Empowered by Innovation NEC

ビッグデータ分析を高速化する分散処理技術を開発

日本電気株式会社

概要

NECは、ビッグデータの分析を高速化する分散処理技術を開発しました

本技術により、レコメンド・価格予測・需要予測などに必要な機械学習処理を従来の10倍以上高速に行い、分析結果の迅速な活用に貢献します

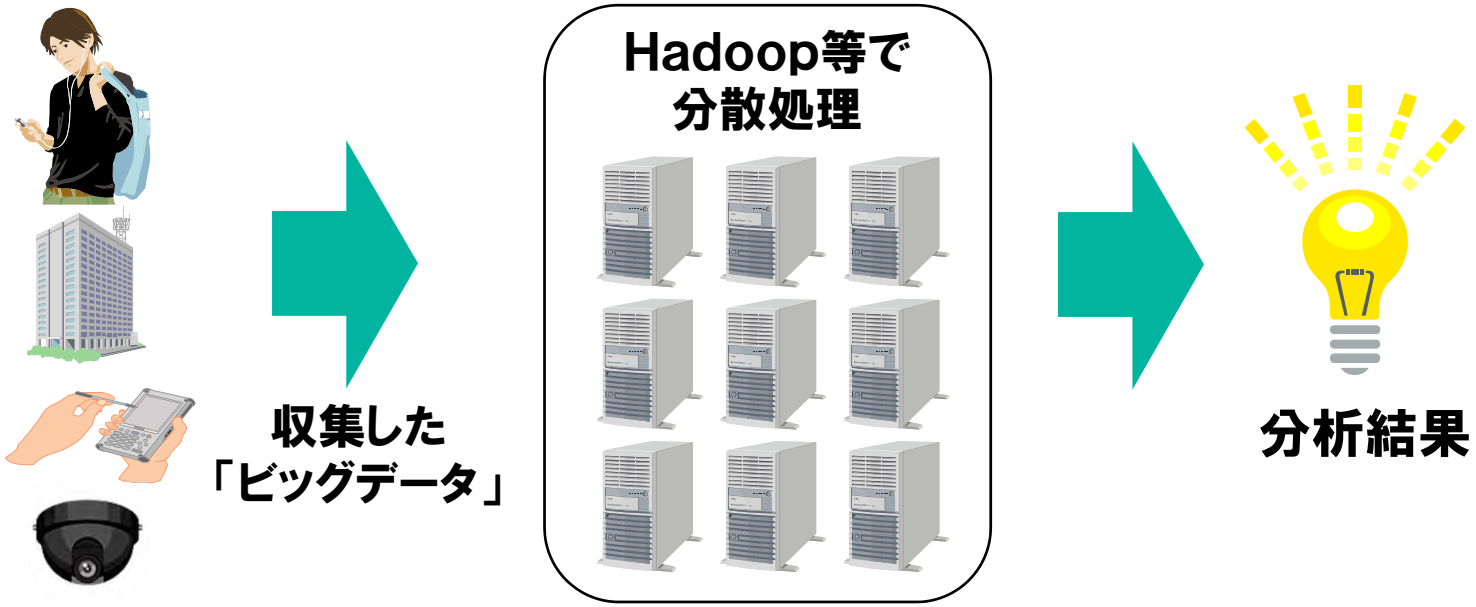
- ビッグデータの分散処理で一般的なオープンソース「Hadoop」を利用。

