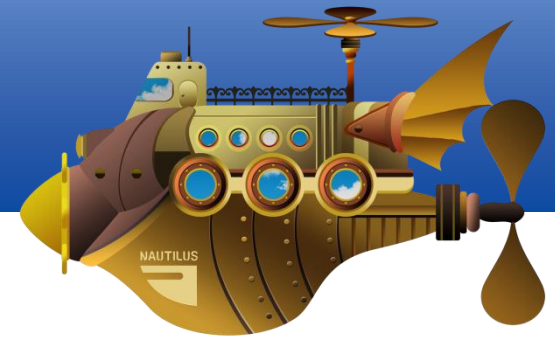


Tsurugi(劔)について

2020/10



 **NAUTILUS**

概要

■ RDBMSの開発

- RDBMSの開発を目的としたプロジェクト
 - DB系Tx系の企業・有志グループ・ユーザ会等により現在開発中です
- NEDO案件（経産省）
 - このProjectの成果は、国立研究開発法人新エネルギー・産業技術総合開発機構（N E D O）の委託業務の結果得られたものになるものがあります

開発の目的

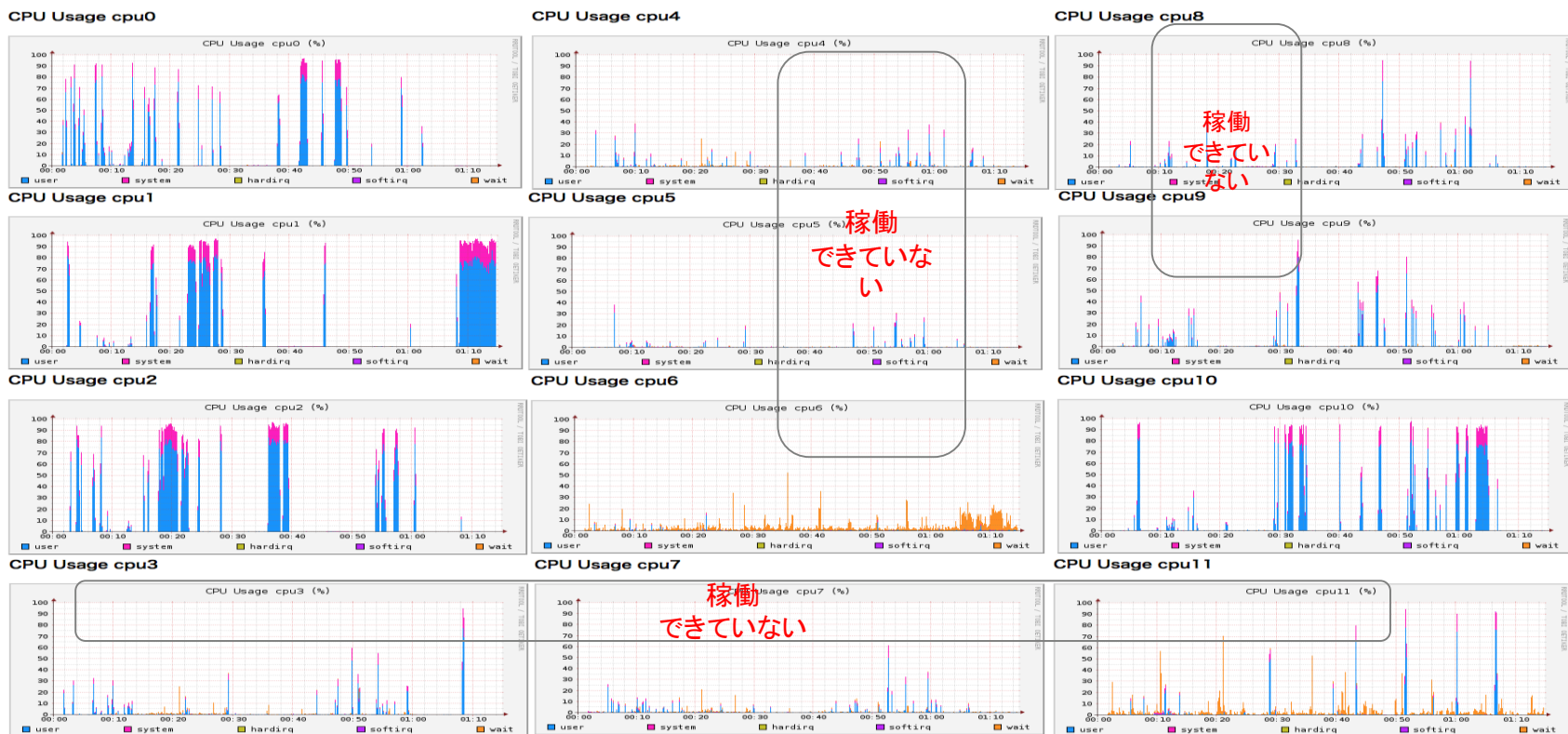
- 選択肢の確保
 - 現状のH/Wアーキテクチャに合致したRDBMSの選択肢をより増やす
- 透明性
 - より広く利用可能であること
 - 実装透明性を確保するためにOSSで開発する
- 受け入れやすいものを
 - エンジンを中心：外側はPostgresそのものとする
 - 既存ユーザに受け入れやすいようにする
- HTAPで利用可能な形態にする
 - 分析と書き込みの両方ニーズを満たす必要がある
- コミュニティベースの開発を企図
 - 可能な限り国内のDB研究者や開発者を集めて共同でつくり上げる
 - すでにユーザ企業を中心としたコミュニティが発足している
 - 協調関係をとりながら加速させる

