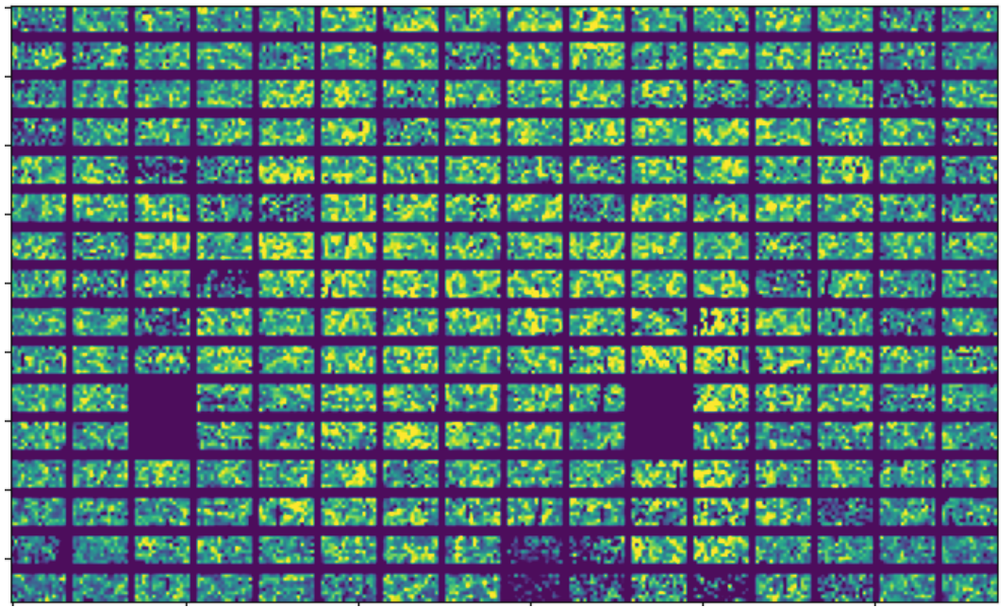
Neil Gehrels Swift Observatory

- 「ガンマ線バースト」観測を主目的とした衛星
- そのための装置が **BAT** (Burst Alert Telescope)
- 空き時間は望遠鏡(X,UV)で既知天体を観測しており、望遠鏡の向きに追随してBATが広い空(~1.4 Sr.)をカバー

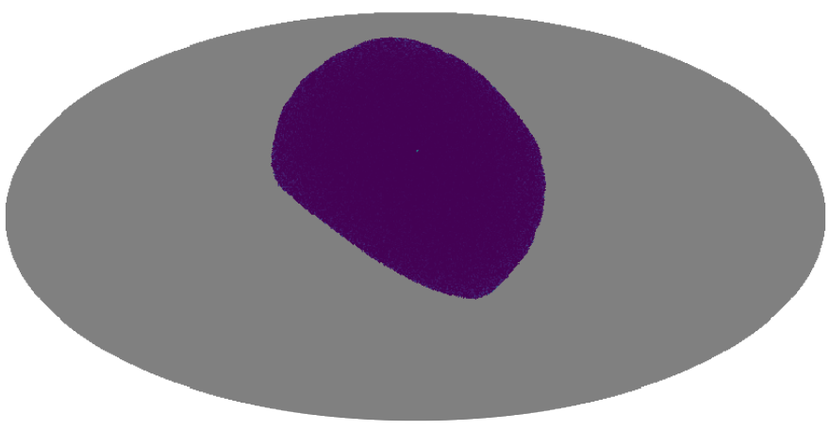


(c) GSFC/NASA

二次元符号化マスク+**C**d**Z**n**T**e array



Sco X-1に対するマスクパターン / 検出器面イメージ
(Sco X-1が正面にいるとき)



典型的に1-2ks観測して別の星に向きを変えるのでBATの「サーベイデータ」を観測すると多くの天体の硬X線強度変化を追うことができる
→新天体が見つかった際に過去の活動を知りたい!