これにより、レコメンド・価格予測・需要予測などの分析において、従来提供まで1週間以上かかっていた最終的な分析結果を、翌日に提供できるようになります

- 長期間分析結果が得られないことでかかるコストや機会損失リスクを低減

背景

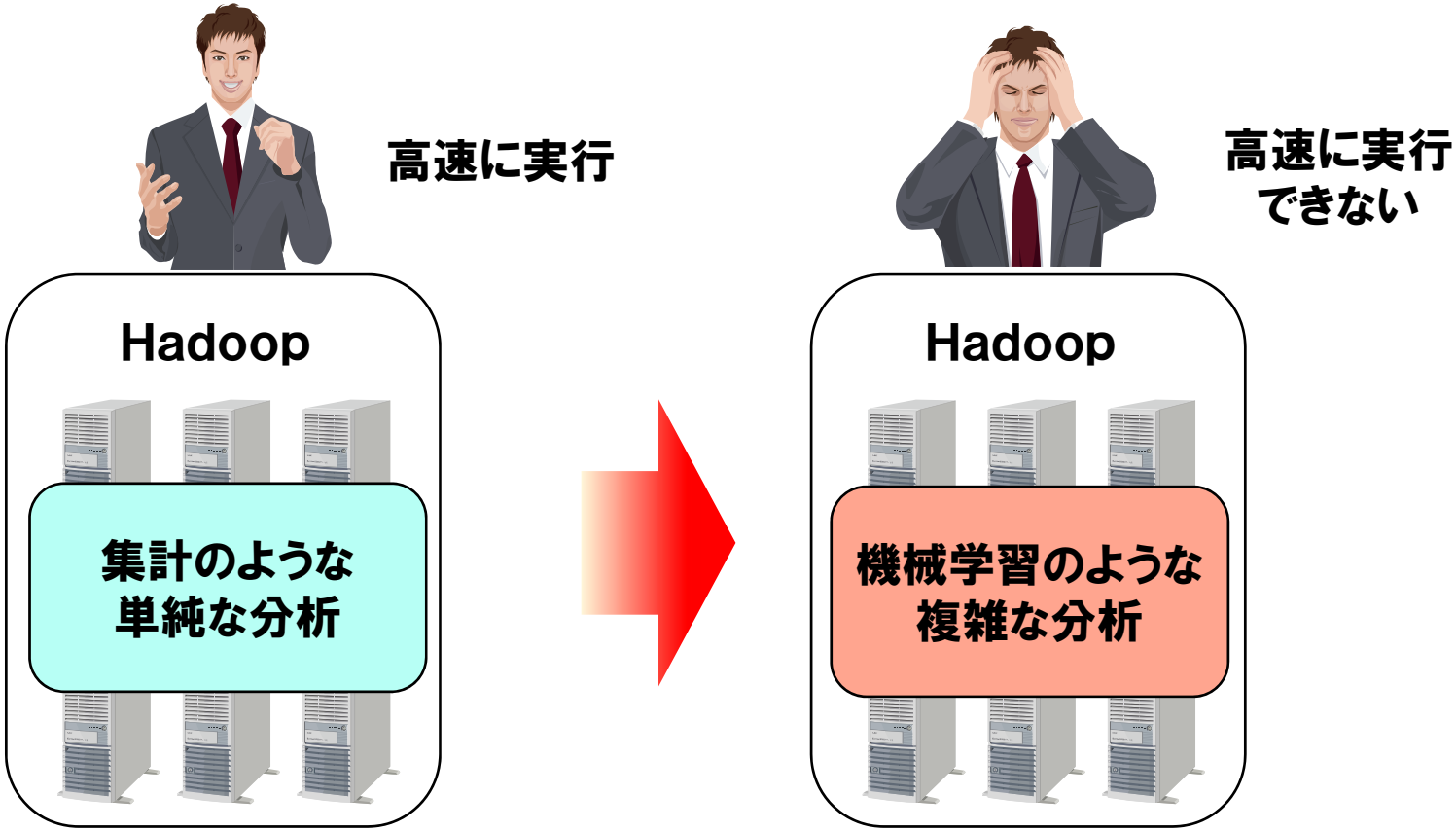
- 近年、インターネットやセンサなどから集まるビッグデータを分析することで、有用な情報を抽出し、ビジネスに活用するニーズが高まっている
- 現在、このようなビッグデータの分析は、Hadoop[*] 等の分散処理基盤を用い、多数のサーバで行うことが一般的