課題: 端的に言えば「今のRDBはスケールアウトしない」

■ 抜本的なアーキテクチャの問題

- Diskベース+CSRベース→メモリーベース+MVCCを生かし切れていない
 - 特にWriteのロングバッチ処理は致命的に弱い
- アーキテクチャの変更が困難 ← 「イノベーションのジレンマ」

実際のバッチ処理でのパフォーマンス: 12コアであっても4コアしか利用できていない



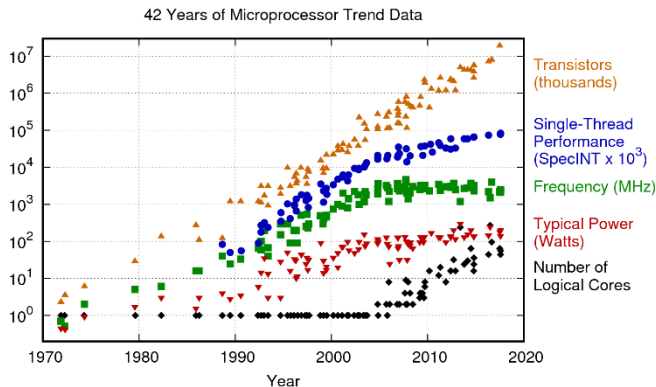
背景

■ 「DBルネッサンス」

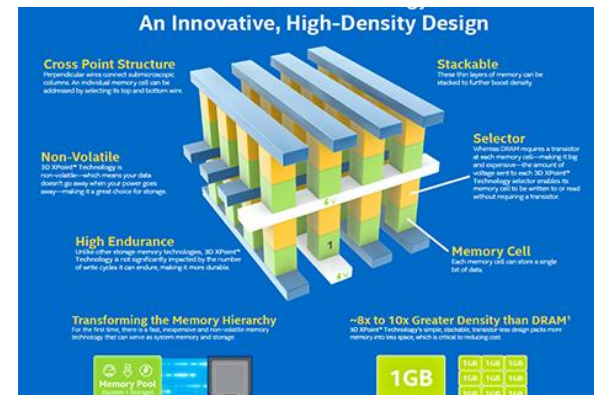
- メニーコア・インメモリーの要請
- ムーアの法則の終焉によりプロセス微細化は限界にあたりつつある
- この結果として半導体業界はCPUのメニーコア化を進めつつある

■ メモリーデバイスの進化

- メモリーの高機能化と高密度化
- 不揮発性メモリーの発展が進んでおり、NVM/SCM NVDIMM等の高速の不揮発性メモリーも登場し、市場に投入されつつある
- 同時にメモリーの高密度化も進みつつあり、サーバあたりのメモリー容量もTByteクラスまで伸張しつつある



