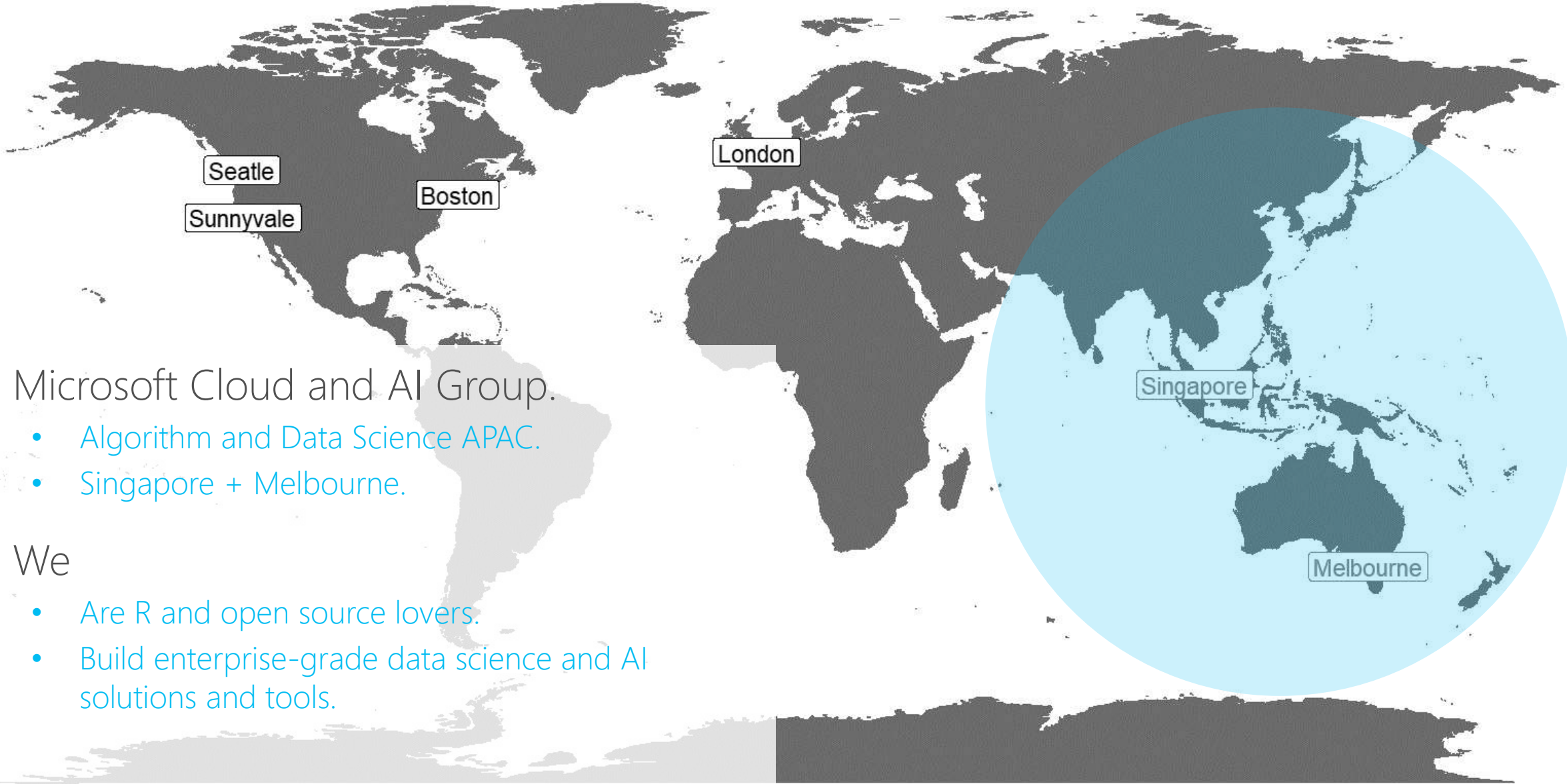
R you ready for Cloud and AI?

Rihad Variawa, Data Scientist

Self-introduction



- Microsoft Cloud and AI Group.
  - Algorithm and Data Science APAC.
  - Singapore + Melbourne.
- We
  - Are R and open source lovers.
  - Build enterprise-grade data science and AI solutions and tools.

A glimpse of R

Firstly appeared in the year of 1993, created by Ross Ihaka and Robert Gentleman.

Programming language and software environment for statistical computing.