[*] Apache Software Foundationが開発・公開しているオープンソースの分散処理基盤

課題

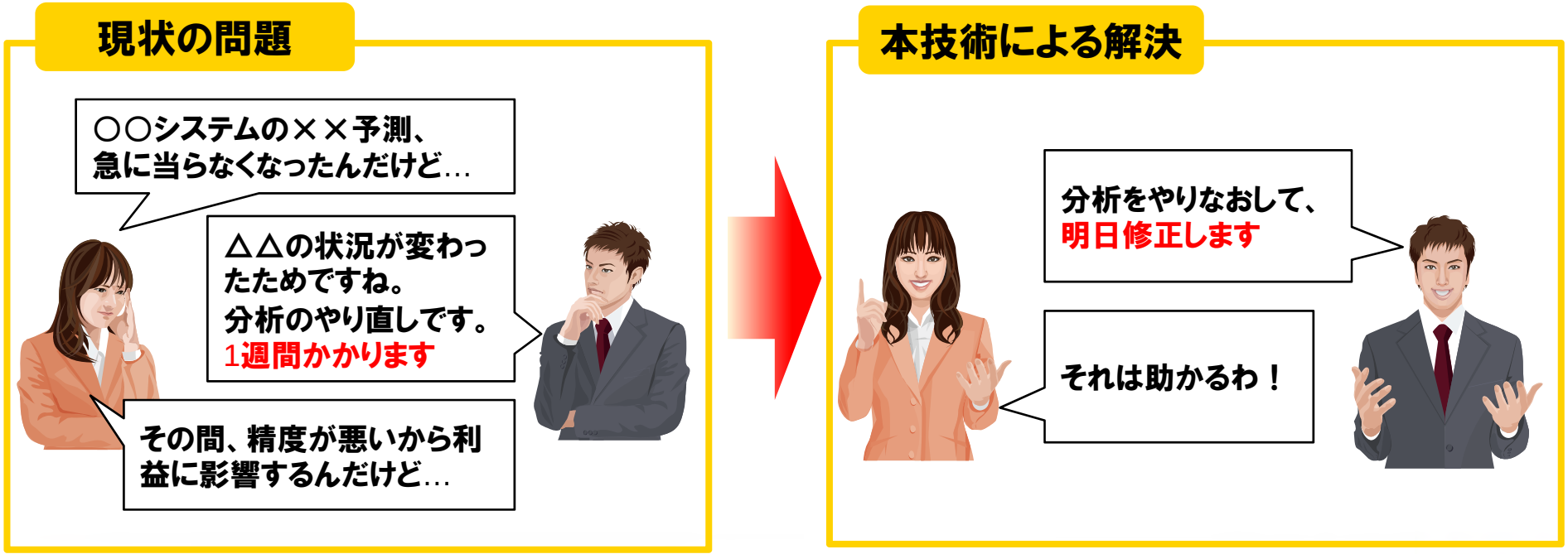
Hadoopは、集計のような単純な分析は高速に実行できるものの、
レコメンド・価格予測・需要予測などに用いる、機械学習[*] のような
複雑な分析は高速に実行することができない



[*]データから規則やパターン、知識を抽出し、現状認識や将来予測を行う技術

高速に実行できないことによる問題

- 分析を行う際は、処理結果を基に分析方法（パラメータの設定等）を修正するなどし、**分析処理を複数回行う。**
- 例えば、5回分析する場合、1回の処理に10時間かかると、**最終的な分析結果を得るまでに1週間以上かかる**
- **1時間で処理できれば、1日で分析結果を得ることが可能**



高速化技術の概要

- 機械学習で頻繁に用いられる繰り返し演算および行列演算を
  信頼性を損なわずに高速化

- 新技術を用いた新たな分散処理ソフトウェアを試作、大量データを用いた
  機械学習プログラムで検証、MapReduce部分を置き換えることで、
  従来のHadoopの10倍以上の速度を達成

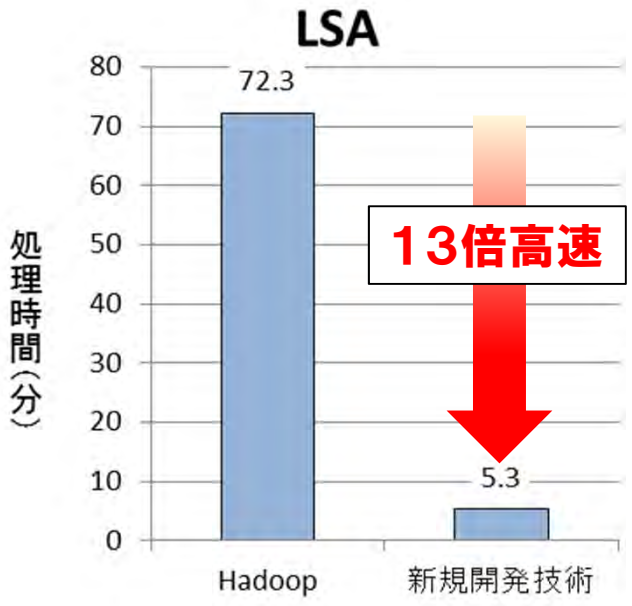
- これにより、最終的な分析結果を得るまで、1週間以上かかっていた[*]
  時間を1日に短縮し、分析結果の迅速な利用を可能に

