

MATLAB EXPO 2018

MATLABによる 大規模フリートデータ解析

アプリケーションエンジニアリング部
齊藤 甲次朗



アジェンダ

- はじめに
 - ビッグデータ解析の課題
 - MATLAB®を活用したフリートデータ解析事例
- フリートデータ解析実践
 - デスクトップでの解析
 - クラスタへのスケールアウト
- MATLAB解析のシステムへの統合
- まとめ



25 GB

/ 1hour

フリートデータ解析を含むビッグデータ解析の課題

1. ビッグデータのための**新しいツールを学ぶコスト**が掛かる
2. 大規模な計算に移行するために、プロトタイプで書いた**コードの書き直し**が必要になる
3. データ解析をシステム化するのに**時間と工数**が掛かる

MATLABを活用したフリートデータ解析事例

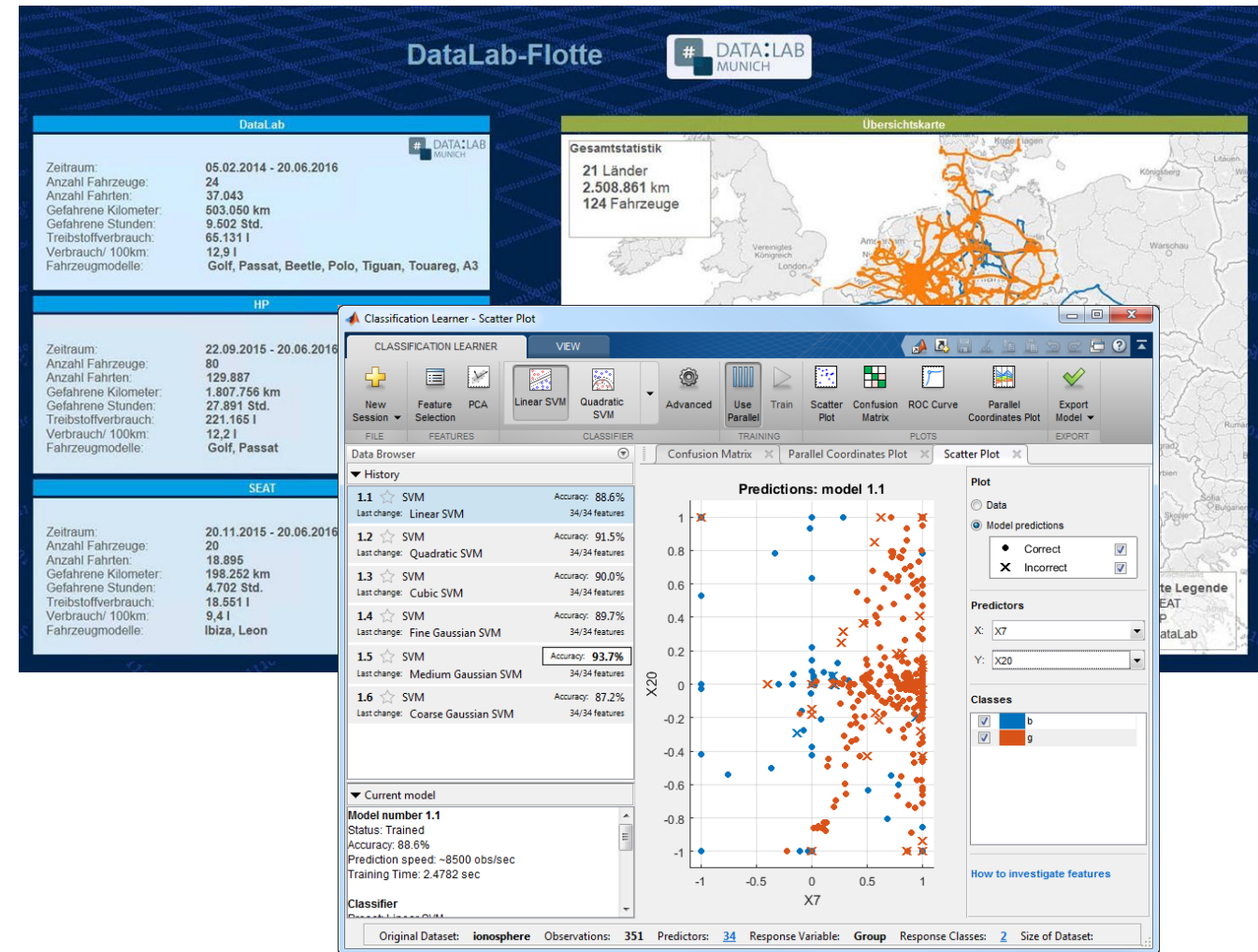
Volkswagen Data Lab

目的

- 運転行動から個々の運転者を特定し、運転者の安全性の向上や自動車保険などに役立つサービスを開発

課題

- 少ない学習データを使って多くの運転者を分類
- 環境状況に依らない解析の堅牢性
- 計算時間



“Connected Car – Fahrererkennung mit MATLAB”
 Julia Fumbarev, Volkswagen Data Lab
 (2017年6月MATLAB EXPO Germanyでの講演)

フリーストデータ解析実践

フリートデータ解析実践 使用するデータ

- 車にOBD Dongleを付け走行データを記録

トリップ数：**1300**以上

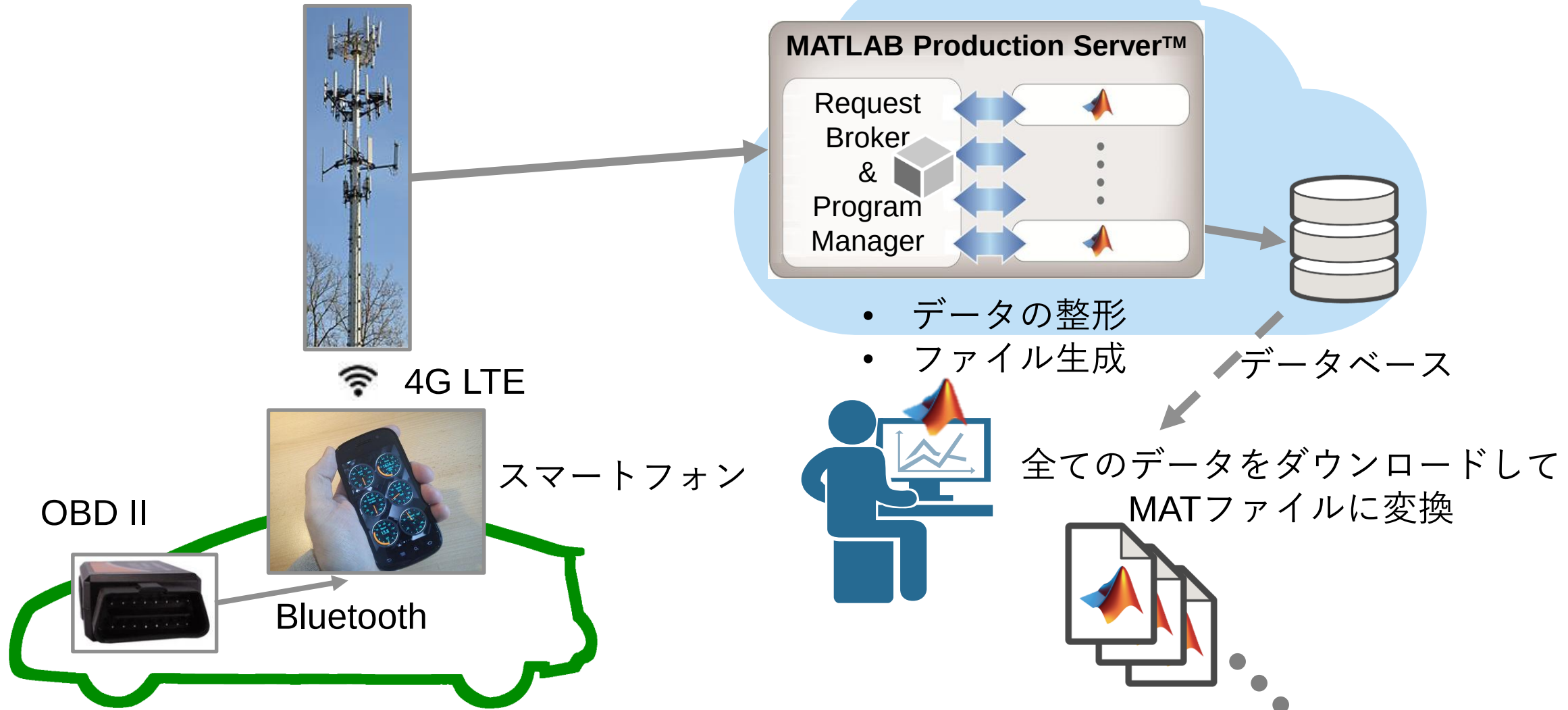
車両：**21**台

チャネル数：**39**

データ収集期間：約**1.5**年



フリートデータ解析実践 使用するデータ



フリートデータ解析のワークフロー

データへのアクセス

データの前処理

予測モデルの開発

システムへの統合

ファイル



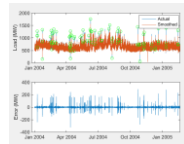
データベース



センサー



異常・欠損データの扱い



データ削減/
変換



特徴抽出



モデルの作成
(機械学習)