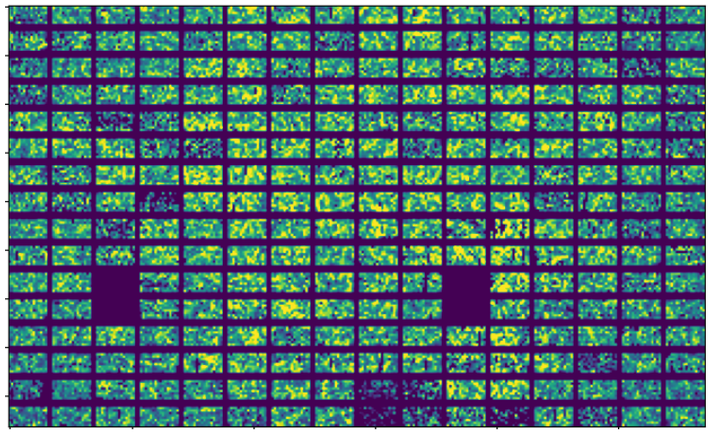
2. Swift.BAT硬X線長期データ抽出

データ解析の流れ

・全てGSFC/NASA提供のHEASoftパッケージ
・JSS3上でコンパイル(Singularity使っていない)

観測条件判定(利用できる時間が十分か等)