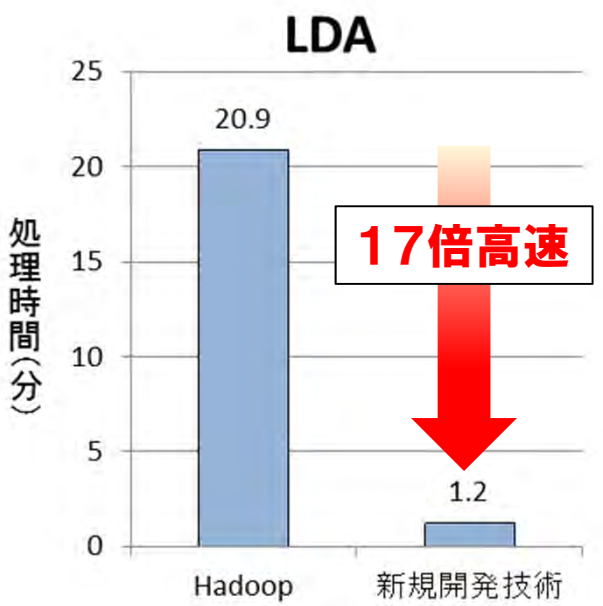
[*] 利用者数400万人、商品数50万点、購入履歴数2000万の購入履歴を用いて、
レコメンド処理を実行した場合

従来のHadoopと本技術の比較評価

2種類の機械学習プログラムで比較

- **いずれも大量の文書を入力として、類義語等関連の深い単語を抽出するもの**
  - LSA (Latent Semantic Analysis), LDA (Latent Dirichlet Allocation)





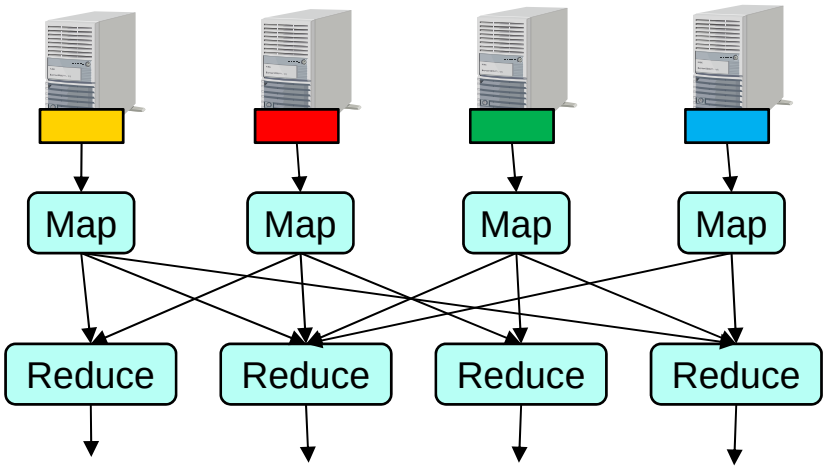
評価環境：

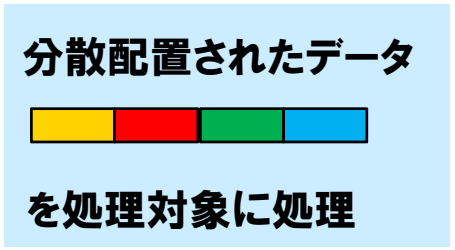
- 18台、72CPUのクラスタで評価
- 従来のHadoopはMahout（Hadoopを用いて記述された機械学習プログラム）による実装を利用
- 入力は英語版Wikipedia（文書数約400万、単語種類約50万、総単語数約2000万）。
  LSAは全文、LDAは従来のHadoopの実行速度が遅かったため1/30のデータで評価