パラメータ最適化



モデルの検証



デスクトップ
アプリケーション



エンタープライズ
システム

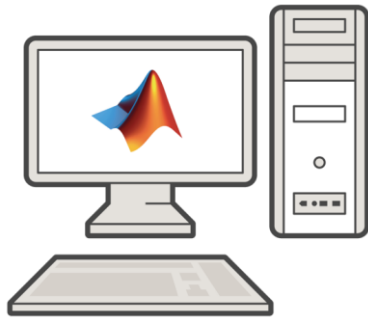
MATLAB Excel
.NET C/C++
.exe Java .dll

組込デバイスと
ハードウェア



ビッグデータの扱い フリーストデータ解析 サマリー

ステップ1



デスクトップPCでの解析

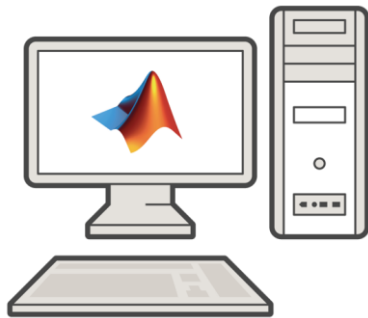
ステップ2



Hadoop® クラスタでの解析

フリートデータ解析実践 デスクトップでの解析

ステップ1



フリートデータ解析アルゴリズムを検討するために、
まずはデスクトップで**試行錯誤**

今後のクラスターへの**スケールアウトを意識**してコードを書く

フリートデータ解析実践

データへのアクセス

生データを見てみる



1ファイル

現在のフォルダー

名前
55a41c8969702d115b0562e1
55a41ce969702d115b0606ea
55a41d5b69702d115b06cb3
55b7606569702d29a300000
55a41dee69702d115b07d014

MATLAB上でファイルを
ダブルクリック

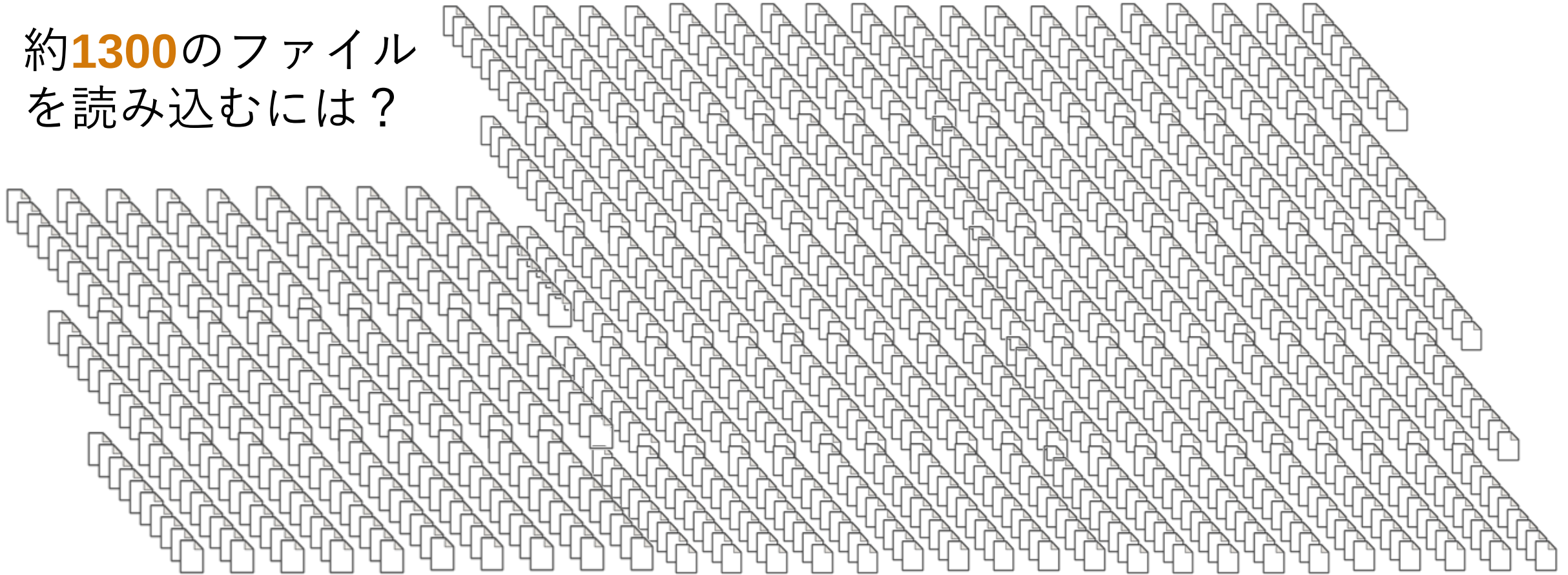
フィールド	値
x_id	13077x1 cell
created_at	13077x1 cell
k10	13077x1 cell
k11	13077x1 cell
k1f	13077x1 cell
k33	13077x1 cell
k44	13077x1 cell
k45	13077x1 cell
k47	13077x1 cell
k5	13077x1 cell
kc	13077x1 cell
kf	13077x1 cell
kfe1805	13077x1 cell
kff1001	13077x1 cell
kff1005	13077x1 double
kff1006	13077x1 double
kff1007	13077x1 double

tripData.kff1005	
	1
1	-83.6308783100000000
2	-83.6308613100000000
3	-83.6308384500000000
4	-83.6308085400000000
5	-83.6307792500000000

経度

フリートデータ解析実践 データへのアクセス

約**1300**のファイル
を読み込むには？



フリートデータ解析実践 クイズ

解1

dirコマンドとfor文で繰り返し処理

ファイル名の取得

```
fileNames = dir(fullfile('..', 'LogFiles', '*.mat'));
```

ファイル分だけ読み取り

```
for ii=1:length(fileNames)
    fileName = fileNames(ii).name;
    load(fileName)

    % 何らかの処理
end
```

解2

dirコマンドと並列for文で繰り返し処理

ファイル名の取得

```
fileNames = dir(fullfile('..', 'LogFiles', '*.mat'));
```

ファイル分だけ読み取り

```
parfor ii=1:length(fileNames)
    fileName = fileNames(ii).name;
    tripData = load(fileName);

    % 何らかの処理
end
```

解3

それ以外

どれが一番良いアプローチでしょうか？

フリートデータ解析実践

データへのアクセス

datastore: データ、ファイルの集合体を読み取るオブジェクト
特に**機械学習**や**ディープラーニング**で使用

対象データ	データストアの種類
表形式のテキストファイル (CSVなど)	TabularTextDatastore
Excel®形式のスプレッドシート (XLSXなど)	SpreadsheetDatastore
画像	ImageDatastore
リレーショナルデータベースのデータ	DatabaseDatastore
カスタム形式のファイル	FileDatastore
MDF形式のファイル	mdfDatastore

など

https://jp.mathworks.com/help/matlab/import_export/what-is-a-datastore.html

フリートデータ解析実践

データへのアクセス

トリップIDとVIN(車両識別番号)リストを読み取り

```
tripTable = readtable('tripTable.csv');
```

ワイルドカードで
指定可能

個々のトリップファイルの読み取り

```
dataDir = fullfile('..', 'LogFiles', '*.mat');
```

fileDatastoreを使ったカスタム読み込み関数での読み取り

```
readFcn = @(filename) readTrip(filename, tripTable);
fds = fileDatastore(dataDir, 'ReadFcn', readFcn, 'UniformRead', true);
disp(fds)
```

FileDatastore のプロパティ:

datastoreの作成

```
Files: {
    ...\FleetDataAnalytics\LogFiles\55a3fd0069702d5
    ...\FleetDataAnalytics\LogFiles\55a3fd0169702d5
    ...\FleetDataAnalytics\LogFiles\55a3fe3569702d5
    ... and 1371 more
}
UniformRead: 1
ReadFcn: @(filename)readTrip(filename,tripTable)
AlternateFileSystemRoots: {}
```