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Laborte, O. Shacham, K. Okukotun, L. Hammond, and C. Batten
 New plot and data collected for 2010-2017 by K. Rupp



背景

- アーキテクチャ的な限界～既存の環境の変化
 - 従来のRDBMSは、ハードウェアはコア数は少なく、メモリー容量は制限的という前提を敷いており、現在に至るまで基本アーキテクチャの思想は変わっていない
 - In placeまたは2-versionまでのページモデルを前提とし、リカバリーはARIESをベースにしている。結果として、ハードウェアの性能が高進しても、それを生かし切れずスケールアップに限界がきている
- 商用DBの減少
 - 現状の商用DBはベンダーの統合の結果、その数が減少している
 - 企業システムにおいてはほぼOracle社/MS社の寡占状態にあるといつてよく、ユーザの選択の幅は極めて狭い

背景

■ OSS-DBの課題

- 完成度の向上とともに、コードベースの増大・ステークホルダーの増加が起こっている
- 抜本的なアーキテクチャの変更は難しい
- 実装変更の合意が取りにくく、環境の大幅な変更には追従できなくなっている

■ クラウドDBが間尺に合わない

- クラウドDBは各クラウドベンダーの自社ビジネスを背景に開発されたものが多く、既存ユーザのスケールには合わないことが多い
- 透明性が欠如している
 - 結局のところ中身がわからない
 - RDBで原理的に無理な部分はアプリ側で手当することが多いが、そもそものその原理が不透明