技術の特長①
機械学習等の複雑な処理を高速化

[ご参考] MapReduceとは

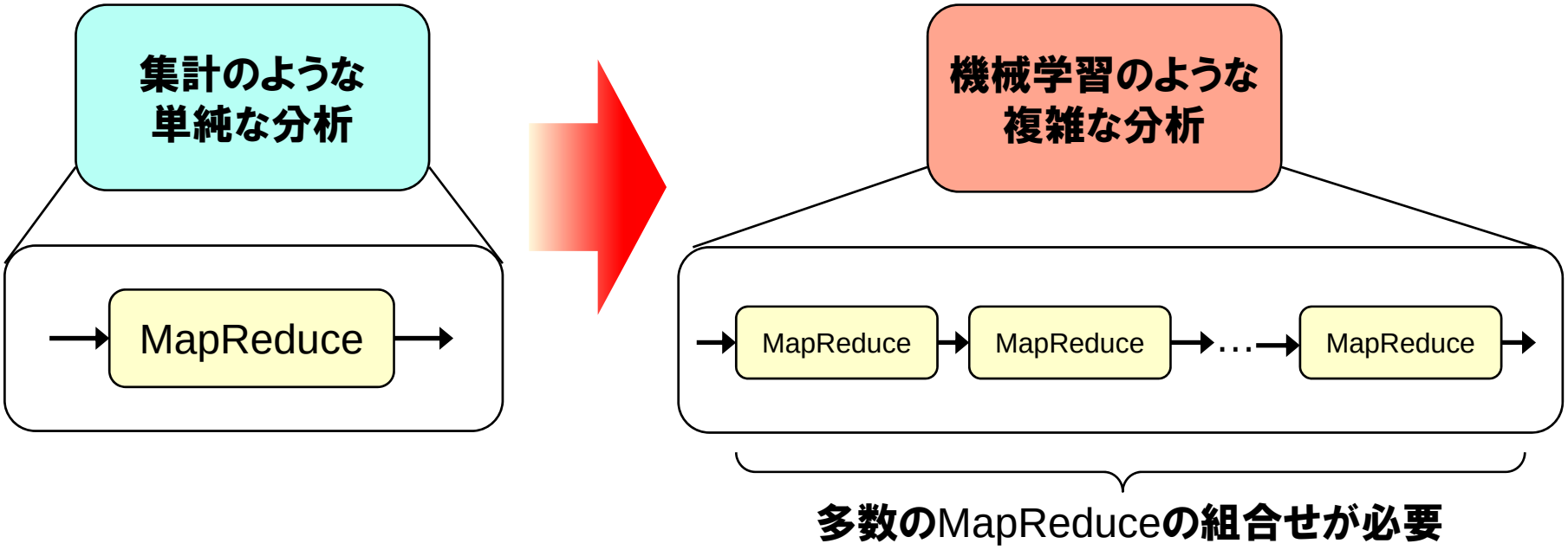
- Hadoopでは、「MapReduce」を単位として分散処理を実現
- 分散して処理を行うMap処理とその結果を集約するReduce処理から構成
  - Map処理の出力では、データの種類を「キー」として指定
  - Reduce処理には、同じ「キー」のデータが集められる
- プログラマはMap及びReduce関数を記述、システムが自動的に分散実行





従来のHadoopでの機械学習処理

- **集計のような単純な分析は、MapReduce単一で実現できるが、機械学習は繰り返し演算を必要とするため、これを実現するため多数のMapReduceを組み合わせる必要がある**