カスタムの読込関数

```
function tOut = readTrip(fpath,tripTable)
%% 個々のトリップファイルを読み取り、整形する読込関数
% Copyright 2018 The MathWorks, Inc.

% [入力]
% fpath: 生データファイルへのパス
% tripTable: トリップIDとVINの関係が記載されているテーブル

% [出力]
% tOut: 整形されたタイムテーブル

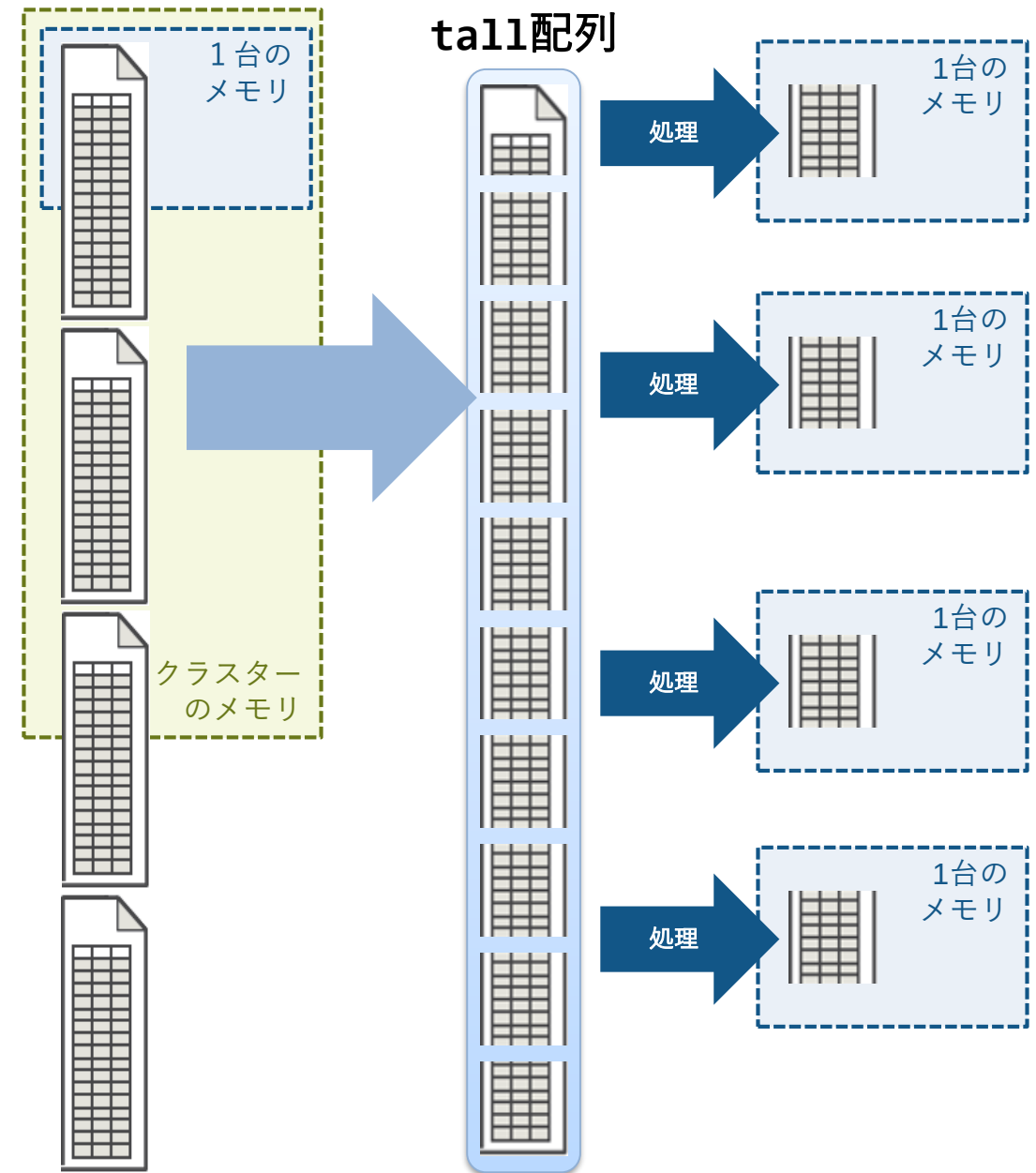
%% ファイルを読み込み生データを抽出する
% .matファイルメモリーに読み込む
data = load(fpath);

% ファイルから構造体データを取り出す
data = struct2table(data.tripData,'AsArray',false);
```

フリートデータ解析実践

データへのアクセス **ta11**

- メモリに収まる小さな塊にデータを自動的に分割
- データアクセスを最適化して実行
- 並列演算もサポート



フリートデータ解析実践

データへのアクセス **tall**

tall配列の作成

```
tt = tall(fds);
```

'local' プロファイルを使用して並列プール (parpool) を起動中...
2 ワーカーに接続されます。

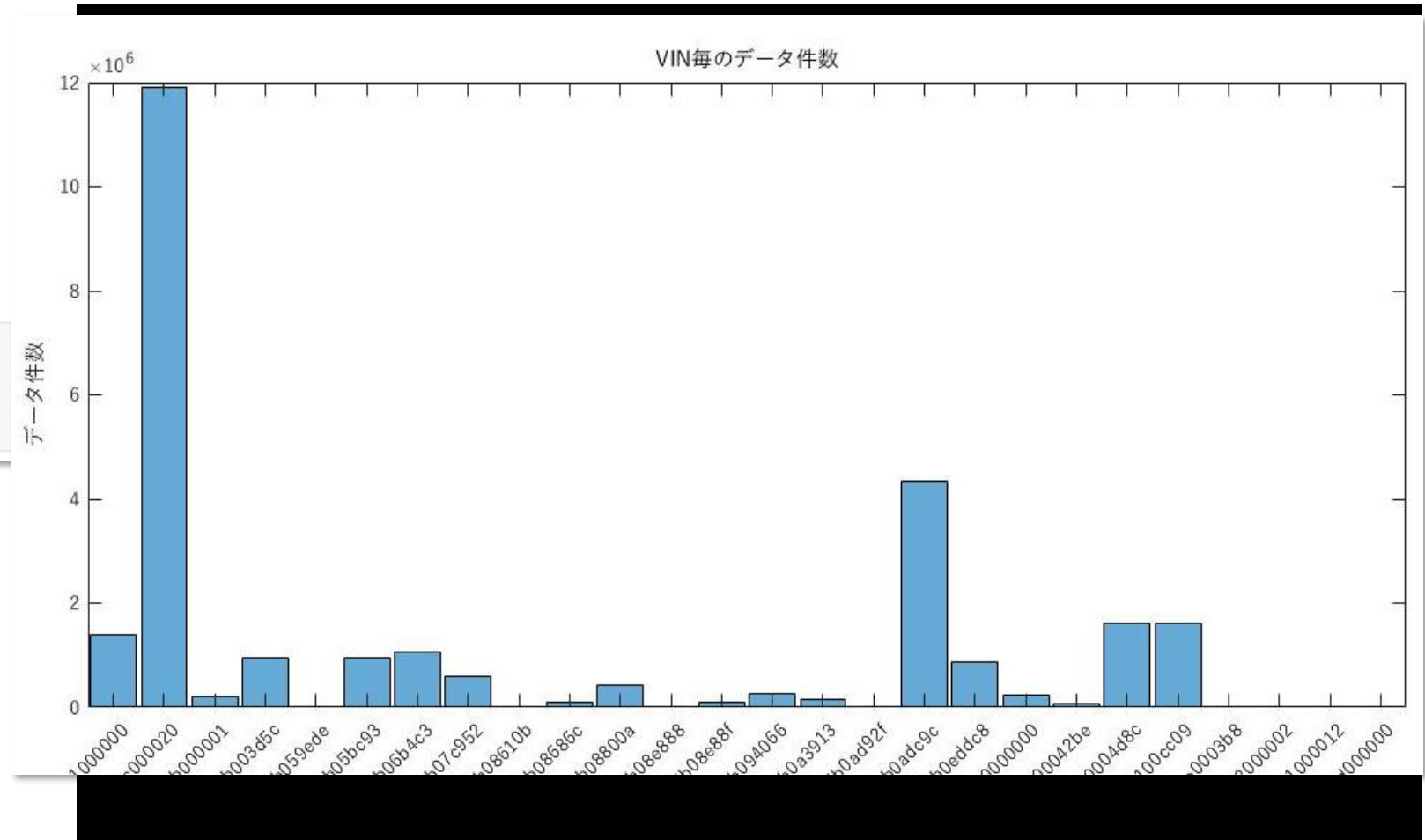
関連ツール	tallでできること	
MATLAB	tall処理	} ステップ1
+ Parallel Computing Toolbox	ローカルマシンでの並列tall処理	
+ MATLAB Distributed Computing Server™	クラスターでの並列tall処理	} ステップ2
+ Apache™ Hadoop / Apache Spark™ (サードパーティ)	Hadoop/Sparkクラスター上での 並列tall処理	

フリートデータ解析実践 ビッグデータの可視化

データ全てを使って可視化
histogram

tall配列の可視化

```
%% どの車両が多いのかプロット  
histogram(tt.VIN)
```



tall配列の可視化

https://www.mathworks.com/help/matlab/import_export/tall-data-visualization.html

フリートデータ解析実践

tall配列の保存

- ビッグデータ処理において、生データの読み込みを繰り返したくない



tall配列の保存

```
write('PreProcessedFiles', tt);
```

- ローカルフォルダー
 - Hadoop® Distributed File System (HDFS™)
 - Amazon S3™
 - Azure® Storage Blob
- に保存

MATLAB_EXPO2018 > FleetDataAnalytics > PreProcessedFiles			
Name	Date modified	Type	Size
array_r1_00001_snapshot_0x28.mat	2018/09/06 15:32	MATLAB Data	27,166 KB
array_r1_00002_snapshot_0x50.mat	2018/09/06 15:32	MATLAB Data	52,644 KB
array_r1_00003_snapshot_0x39.mat	2018/09/06 15:32	MATLAB Data	37,573 KB
array_r1_00004_snapshot_0x38.mat	2018/09/06 15:32	MATLAB Data	36,746 KB
array_r1_00005_snapshot_0x38.mat	2018/09/06 15:33	MATLAB Data	36,830 KB
array_r1_00006_snapshot_0x35.mat	2018/09/06 15:33	MATLAB Data	35,123 KB
array_r2_00001_snapshot_0x37.mat	2018/09/06 15:32	MATLAB Data	36,230 KB
array_r2_00002_snapshot_0x36.mat	2018/09/06 15:32	MATLAB Data	35,785 KB
array_r2_00003_snapshot_0x38.mat	2018/09/06 15:32	MATLAB Data	38,398 KB
array_r2_00004_snapshot_0x31.mat	2018/09/06 15:32	MATLAB Data	34,860 KB
array_r2_00005_snapshot_0x3D.mat	2018/09/06 15:33	MATLAB Data	41,288 KB
array_r2_00006_snapshot_0x34.mat	2018/09/06 15:33	MATLAB Data	32,265 KB
array_r3_00001_snapshot_0x86.mat	2018/09/06 15:33	MATLAB Data	81,237 KB
array_r3_00002_snapshot_0x44.mat	2018/09/06 15:33	MATLAB Data	43,832 KB
array_r3_00003_snapshot_0x29.mat	2018/09/06 15:34	MATLAB Data	26,837 KB
array_r4_00001_snapshot_0x40.mat	2018/09/06 15:33	MATLAB Data	42,059 KB
array_r4_00002_snapshot_0x47.mat	2018/09/06 15:33	MATLAB Data	45,196 KB
array_r4_00003_snapshot_0x3F.mat	2018/09/06 15:34	MATLAB Data	41,538 KB
array_r4_00004_snapshot_0x32.mat	2018/09/06 15:34	MATLAB Data	32,084 KB
array_r4_00005_snapshot_0x2C.mat	2018/09/06 15:34	MATLAB Data	27,931 KB