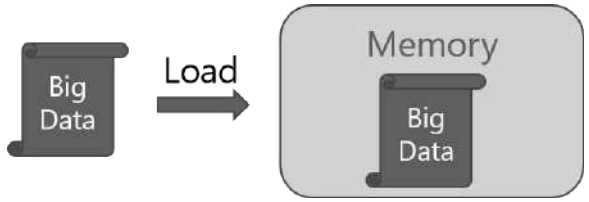
More than 11000 packages on CRAN, Github, Bioconductor, Bitbucket, etc.

Cross-platform, functional, graphical, etc.

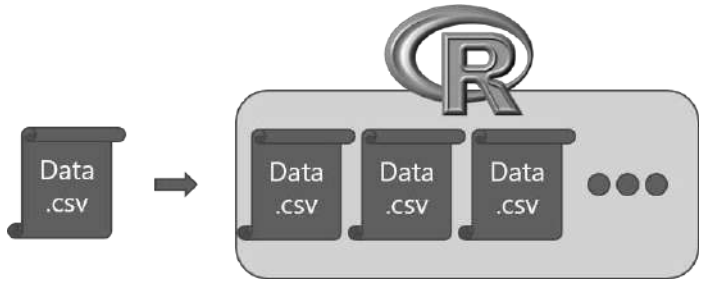
A glimpse of R

- R in the past
  - Deficiency in execution mechanisms.
  - Inadequacy of community and commercial support.

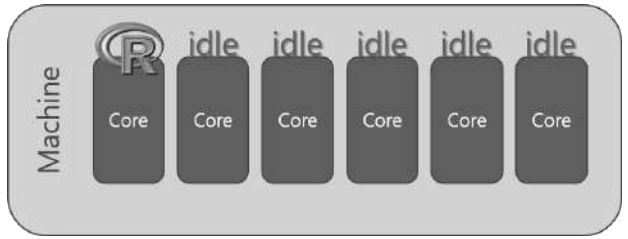
In-memory operation



Expensive data movement and duplication



Lack of parallelism



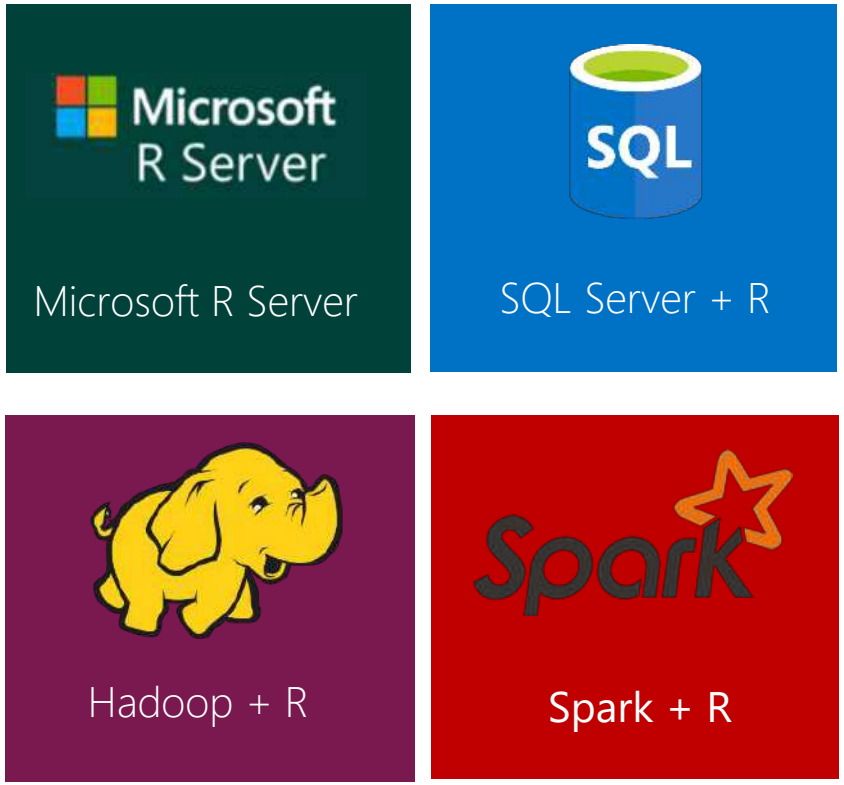
Inadequacy of community support

Lack of guaranteed support timeliness

No SLAs or support models

A glimpse of R

- R in 2017
  - Scalable, parallelizable, product-ready, and commercial support from eco-system.
  - Big data, deep learning, and artificial intelligence.