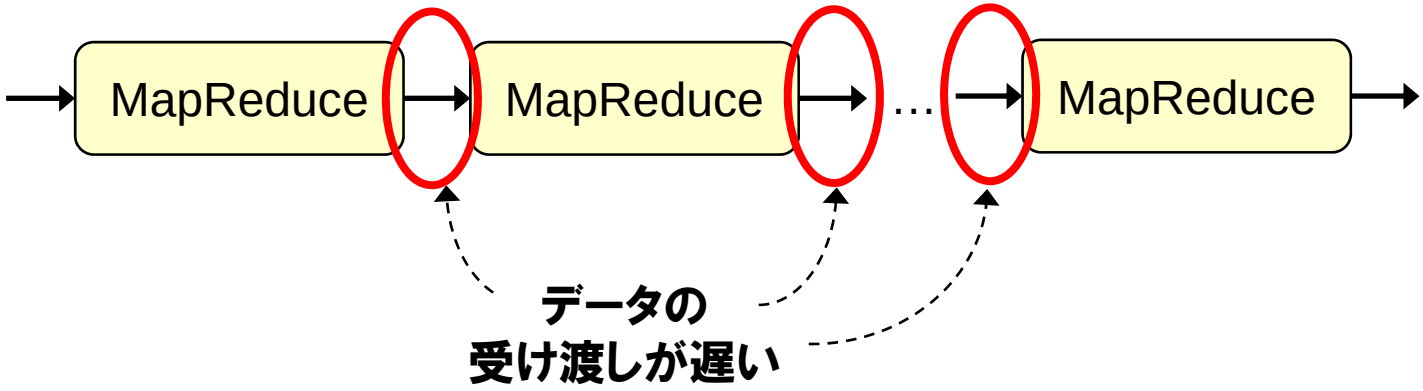


- **また、機械学習は行列演算も必要とするが、MapReduceは不得意**
  - **多数のMapReduceを組み合わせる必要があるとともに、処理を実行する際にサーバ間の通信が非効率になるケースがある**

多数のMapReduceの組合せで低速になる理由

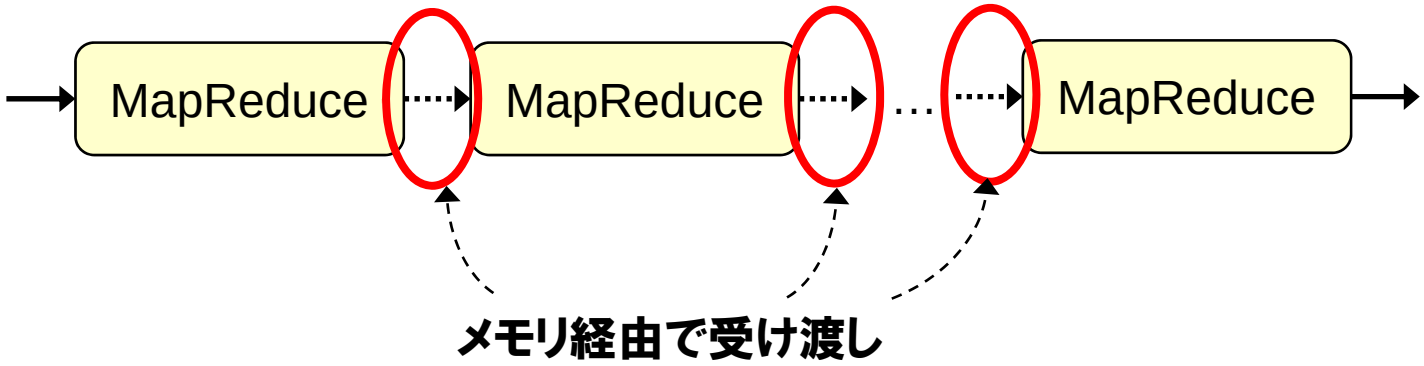
MapReduce間のデータ受け渡しが遅いため

- ハードディスク経由で多量のデータが受け渡される



高速化の内容

- **データの受け渡しをハードディスクではなく、メモリ経由とし、高速化**



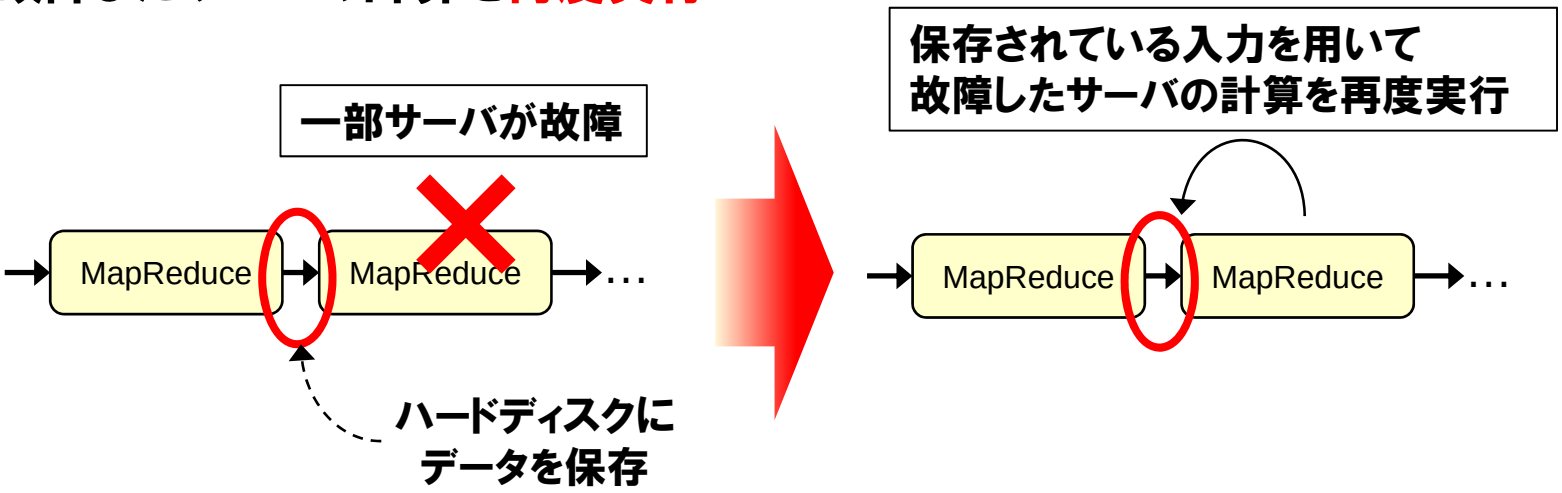
- **行列演算を得意とするMPI (Message Passing Interface) [*]を利用可能とし、行列演算を高速化**