フリートデータ解析実践

保存したtall配列の読み込み

保存したtallデータの読み込み

```
fpath = fullfile('..', 'PreProcessedFiles', '*.mat');
ds = datastore(fpath, 'Type', 'tall');
ds.ReadSize = 100000;
tt = tall(ds);
% 時刻の表示を英語表記に変更
tt.time.Format = 'default';
```

TallDatastoreとして
読み込み

tt x

Mx4 tall timetable (unevaluated)

val =

Mx4 tall timetable

time	trip_id	VIN	ChannelName	ChannelValue
2015/07/13 18:01:36	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
2015/07/13 18:01:36	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
2015/07/13 18:01:37	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
2015/07/13 18:01:38	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
2015/07/13 18:01:39	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
2015/07/13 18:01:40	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
2015/07/13 18:01:41	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
2015/07/13 18:01:42	55a3fd0169702d5b41000005	55a3fd0069702d5b41000000	defaultUnit05	NaN
:	:	:	:	:
:	:	:	:	:

フリートデータ解析実践 ビッグデータの可視化

データの散らばりを見たい

scatter

地理空間的な分布をプロット

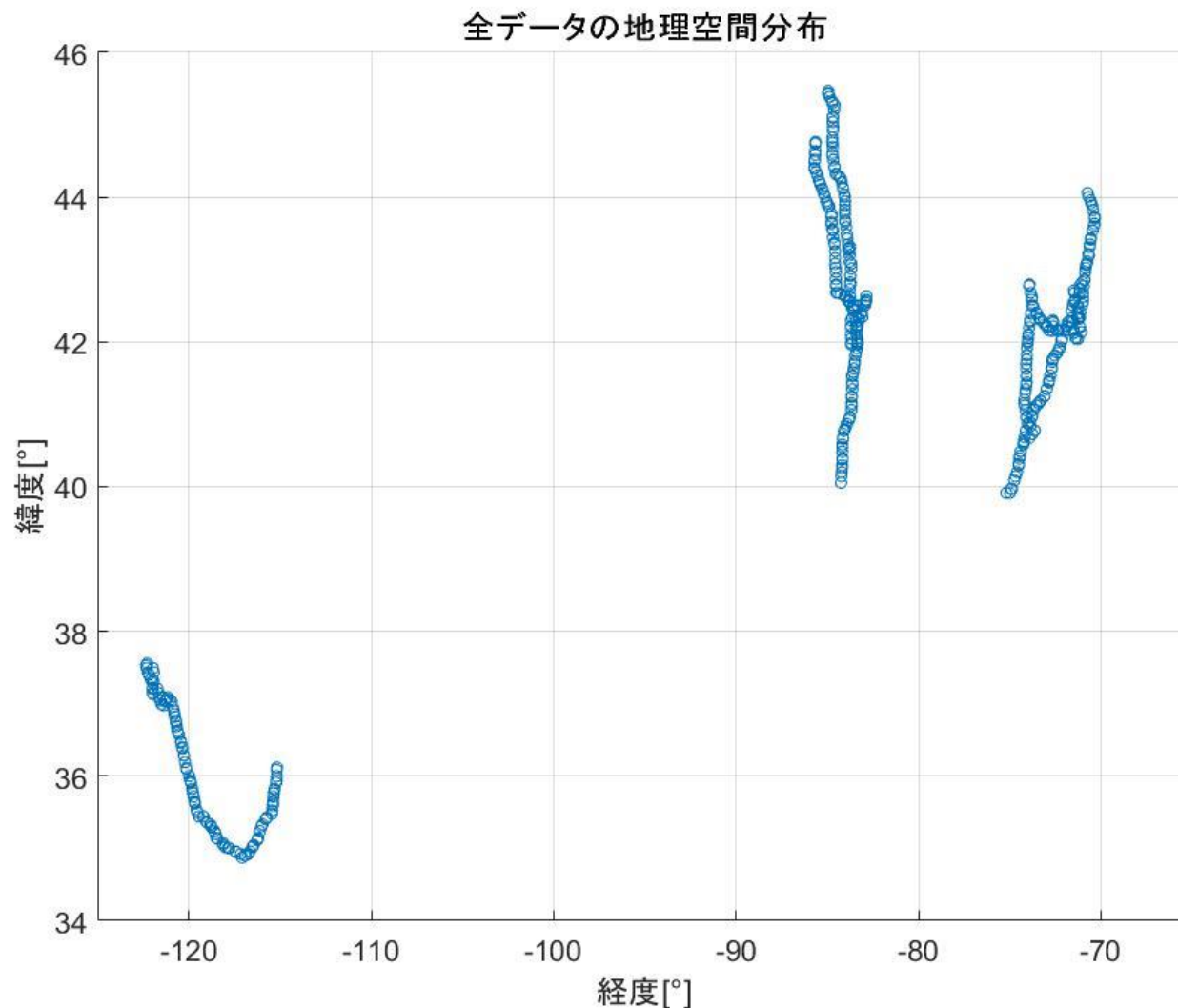
緯度経度だけ取り出し

```
latChannel = 'kff1006';  
lonChannel = 'kff1005';  
mskLat = ismember(tt.ChannelName, latChannel);  
mskLon = ismember(tt.ChannelName, lonChannel);  
ttLat = tt(mskLat, :);  
ttLon = tt(mskLon, :);
```

メンバーを抽出

scatterプロット

```
scatter(ttLon.ChannelValue, ttLat.ChannelValue)  
grid on  
xlim([-125, -65])  
ylim([34, 46])  
xlabel('経度[°]', 'FontSize', 16)  
ylabel('緯度[°]', 'FontSize', 16)  
title('全データの地理空間分布');  
set(gca, 'FontSize', 16);
```



フリートデータ解析実践 ビッグデータの可視化

地図上にプロットするには
メモリに取り込んでから
geoscatter R2018b

gatherを実行してメモリに取り込み

```
[ttLatGathered, ttLonGathered] = gather(ttLat, ttLon);
```

geoscatterプロット

tall配列をメモリに取り込み

```
gs = geoscatter(ttLatGathered.ChannelValue, ttLonGathered.ChannelValue);
gax = gca;
gax.FontSize = 16;
gax.Title.String = '全データの地理空間分布';
```

ベースマップの変更

```
name = 'openstreetmap';
url = 'a.tile.openstreetmap.org';
copyright = char(uint8(169));
attribution = copyright + "OpenStreetMap contributors";
displayName = 'Open Street Map';
addCustomBasemap(name,url,'Attribution',attribution,'DisplayName',displayName)
geobasemap('openstreetmap')
```



フリートデータ解析実践 ビッグデータの可視化

全トリップのトリップ時間を調べる

トリップIDでグループ分け

```
fileIdx = findgroups(ttSet.trip_id);
```

それぞれのグループをミニテーブルのcell配列に分割する

```
ttSet = splitapply(@(g) {g}, table(ttSet), fileIdx);
```

タイムスタンプの確認

時刻でソート

```
ttSet = cellfun(@(x) sortrows(x, 'time', 'ascend'), ttSet, 'UniformOutput', false);
```