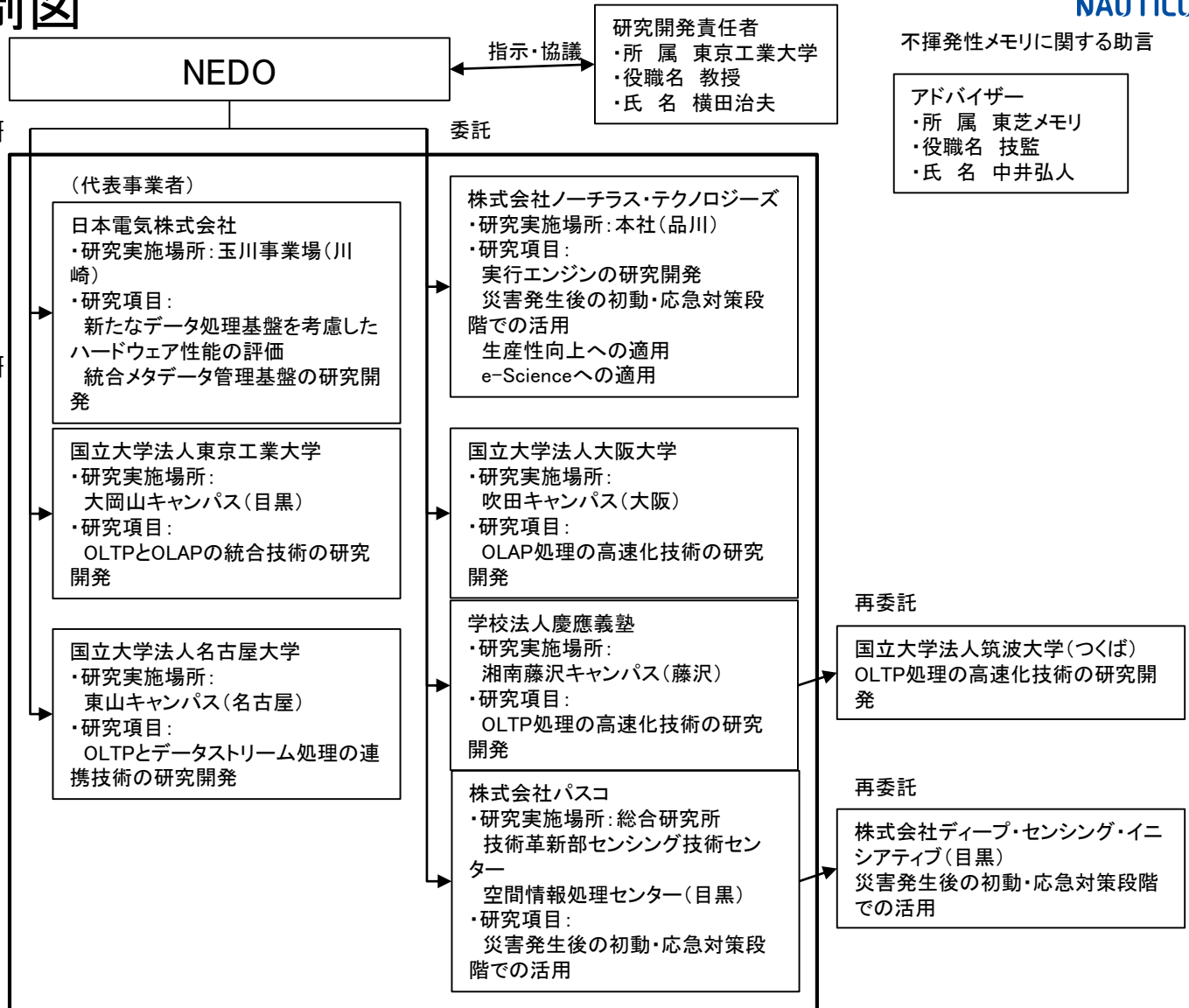
メンバー

- 当初の起こり
 - DB/TXの勉強会の有志から開始
 - もともとは「いつまでたってもできそうもないで、集まってゆっくり作りますか？」というスタンス
 - NEDOのサポート
 - NEDOへ応札したらどうか？という話があって、有志内部で調整
 - 概ね反対
 - 「紐付き予算はオーバーヘッドが大きい」というのが理由
 - いろいろ調整
 - 参加組と不参加組で分かれている、ただし活動としては連携している
- 要するに「親方日の丸」というにはほど遠い。
- やれる人間でプロダクト経験のある企業を中心に「とにかく必要なものを作る」というスタンス
 - うまくいかなくてもNEDOや国の責任ではありません
 - 念のため