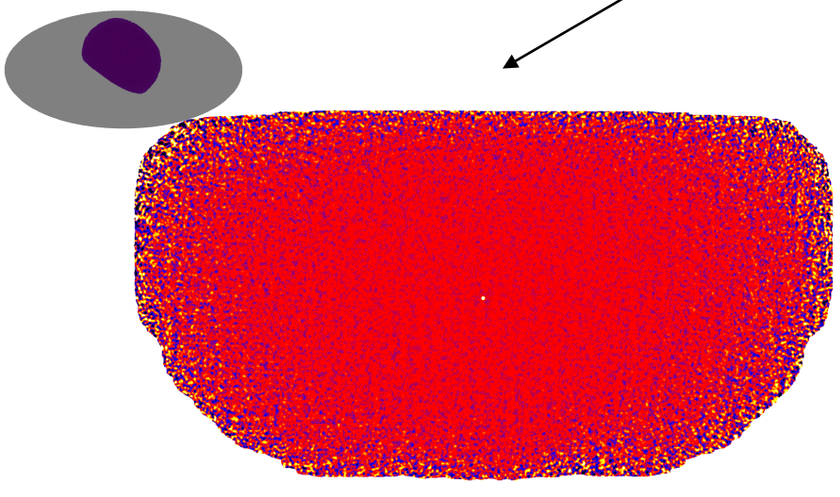
検出器面イメージ



(X,Y,PHA)のヒストグラム
5分程度で分割

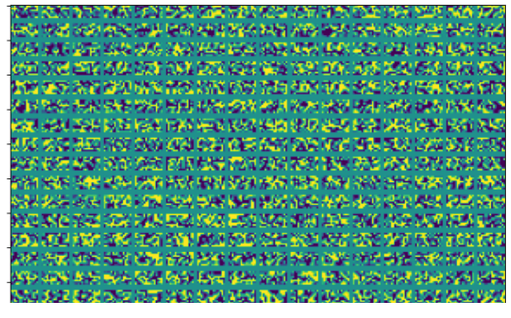
強度を計算
× 時間区切り
× エネルギー区切り

符号化マスクパターンを使って2D FFTして空の座標での画像を計算

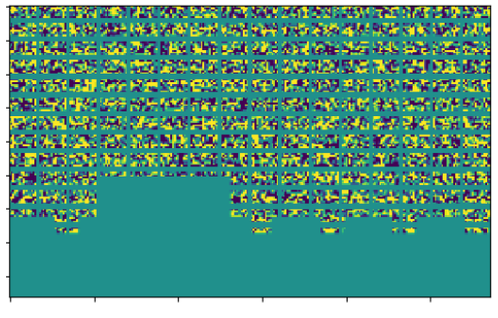


明るい天体検出

視野中心



やや端



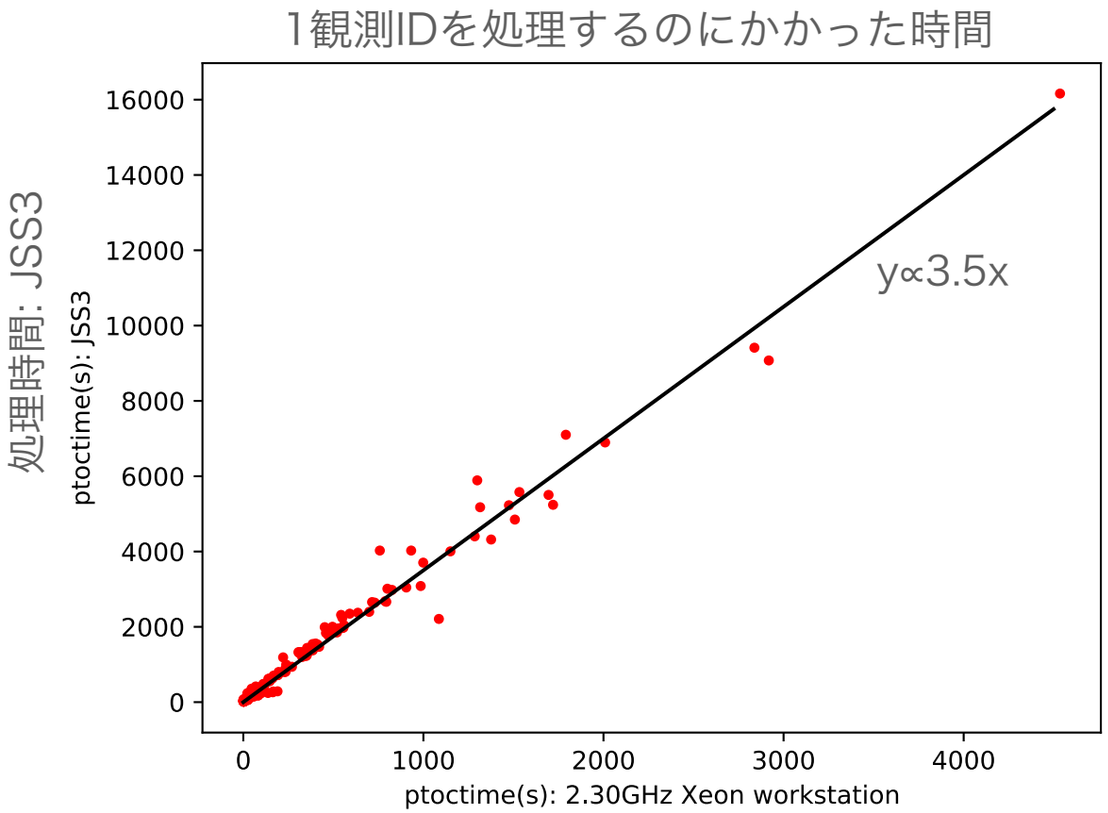
外から与えた解析対象天体リスト

対象天体に対するマスクパターン
× 対象天体数

2. Swift/BAT硬X線長期データ抽出

実行対象: 2004/12-2021/04

~26万pointing
~50万DPH
1.5 TB



処理時間: 手持ちのサーバー(2.3 GHz CPU)

JSS3が約3.5倍遅い!

正確な理由は不明だが

I/Oが遅くなっている感触

HEASoftはメモリを節約するためデータをDiskへ細かく書き出している印象。
中間データの書き込み負荷のため遅い?

3. 考察: 試験結果のまとめ

今回対象とした様なデータのの性質による制約

- 1つの入力データから1つの出力データを作る形になる
- 前後関係のあるパイプライン処理になる
- 一般公開されているソフトを多用して利用する(チューニングが効きづらい)

スレッドの並列化ができる部分は限定的で期待できない
そのため主に、独立な複数プロセスを同時に起動することが高速化の手段

シミュレーションの様に数万コアを生かした高速化はできないが、10年分の処理を100倍程度早く(数ヶ月が1日!)処理することはでき、衛星データの再処理目的には十分利用できそう

3. 考察とまとめ: 衛星データ処理における利用モデルの検討

宇宙研の科学衛星データの高次データ処理基盤として「Reformatter」と呼ばれるサービスが用意されている、JSSをどの様に組み込めるか?