トリップの所要時刻を求める

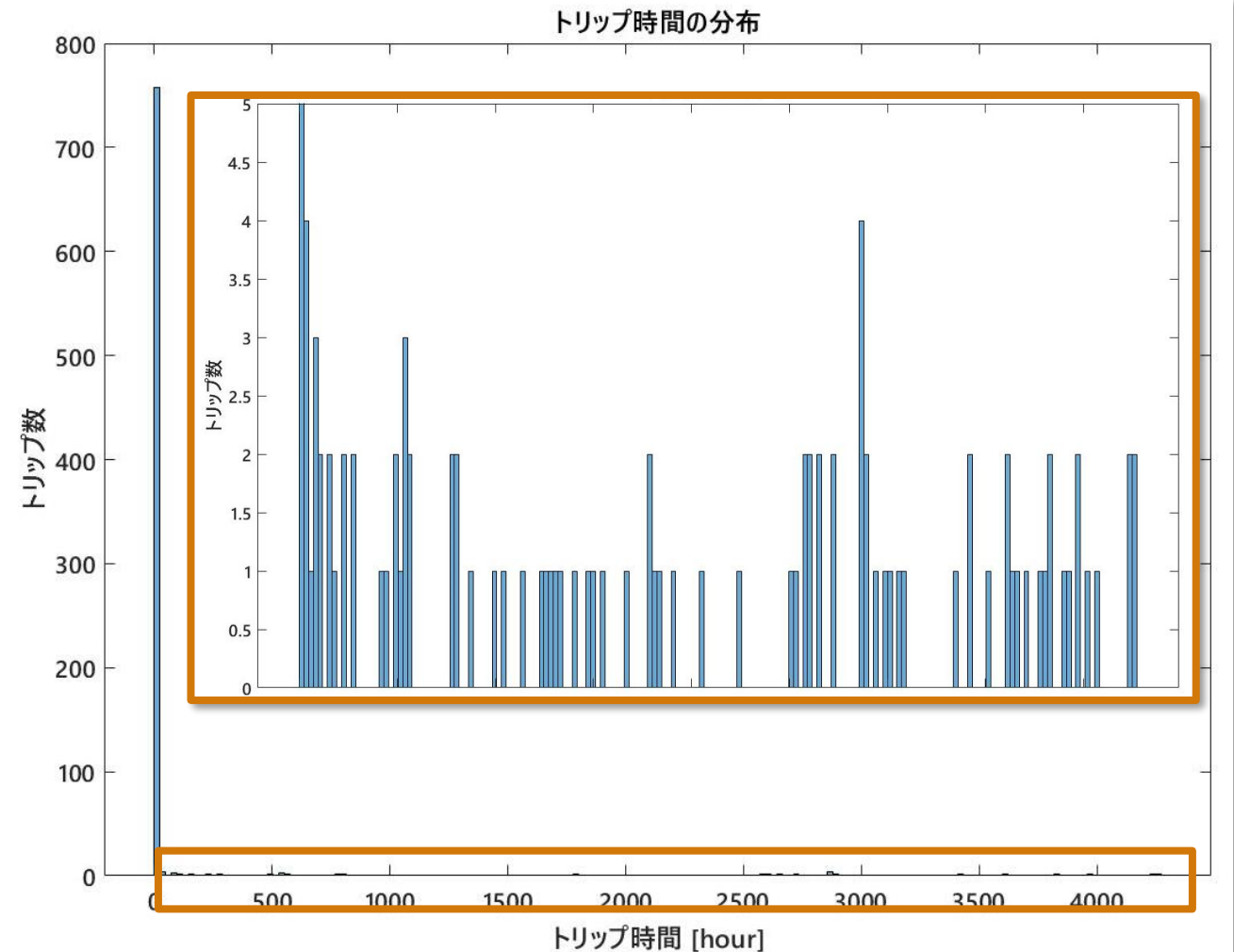
```
ttTripDuration = cellfun(@(x) seconds(x.time(end)-x.time(1)), ttSet, 'UniformOutput', false);
```

hourに変換

```
numOfHours = cell2mat(ttTripDuration) / 60 / 60;
```

histogramのプロット

```
fig = figure;
ax = axes('Parent', fig);
histogram(ax, numOfHours, 'BinWidth', 24)
title(ax, 'トリップ時間の分布')
xlabel(ax, 'トリップ時間 [hour]', 'FontName', 'Yu Gothic UI Semibold');
ylabel(ax, 'トリップ数', 'FontName', 'Yu Gothic UI Semibold');
box(ax, 'on');
set(ax, 'FontName', 'Yu Gothic UI Semibold', 'FontSize', 16);
```



フリートデータ解析実践

ビッグデータの可視化

不自然なトリップ時間を詳しく見てみる

```
numOfHoursGathered = gather(numOfHours);
```

最大のトリップ時間のデータを抽出

```
[maxNumOfHours, idx] = max(numOfHoursGathered);
ttOfMaxTripHours = ttSet(idx, :);
ttOfMaxTripHours = gather(ttOfMaxTripHours);
```

ttOfMaxTripHours{1, 1}

		1	2	3	4
	time	trip_id	VIN	ChannelName	ChannelValue
3432	2015/01/15 22:44:21	55a41a...	55a3...	kff1006	42.474901140...
3433	2015/01/15 22:44:22	55a41a...	55a3...	kff1005	-83.62906784...
3434	2015/01/15 22:44:22	55a41a...	55a3...	kff1006	42.474908820...
3435	2015/01/15 22:44:23	55a41a...	55a3...	kff1005	-83.629061190...
3436	2015/01/15 22:44:23	55a41a...	55a3...	kff1006	42.474917500...
3437	2015/07/12 17:49:20	55a41a...	55a3...	kff1005	-84.985817060...
3438	2015/07/12 17:49:20	55a41a...	55a3...	kff1006	45.435388420...
3439	2015/07/12 17:49:21	55a41a...	55a3...	kff1005	-84.98563572...
3440	2015/07/12 17:49:21	55a41a...	55a3...	kff1006	45.435394190...
3441	2015/07/12 19:00:38	55a41a...	55a3...	kff1005	-84.68864308...
3442	2015/07/12 19:00:38	55a41a...	55a3...	kff1006	44.878871850...

半年ほど間隔が空いている

フリートデータ解析実践 ビッグデータの前処理

■ 前処理

タイムスタンプ間隔が24hourより大きいデータを消去

```
thd = hours(24);  
ttSet = RejectSpuriousTimestamps(ttSet, thd);
```

```
function cellOut = RejectSpuriousTimestamps(cellIn,thd)  
%% データ内の不自然なタイムスタンプを消去する  
% Copyright 2018 The MathWorks, Inc.  
  
%% トリップデータのタイムスタンプをソート  
cellIn = cellfun(@(x) sortrows(x,'time','ascend'), cellIn,'UniformOutput', false);  
  
% 不自然なタイムスタンプの行を消去する関数ハンドルの定義  
removeSpuriousTimeFcn = @(T)DeleteRows(T,thd);  
  
% 不自然なタイムスタンプ行の消去  
cellOut = cellfun(removeSpuriousTimeFcn, cellIn, 'UniformOutput', false);  
end  
  
%% タイムスタンプの行を消去する関数  
function out = DeleteRows(in,thd)  
dt = vertcat(0,diff(in.time));  
msk = dt > thd;  
  
if sum(msk) == 0  
    out = in;  
else  
    idx = find(msk,1);  
    % 行の消去 (事前に時刻でソートしておくこと)  
    in(idx:end,:) = [];  
    out = in;  
end  
end
```

フリートデータ解析実践

ビッグデータの前処理

- 時刻が異なるデータを同期 (**synchronize**)

1734x4 timetable

	time	1 trip_id	2 VIN	k
1	2015/03/09 11:41:22	55a41...	55a3fe3569702d5c5c000020	-8
2	2015/03/09 11:41:23	55a41...	55a3fe3569702d5c5c000020	-8
3	2015/03/09 11:41:24	55a41...	55a3fe3569702d5c5c000020	-8
4	2015/03/09 11:41:25	55a41...	55a3fe3569702d5c5c000020	-8
5	2015/03/09 11:41:26	55a41...	55a3fe3569702d5c5c000020	-8

1087x4 timetable

	time	1 trip_id	2 VIN	k
22	2015/03/09 11:41:21	55a41f...	55a41f8069702d115b0adc9c	-7
23	2015/03/09 11:41:22	55a41f...	55a41f8069702d115b0adc9c	-7
24	2015/03/09 11:41:23	55a41f...	55a41f8069702d115b0adc9c	-7
25	2015/03/09 11:41:24	55a41f...	55a41f8069702d115b0adc9c	-7
26	2015/03/09 11:41:25	55a41f...	55a41f8069702d115b0adc9c	-7
27	2015/03/09 11:41:26	55a41f...	55a41f8069702d115b0adc9c	-7

synchronizedTbl

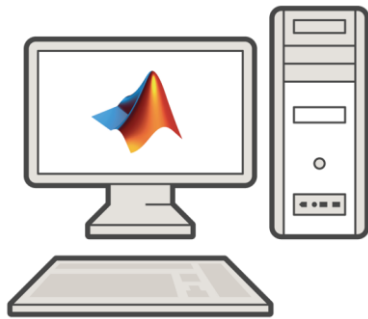
1952x8 timetable

	time	1 trip_id_tripDataExt	2 VIN_tripDataExtra	3 kff1005_tripDataExt	4 kff1006_tripDataExt	5 trip_id_tripDataExt	6 VIN_tripDataExtra	7 kff1005_tripDataExt	8 kff1006_tripDataExt
19	2015/03/09 11:41:18	<undefined>	<undefined>	NaN	NaN	55a41fac69702d...	55a41f8069702d...	-71.2241032800...	42.301...
20	2015/03/09 11:41:19	<undefined>	<undefined>	NaN	NaN	55a41fac69702d...	55a41f8069702d...	-71.2241489400...	42.302...
21	2015/03/09 11:41:20	<undefined>	<undefined>	NaN	NaN	55a41fac69702d...	55a41f8069702d...	-71.2241965700...	42.302...
22	2015/03/09 11:41:21	<undefined>	<undefined>	NaN	NaN	55a41fac69702d...	55a41f8069702d...	-71.2242442600...	42.302...
23	2015/03/09 11:41:22	55a41b7569702...	55a3fe3569702d...	-83.6290871400...	42.47475853000...	55a41fac69702d...	55a41f8069702d...	-71.2242942000...	42.302...
24	2015/03/09 11:41:23	55a41b7569702...	55a3fe3569702d...	-83.6290943400...	42.47477350000...	55a41fac69702d...	55a41f8069702d...	-71.2243444800...	42.302...
25	2015/03/09 11:41:24	55a41b7569702...	55a3fe3569702d...	-83.6290863700...	42.47476400000...	55a41fac69702d...	55a41f8069702d...	-71.2243067200...	42.302...

