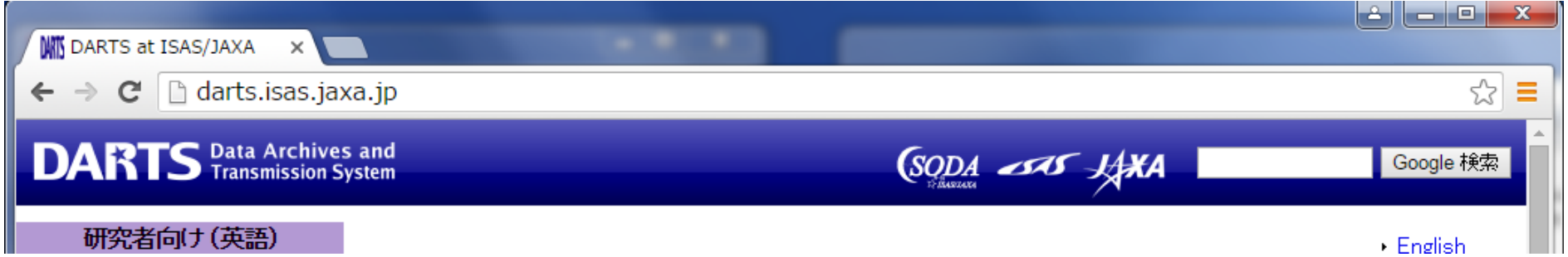


宇宙科学データアーカイブDARTSの現状と課題

殿岡 英顕、松崎 恵一、海老沢 研、山本 幸生、北條 勝己、藤嶋 幸美、稲田 久里子
宇宙航空研究開発機構 宇宙科学研究所　科学衛星運用・データ利用ユニット

2016/2/12
宇宙科学情報解析シンポジウム

This document is provided by JAXA.

本講演の目的

- 宇宙科学データアーカイブDARTSの現在の構成及び過去の運用上の問題点とその解決策をまとめ、報告する。
- DARTS URL .. http://darts.isas.jaxa.jp/

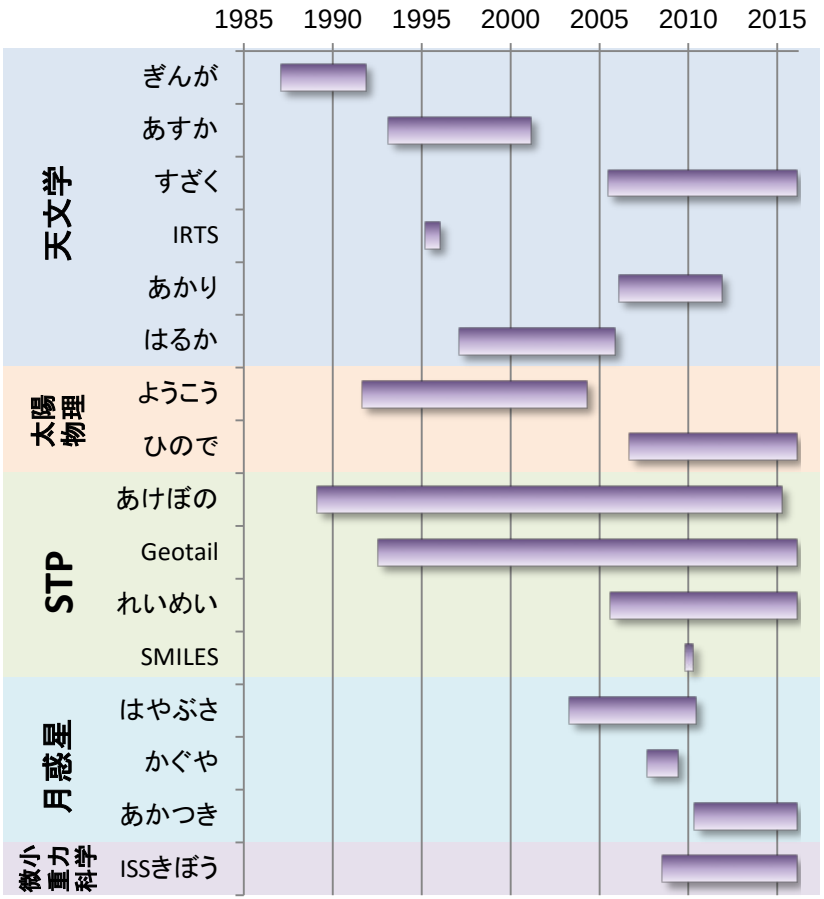
This document is provided by JAXA.

目次

This document is provided by JAXA.