NEDOでの体制図

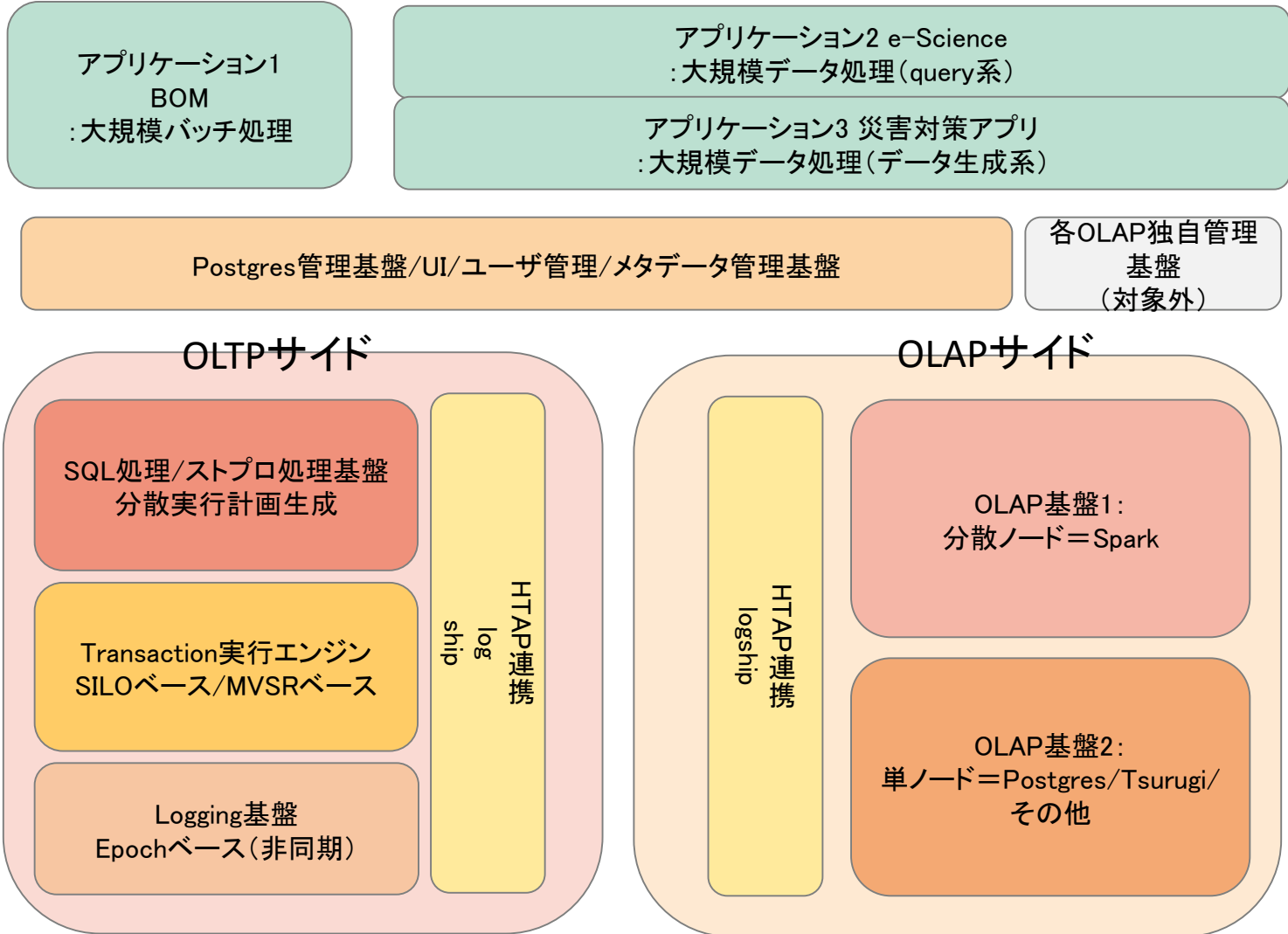
- HTAP
 - 担当：東工大・横田研
- クエリー最適化
 - 担当：阪大・鬼塚研
- Txエンジン
 - 担当：慶応大・川島研
- 空間処理
 - 担当：名大・石川研
- 実行エンジン
 - ノーチラス・NEC
- 不揮発性メモリー検証
 - NEC



不揮発性メモリに関する助言

アドバイザー
 ・所属 東芝メモリ
 ・役職名 技監
 ・氏名 中井弘人

構成要素



構成要素: プロトタイプ アプリケーション

■ 目的

- Tsurugiの使い方の例示として提供

- 1. 高速BOM
 - マスター更新処理とBOM再計算バッチの超高速化
 - 現状ではBOMの在庫データ・マスター更新と、原価シミュレーションの基盤を統合する
 - 生産効率の向上と意思決定の高速化が期待できる
 - 大規模バッチ処理と通常クエリーの併存

- 2. 災害対策マップリアルタイム作成
 - 大規模災害時の初動マップを航空機からのデータをセミリアルタイムで処理し、速やかな減災情報を生成・DB化する
 - 高速データ生成とそのDB化

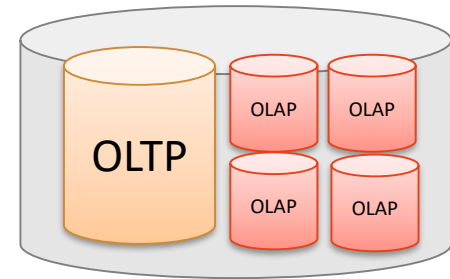
- 3. 広視野深宇宙データ処理アプリケーション
 - 国立天文台のデータを大規模・高速DBとして構築する
 - 重力波の発見等の世界的な新発見のスピードを世界一にする
 - 大規模クエリーの高速化