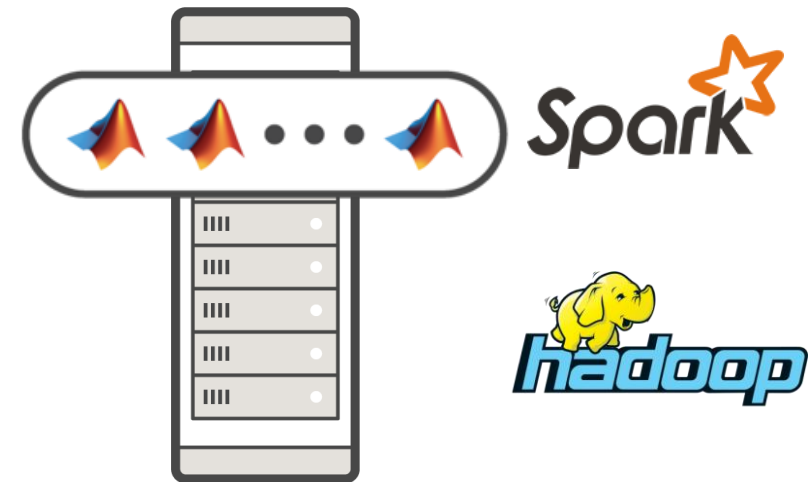
フリートデータ解析実践 クラスターへのスケールアウト

ステップ2

デスクトップの限界



- 処理時間
- データコピーの手間
- ディスク容量



フリートデータ解析実践

クラスターへのスケールアウト

- Hadoop  **hadoop**
 - ビッグデータのためのプラットフォーム
 - **HDFS** : 分散ファイルシステム
 - **YARN** : リソース管理やスケジューリング
 - **Hadoop MapReduce**: MapReduceフレームワークの実装

- Spark 
 - クラスター計算のフレームワーク
 - フォールトトレラントな分散処理
 - **機械学習**のような繰り返し演算に適性あり
 - **Hadoop YARN**と連携可能

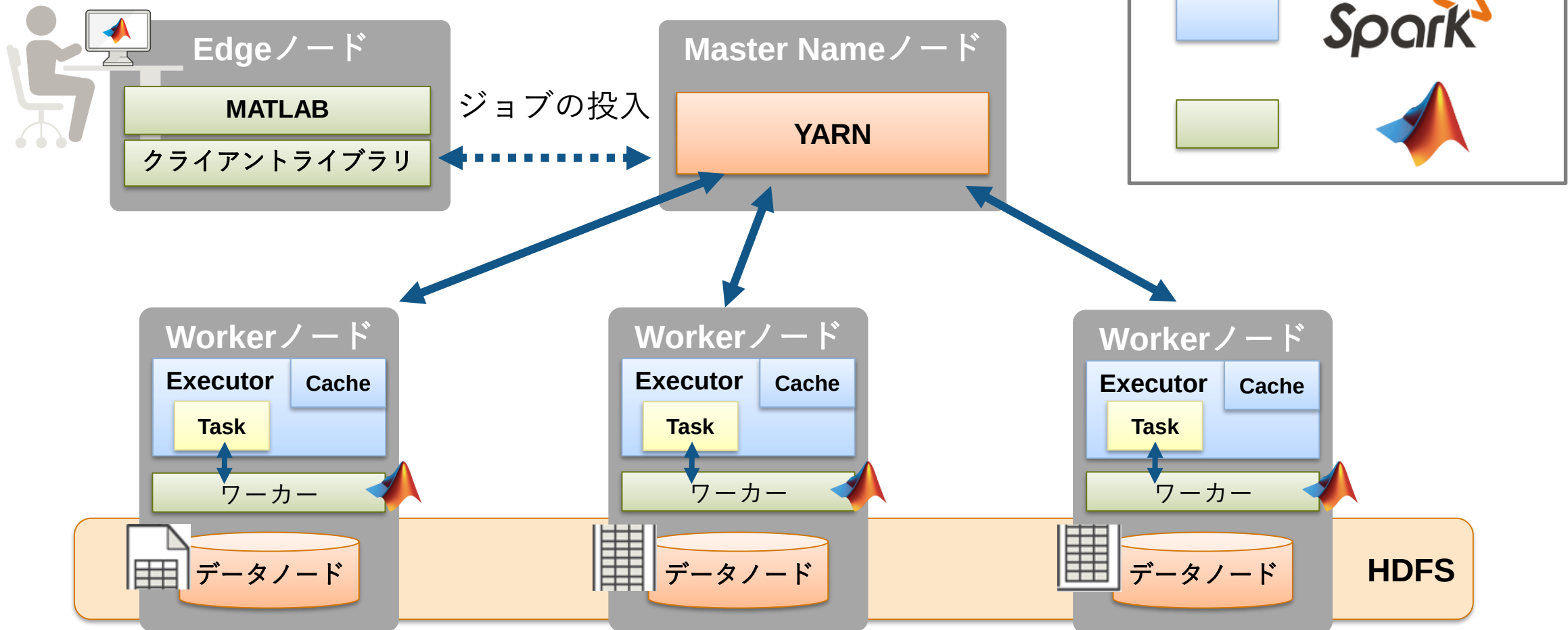
フリートデータ解析実践 クラスターへのスケールアウト

■ Hadoop/Sparkとの2つの連携方法

用途	使用する製品	対象ユーザー
MATLABからtall配列の処理をSparkで実行	MATLAB Distributed Computing Server 	試行錯誤でアルゴリズムを検討するユーザー
- tall配列の処理をSpark上でスタンドアロンアプリで実行 - Spark APIを使ったスタンドアロンアプリの実行	MATLAB Compiler™  	定形処理を利用するユーザー - SparkのAPIを詳細に制御した処理を作成したいユーザー

フリートデータ解析実践 クラスターへのスケールアウト

- MDCSを使ったHadoop/Spark連携



フリートデータ解析実践 クラスターへのスケールアウト

MATLABデスクトップでの処理

個々のトリップファイルの読み取り

```
dataDir = fullfile('..', 'LogFiles', '*.mat');
```

fileDatastoreを使ったカスタム読み込み関数での読み取り

```
readFcn = @(filename) readTrip(filename, tripTable);  
fds = fileDatastore(dataDir, 'ReadFcn', readFcn, 'UniformRead', true);
```

tall配列の作成

```
tt = tall(fds);
```

クラスターの設定とファイルパスの変更だけで
MATLABの処理は書き直さずにスケールアウト

Hadoop/Sparkクラスターでの処理

Hadoopクラスターの設定

```
setenv('HADOOP_HOME', '/usr/local/hadoop/hadoop-2.9.1');  
setenv('HADOOP_PREFIX', getenv('HADOOP_HOME'));  
setenv('SPARK_HOME', '/usr/local/spark/spark-2.3.2-bin-hadoop2.7');  
cluster = parallel_cluster.Hadoop;
```

MDCSインストールディレクトリ及びクラスターに渡すスクリプトの指定

```
cluster.ClusterMatlabRoot = '/usr/local/MATLAB/MDCS/R2018b';  
cluster.AttachedFiles = {'readTrip.m'};
```