1. DARTSの特徴

- 多様なデータのアーカイブ
  - DARTS は天文学、太陽物理学、STP、月惑星科学、微小重力科学といった多くの学問分野にまたがるデータの集積を行っている。
- 1997年にあすか、ようこうのデータ公開から始まった。
- 現在は16のJAXAの宇宙機・宇宙実験のデータを公開している。
- データの長期保存を目標の一つとする。



This document is provided by JAXA.

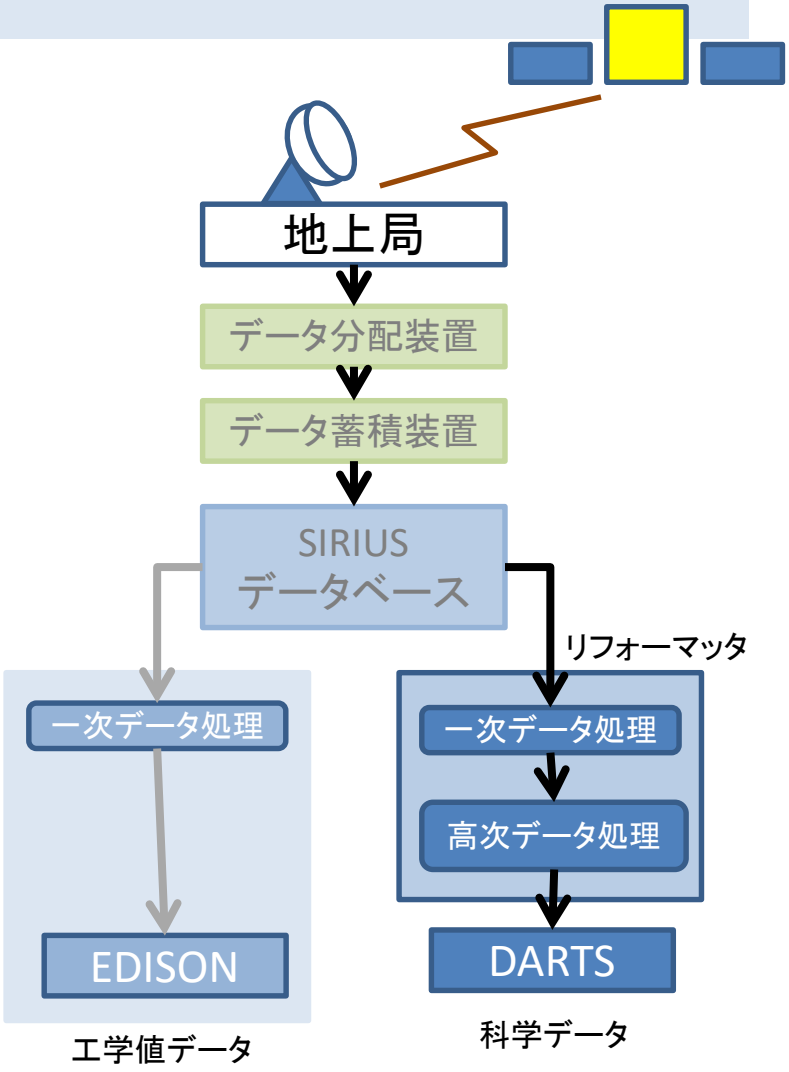
2. DARTSの計算機構成

- ほとんどのサーバを科学衛星データ処理システム(DPSS)仮想マシンで構築。
- ストレージは DPSS の共有ストレージを利用。
- 5年ごとにDPSSのシステムが更新される。前回は2013年9月に更新があった。
- OSは Linux (Red Hat Enterprise Linux 6)を利用している。

This document is provided by JAXA.

データの出口としてのDARTS

- 宇宙機から送信されたデータは、地上局で受信されたのちに、リフォーマッタにてデータ処理をされて科学データになる。
- DARTSは処理された科学データを全世界に公開する。



This document is provided by JAXA.