◯独立なデータを並列実行して高速に処理可能
△依存関係のあるデータは逐次実行するしか無いので、高速化に寄与しない場合が多い
　→毎日の増加するデータ差分の処理に使う利点は小さい→主に一括再処理
x メンテナンスによる停止期間が比較的多い
　→定常的な処理はReformatterに残して問題ない(SLAは無いが実績としては年間2-3日の停止)
✓ 並列処理性能を活かすためにはデータを近くに置く必要がある
　→データをNFS等で共有して使うメリットは薄い

3. 考察とまとめ: 衛星データ処理における利用モデルの検討

○独立なデータを並列実行して高速に処理可能

△依存関係のあるデータは逐次実行するしか無いので、高速化に寄与しない場合が多い

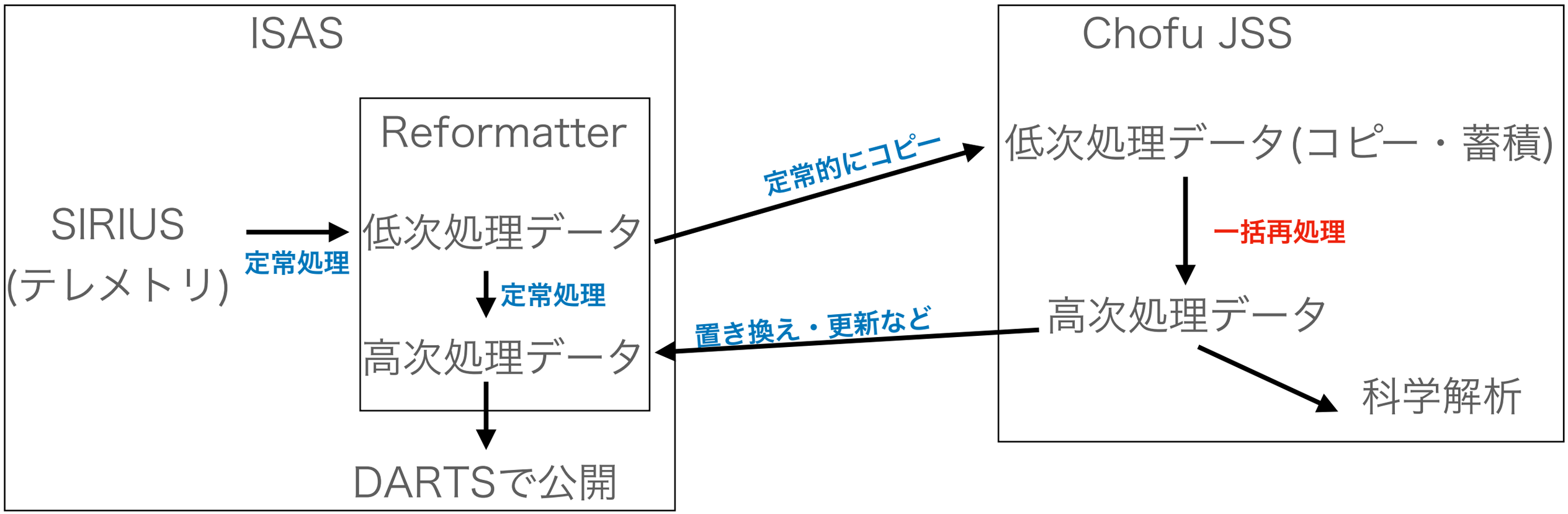
→毎日の増加するデータ差分の処理に使う利点は小さい→主に一括再処理

x メンテナンスによる停止期間が比較的多い

→定常的な処理はReformatterに残すのが良い (SLAは無いが実績としては年間2-3日の停止)

✓ 並列処理性能を活かすためにはデータを近くに置く必要がある

→データをNFS等でexportして使うメリットは薄い



今後Reformatterとしてのサービスに組み込むか、個別に使ってもらうのかを検討したい