[*]サーバ同士がメッセージを送りあうことで分散処理を行う手法

技術の特長②
高速化と高い信頼性を両立

従来のHadoopでの高信頼化

- 分散処理では多数のサーバを用いるため、１台が故障する確率が増大
  したがって、サーバが故障した場合でも処理を継続できる仕組みが必須
- 故障時の動作
  - 各MapReduceは、その入力がハードディスクに保存されていることを仮定
  - 一部サーバが故障したら、保存されている入力を用いて、
    故障したサーバの計算を再度実行

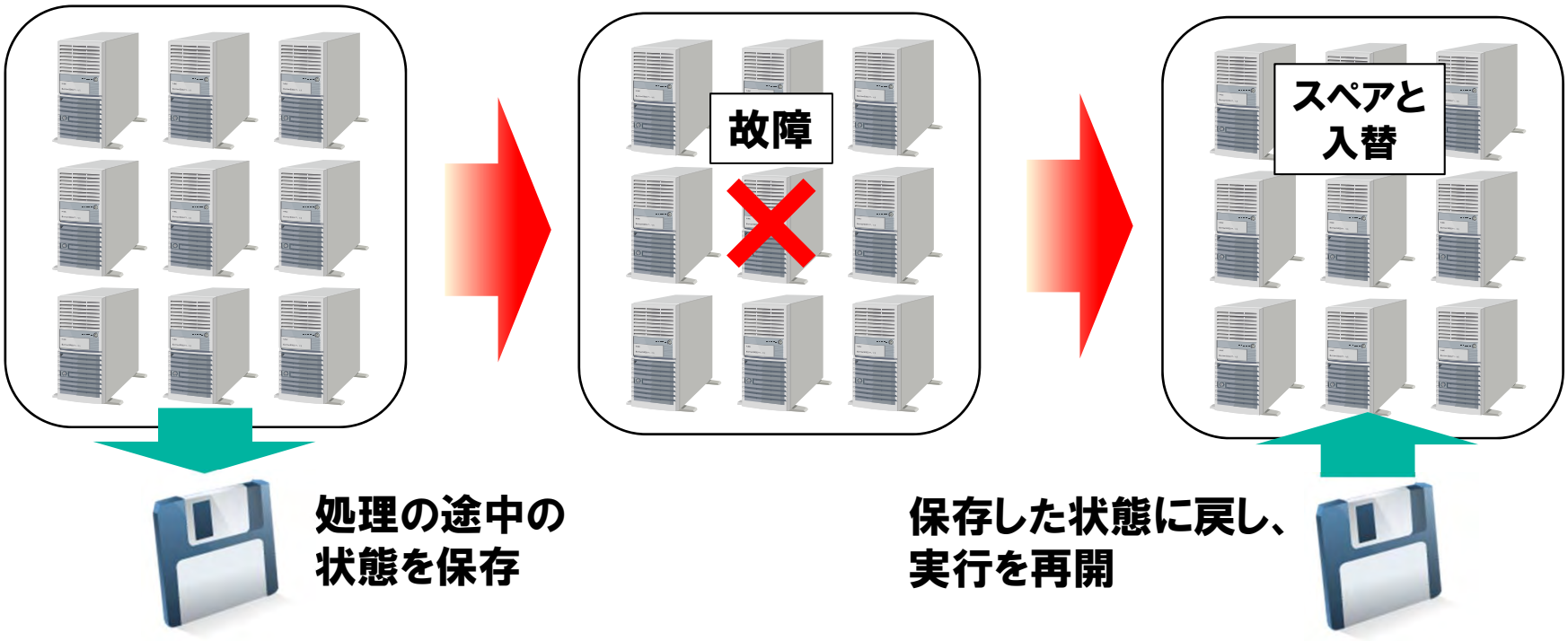


- 今回の高速化技術では、メモリ経由でデータを受け渡すため、この仕組みは