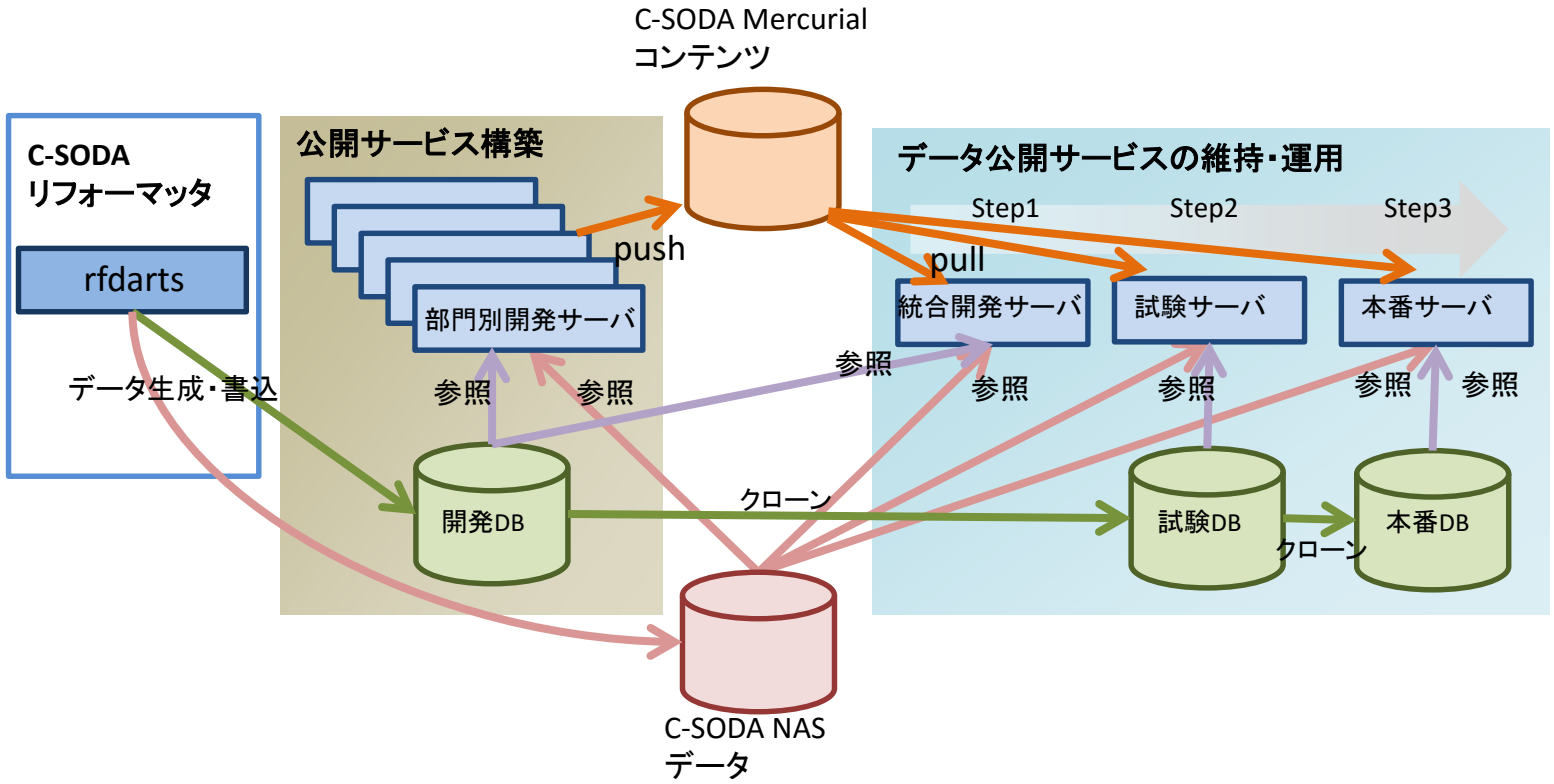
DARTSの主流サービス

- DARTSでは、データの公開と検索サービスの大半を1台の公開サーバで行っている。(ここでは主流サービスとする)
- 試験的サービス(DARTS Labs.)などは、主流サービスとは別のサーバで行っている。(非主流サービス)

This document is provided by JAXA.

DARTS標準システム

- DARTSの主流サービスを行うサーバ(標準システム)では、1台の本番サーバに対して多くの補助サーバが存在している。



This document is provided by JAXA.