構成要素: OLTP

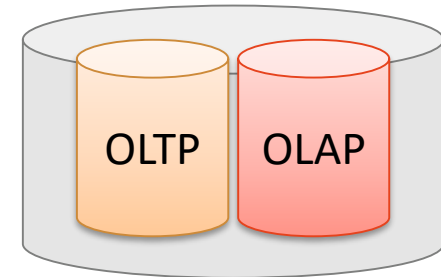
- 管理基盤
 - すべてPostgresベース
 - ユーザ管理
 - メタデータ管理
 - その他システムの管理基盤
 - 特にユーザビリティにかかわる部分は可能な限りPostgresによる
- SQL処理系
 - 可能な限りSQL互換
 - 実行計画エンジンはメニーコア・大規模メモリー用にフルスクラッチ
 - ストアド・プロシジャー
 - できるだけPostgres互換
 - ただし、分散実行可能なようにシンボル/注釈追加の予定
- TXエンジン
 - 複数準備ができればしたい
 - TS/MVCCベースは共通
 - SILOは確定 (write lock除去版)
 - MVSRは現在検討中
 - バッチ処理用の仕組みを導入する
- logging
 - TXエンジンと疎結合を検討
 - TXエンジンの種類は複数あっても、Epochベースはすべて共通なので
 - 下側のH/WもSSDからNVRAMもありうる

構成要素: OLAP

- 二方式を採用
- Spark
 - Asymmetrical HTAP
 - データ量がOLTP < OLAPになる
 - Tsurugiでcurrentデータの更新OLTP処理→SparkでOLAP処理
 - 少量のマスターはTsurugiで正規化・クレンジング・結合
 - 大量のTx・logデータはSparkで検索・結合



- Postgres/Tsurugi/その他
 - Symmetrical HTAP
 - OLTPとOLAPが同じデータ量
 - Postgres(Tsurugi)を二ノード準備して
 - ひとつがOLTPノードで、もう一つがOLAPノード
 - 別の商用OLAPがすでに検討開始
 - これは普通に商用なのでOSSではないと思います。



- 両者ともに「リアルタイムの橋渡し」で「一貫性を担保」する
 - 仕組み（フレームワーク）としては同じ

スケジュール感

- 実装自体のスケジュール感
 - 一応、個人的な非公式見解で、アカデミア関連はまた別
 - 2018-2020現在：概ね順調
 - コア部分は普通に着々と実装
 - アプリは一部プロト実装完了
 - 2020/11 NEDOの審査（ステージゲート）
 - これが通らない場合は一旦中止
 - 2021以降
 - 実装よりからインテグレーションにはいっていく
 - 2022にα版がとりあえず出る
 - というか「形はどうであれ一旦出す」
 - ユーザサイドではPoCとかいろいろ触ってみてもらう、という感じ
 - 間違ってもいきなり本番突撃とか止めてくださいね。

			2018	2019	2020	2021	2022
	Status						
アプリ1	BOM	詳細設計		基本設計	ベンチ実装	Tsurugi適用	
アプリ2	災害対策	1stプロト完成	基本設計	詳細設計・実装	プロト実装完了	アルゴリズム調整	Tsurugi適用
アプリ3	天文台	Sparkベース移植終了	基本設計	実装	プロト実装完了	アルゴリズム調整	Tsurugi適用
管理基盤		基本/詳細設計	技術検討	基本設計	詳細設計・実装	実装・インテグレート	インテグレート
SQL		実装中	基本設計	詳細設計・実装	詳細設計・実装	実装・インテグレート	インテグレート
Txエンジン		SILOベースほぼプロト完成	基本設計	詳細設計・実装	詳細設計・実装	実装・インテグレート	
Log		基本設計	—	—	基本設計	詳細設計・実装	実装・インテグレート
HATP連携		基本設計	技術検討	基本設計	基本設計	詳細設計・実装	インテグレート
OLAP：Spark			—	—	—	基本設計	実装・インテグレート
OLAP：RDB			—	—	—	基本設計	実装・インテグレート

お知らせ

- いろいろお手伝いをお願いできたら状態
 - 募集
 - Postgresでのパフォーマンス測定
 - Tsurugiの開発のためのベースパフォーマンスデータの収集
 - 有償です
 - アカデミアの外注先という扱いになるかと思えます
 - ご興味がある方は連絡ください
 - Kambayashi@nautilus-technologies.com