  使えない
  - サーバ故障時には失われてしまう

新たな高信頼化手法を開発

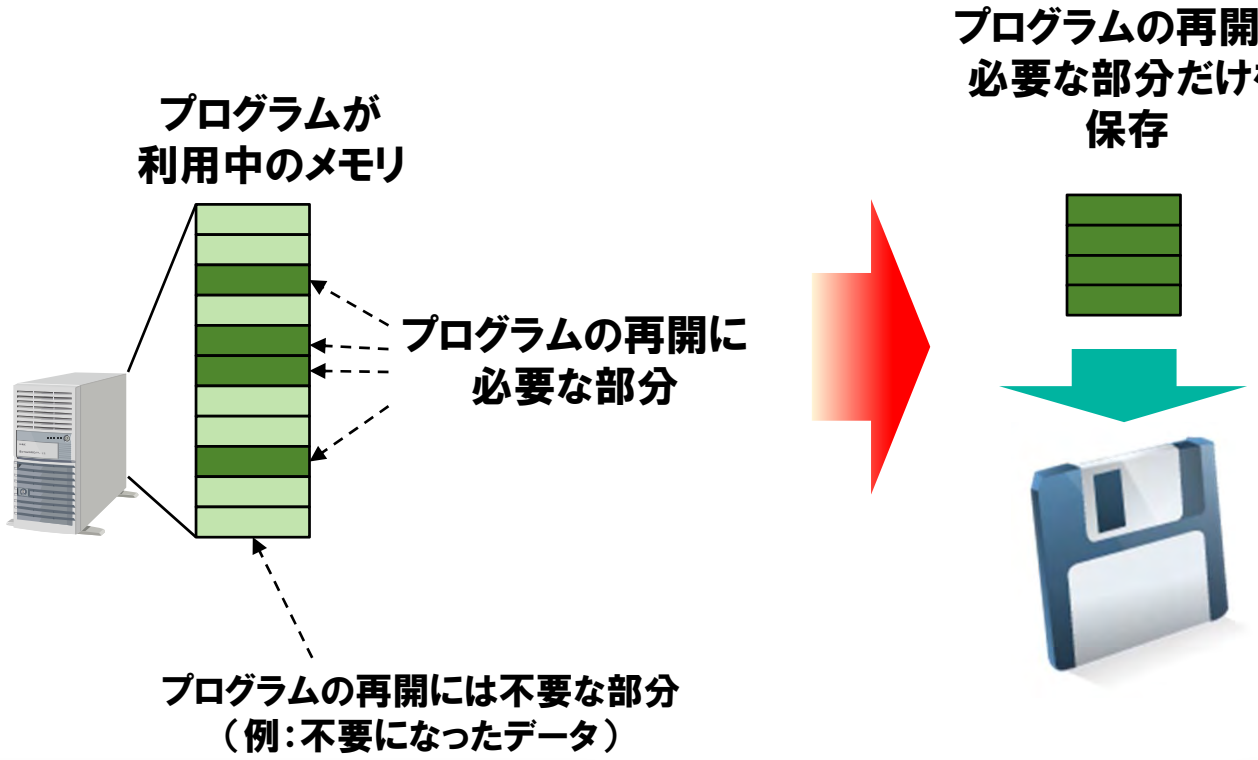
処理の途中の状態を高速に保存する手法を開発

- 適切な頻度で処理の途中の状態を保存
- サーバ故障時には、保存した状態から実行を再開、処理を継続



処理の途中の状態を高速に保存する手法

- 各サーバで動いているプログラムが利用しているメモリのうち、
  プログラムの再開に必要な部分だけを選択して保存する手法を
  世界で初めて実現
- これにより、**保存するデータサイズを大幅に削減**、高速な保存を実現



Empowered by Innovation