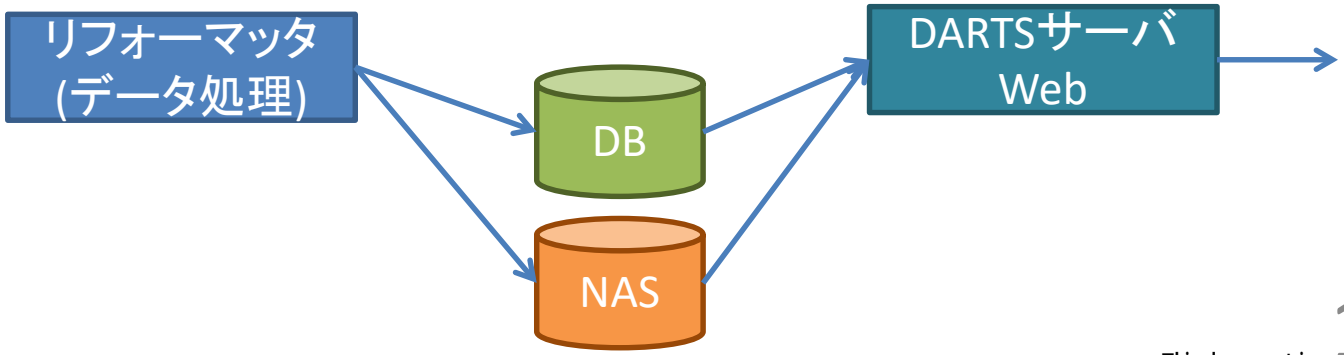
3. DARTSの計算機運用

- サーバの役割分担の明確化
- コンテンツのリリースがサービスに影響を与えないサーバ構成
- OSアップデートに対して影響を最小限にするサーバ構成
- 運用分担の明確化

This document is provided by JAXA.

サーバの役割分担の明確化

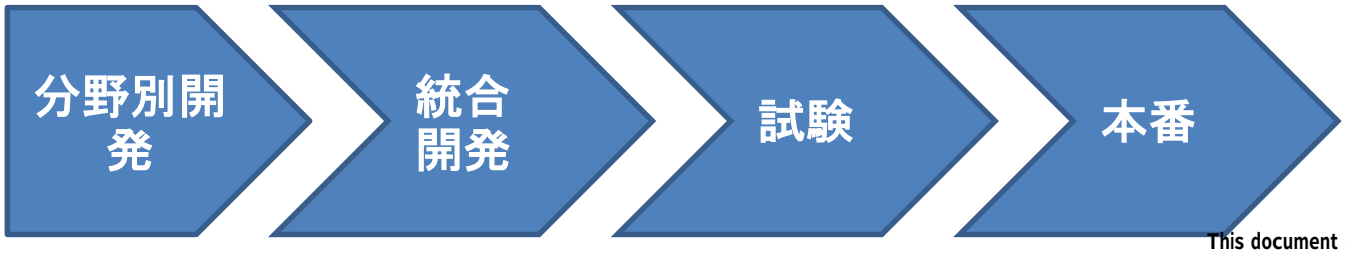
- リスク例
  - 外部に公開されているDARTSサーバがクラックされて、データが消失する。復旧のために多大な時間をとられる。
- DARTSサーバではデータストレージは Read Onlyとし、データの加工(書き込み)は基本的に行わないようにして、役割を分けた。
- データの加工はリフォーマッターで行う。
- 万一 DARTS サーバがクラックされても、被害範囲を限定できる。



This document is provided by JAXA.

コンテンツリリースがサービスに影響を与えないサーバ構成

- リスク例
  - 不具合のあるコンテンツ(HTMLページ、アプリケーションもしくは設定)のリリースにより、本番サーバのサービスが停止する。
- 主流サービスは 3(+1)ステップ のリリース手順とし、事前に不具合を検出できる体制を構築した。
  - 分野別開発機: 開発者がコンテンツの開発を行うマシン。
  - 統合開発機: DARTS管理者が全分野を統合した状態での動作確認を行うマシン。
  - 試験機: 本番に近い状態で試験を行うマシン。JAXA外の関係者からも確認を行えるようにしている。
  - 本番機: 実際に外部に対するサービスを行うマシン。
- 非主流サービスについても、開発機と本番機を分けたサーバ構成を取っている。



This document is provided by JAXA.