Hadoopクラスターをtall配列の実行環境として設定

```
mapreducer(cluster);
```

個々のトリップファイルの読み取り

```
dataDir = 'hdfs://hadoopmaster:54310/Fleet/*.mat';
```

fileDatastoreを使ったカスタム読み込み関数での読み取り

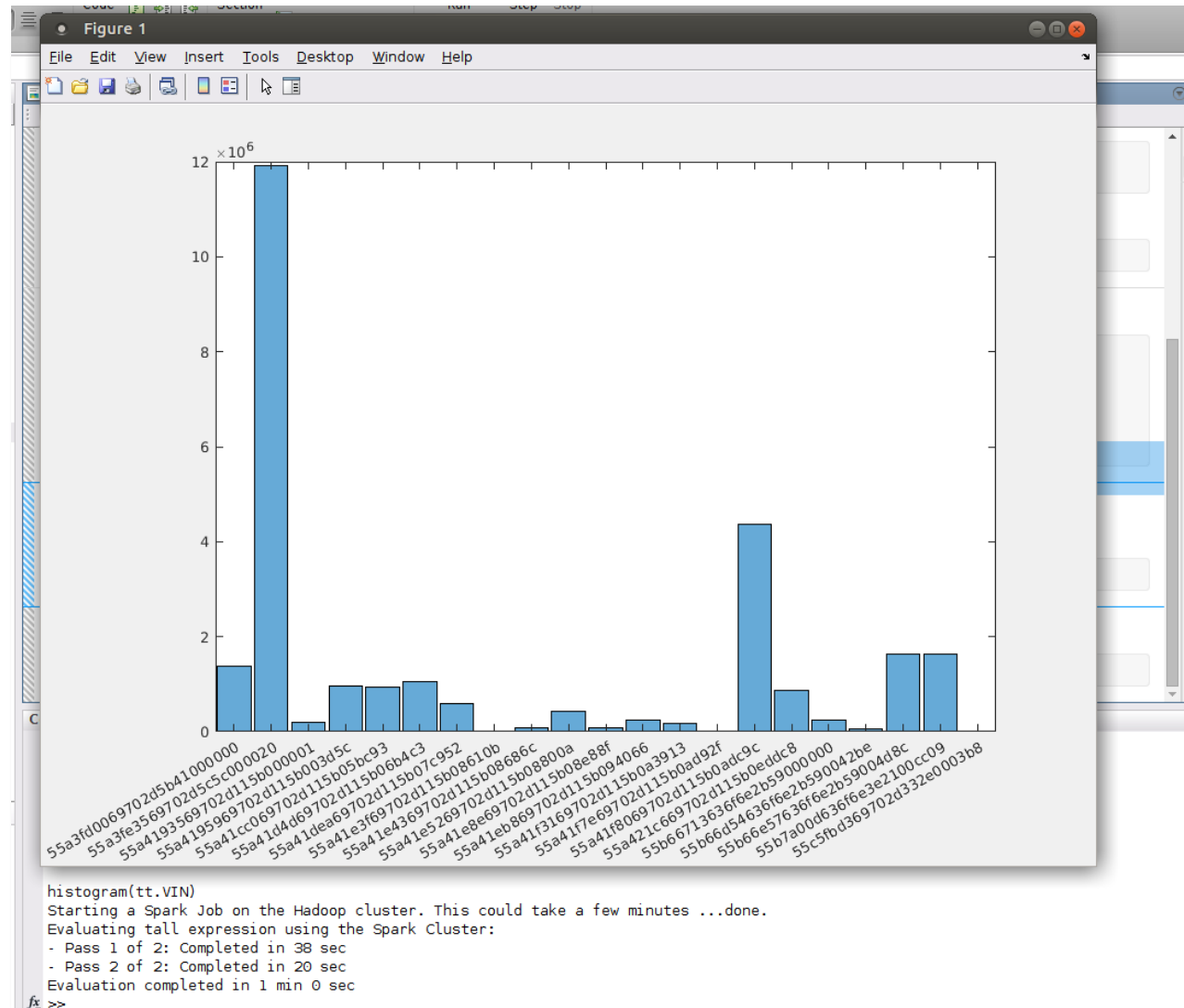
```
readFcn = @(filename) readTrip(filename, tripTable);  
fds = fileDatastore(dataDir, 'ReadFcn', readFcn, 'UniformRead', true);
```

tall配列の作成

```
tt = tall(fds);
```

フリートデータ解析実践

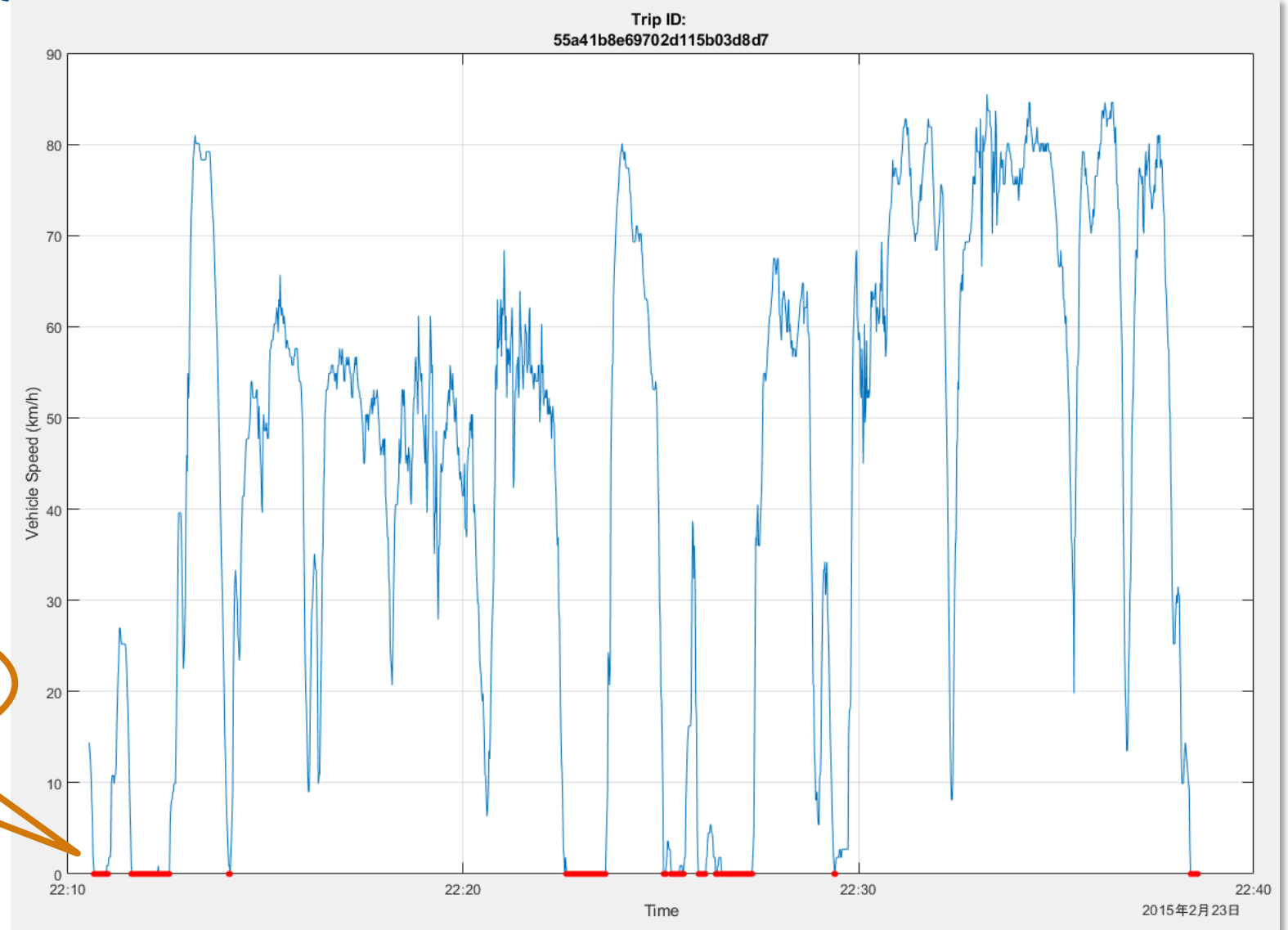
クラスターへのスケールアウト



フリートデータ解析実践

走行データ解析

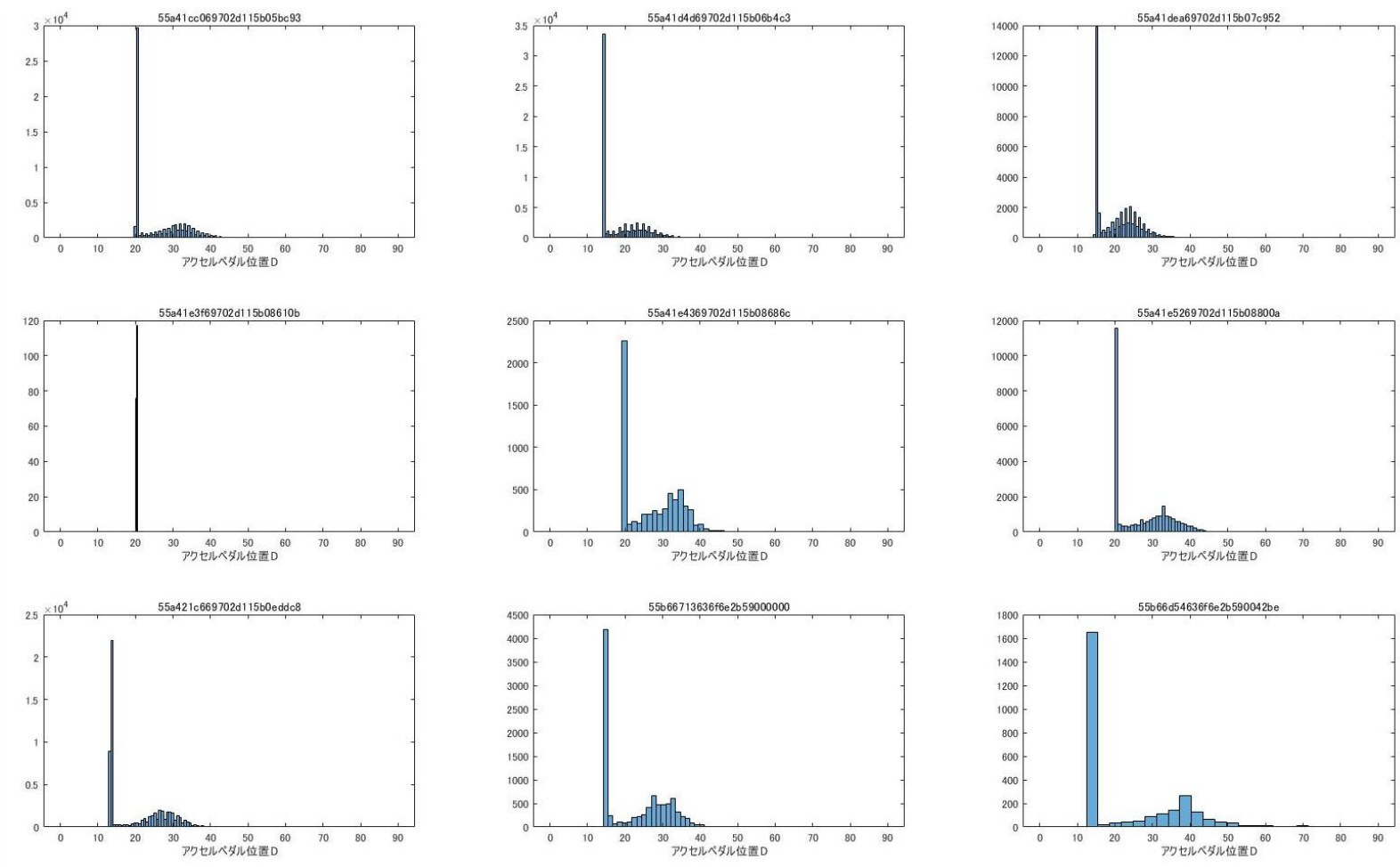
目的：停止イベントを検出する



フリーデータ解析実践

走行データ解析

目的：ドライバーの運転の特徴を調べる

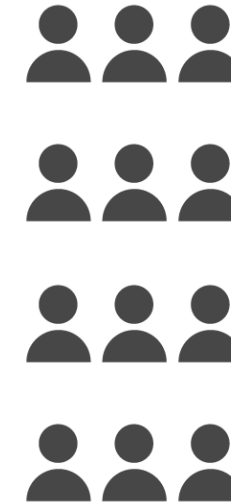


MATLAB解析のシステムへの統合

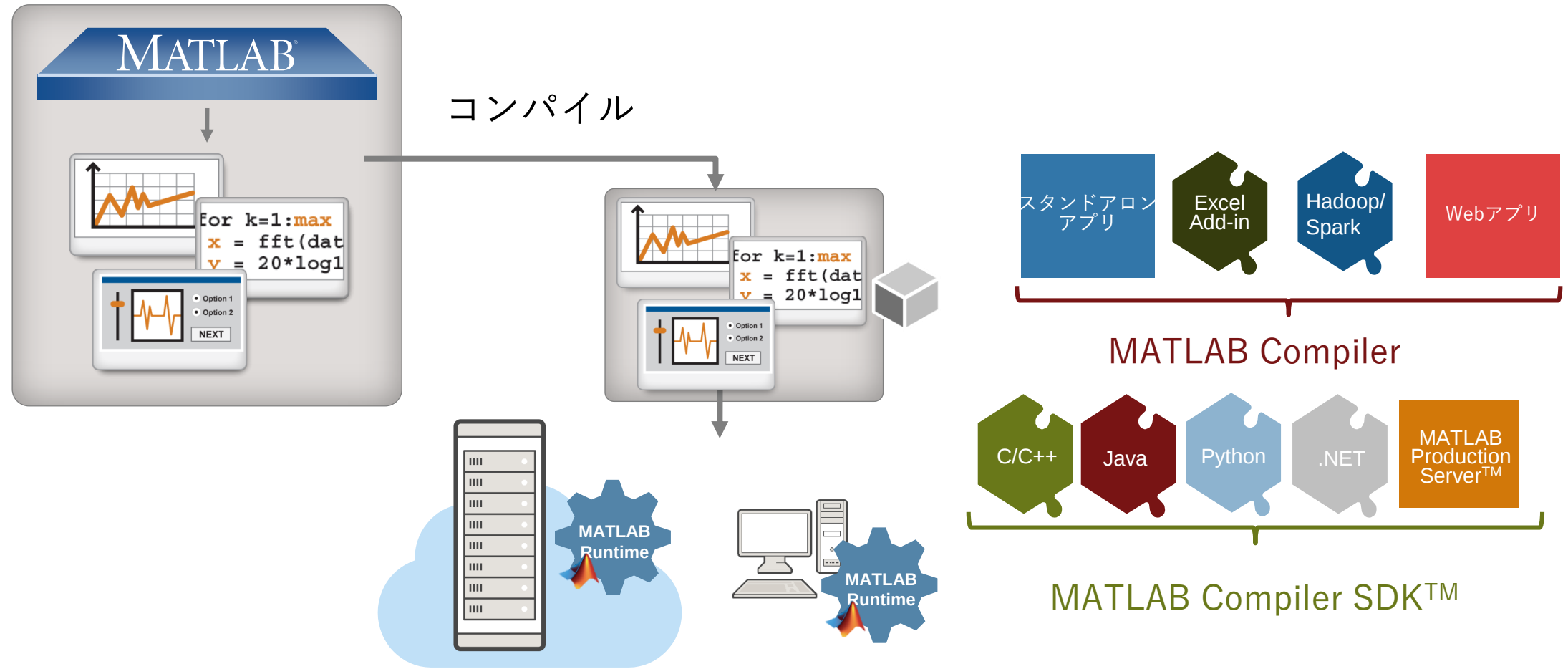
MATLAB解析のシステムへの統合

MATLABで作成したアルゴリズムを広く使ってもらいたい

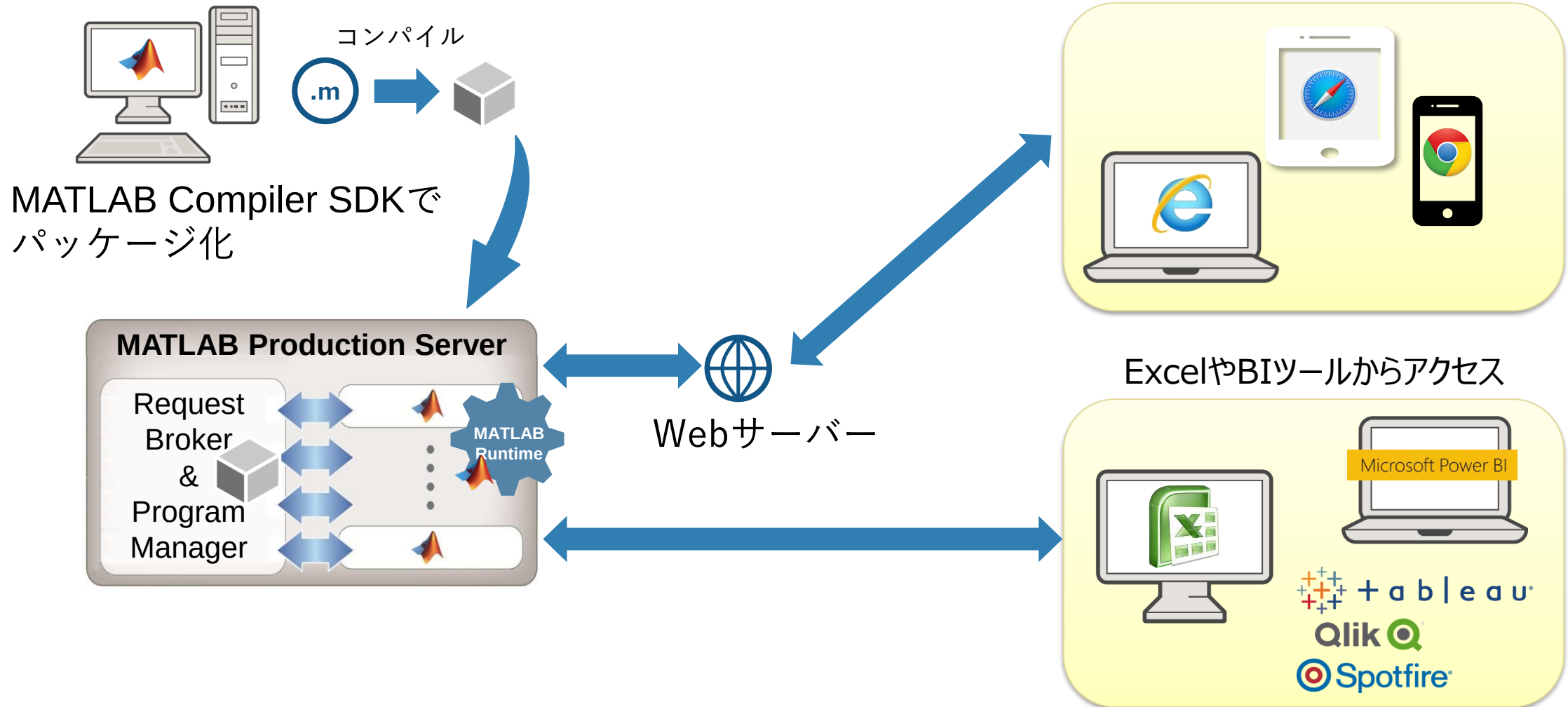
```
for k=1:max  
    x = fft(da  
    y = 20*log
```



MATLAB解析のシステムへの統合 コンパイル



MATLAB解析のシステムへの統合 エンタープライズへの展開

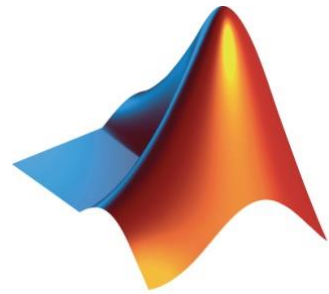


まとめ

Key Takeaways

1. 大規模なフリートデータを **datastore** や **tall** を用いて簡単に処理
2. デスクトップPCからHadoop/Sparkクラスターへ、**コードを書き直さずにスケールアウトが容易**
3. MATLABによるデータ解析をすぐに**エンタープライズシステムに統合**

眠っているビッグデータを
MATLABで解析してみませんか？



MathWorks®

Accelerating the pace of engineering and science

© 2018 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

参考：使用したToolbox

項目	使用した製品
datastoreでのデータ読み込み	(MATLAB)
tall配列を用いた処理	(MATLAB) Parallel Computing Toolbox Statistics and Machine Learning Toolbox™
データのプロット	(MATLAB) Mapping Toolbox™
Hadoop/Sparkクラスターでの処理	MATLAB Distributed Computing Server