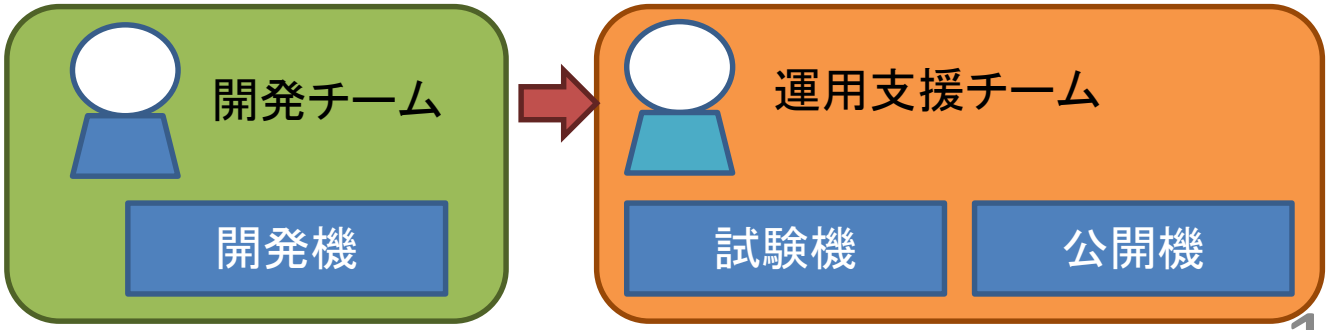
OSアップデートに対して影響を最小限にするサーバ構成

- リスク例
  - OSアップデート時にパッケージ間の不整合により、サービス停止もしくは対処に時間がかかるトラブルを伴う予期せぬ障害が出やすくなっている。
    このような不整合はOS供給者以外のサードパーティパッケージで起きやすい。
- 主流サーバはOS標準パッケージで構成し、アップデート不具合によるサービスの全停止を防ぐ。
- サードパーティパッケージに依存するサービスは極力避けるか、非主流サービスとしてサーバを分割して提供する。(部分停止はやむを得ないと考える)

This document is provided by JAXA.

運用分担の明確化

- リスク例
  - 開発チームと運用チームが同一だと、なれ合いのためサーバの構成変更情報の記録を忘れることがある。構成情報の記録漏れは、システム更新の再構築時に大問題となる。
- DARTS開発チームは極力管理者権限を持たない。構成変更は運用支援チームに依頼する。運用支援チームには記録を取ることを手順化し、残してもらう。



This document is provided by JAXA.

4. DARTSの開発

- 手堅いアプリケーション開発
- Mercurial リポジトリの活用

This document is provided by JAXA.

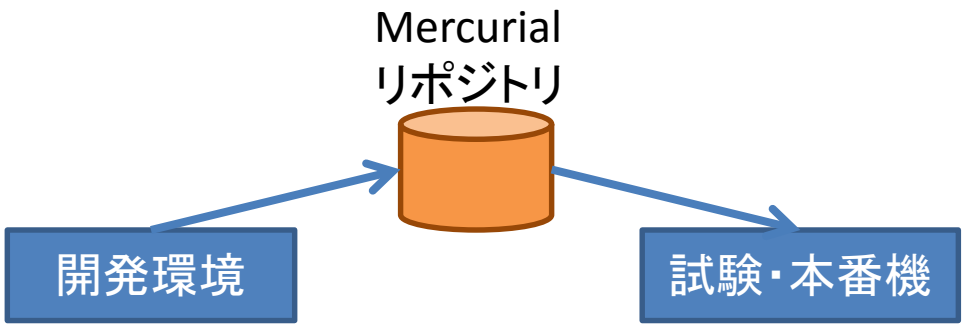
手堅いアプリケーション開発

- アプリケーションの保守コストを上げるプログラム言語のリスク要因
  - 仕様がころころ変わるものは、バージョンアップと共に修正が必要となる。
  - 更新が止まってセキュリティパッチが供給されなくなるとサービスが停止する。
  - フレームワークは流行り廃りが激しく、習得コストもばかにならない。さらにバージョン依存があると管理が煩雑。
- 理想的には開発後に手をかけずに長期間安定して利用できるものがベスト。
- 開発後になるべく保守を必要としない言語を選ぶ。現在は PHP が主流。
- とはいえ、将来何が起きるかわからないのも事実。(例: Adobe Flash の終焉、など）

This document is provided by JAXA.

Mercurial リポジトリの活用

- コンテンツ開発でのリスク例
  - コンテンツを更新したら不具合が発生したが、どこを変更したかわからない。
  - コンテンツを長期間更新し続けるうちに、状態がわからなくなった。
- コンテンツの履歴管理のために、履歴管理ソフトMercurial を利用し、変更点の記録を残すようにした。
- さらにプロジェクト管理ソフト Redmine と組み合わせることで変更点を容易に確認できるようにした。
- Mercurial を境界点として、コンテンツの開発フェーズと展開フェーズ(試験・本番)を明示的に切り分けることができるようになった。



This document is provided by JAXA.

5. まとめと課題

- DARTSでは、サーバの安定的な運用を行うために、様々な工夫を取り込んできた。
- 手堅いサーバ運用のためには規則が多くなる傾向にある。だが、時代の変化により状況は変わるため、変化に応じた規則の適用条件の調整は必要である。
- 自由な開発への対応も検討すべきである。一例として、手堅く運用するサービスと新たな技術で開発するサービスの分離を進めることが考えられる。
- コストが上昇する傾向に対して、掛けられるリソースは限られている。リソース不足に対して開発を止めない策の検討を開始している。(外部リソースによるアプリケーション開発など)

This document is provided by JAXA.