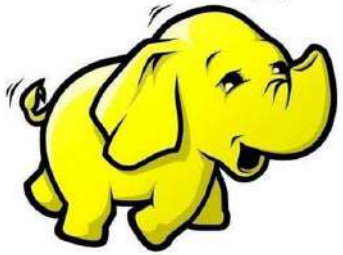
acxiom
dv01
eToro
PROS
Powering Modern Commerce with Dynamic Pricing Science
MERCK
125 YEARS
HEARTLAND BANK

R consortium
Microsoft
RStudio
ROpenSci
software carpentry
rOpenGov

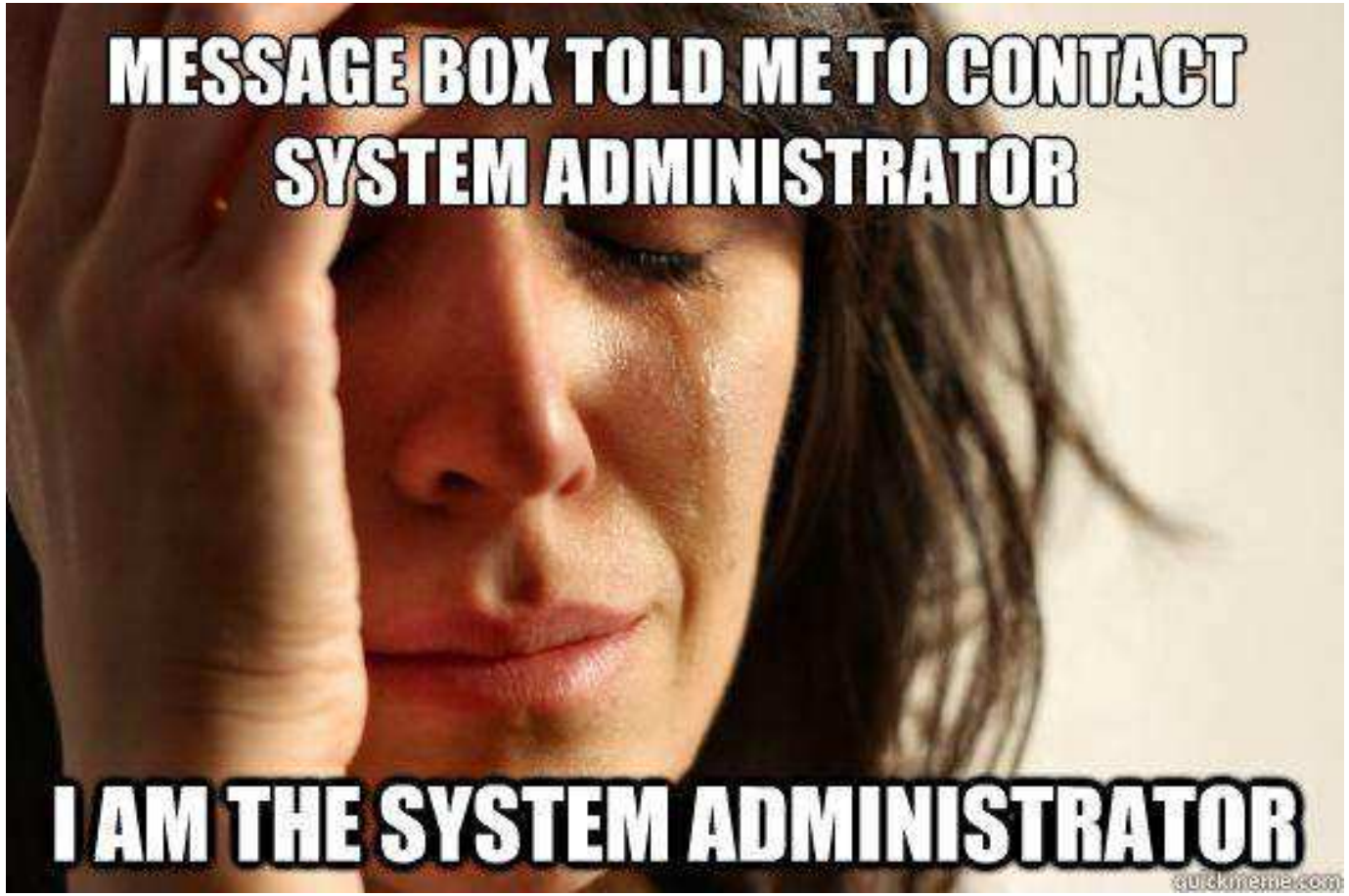
https://www.slideshare.net/RevolutionAnalytics/r-then-and-now

Why cloud computing

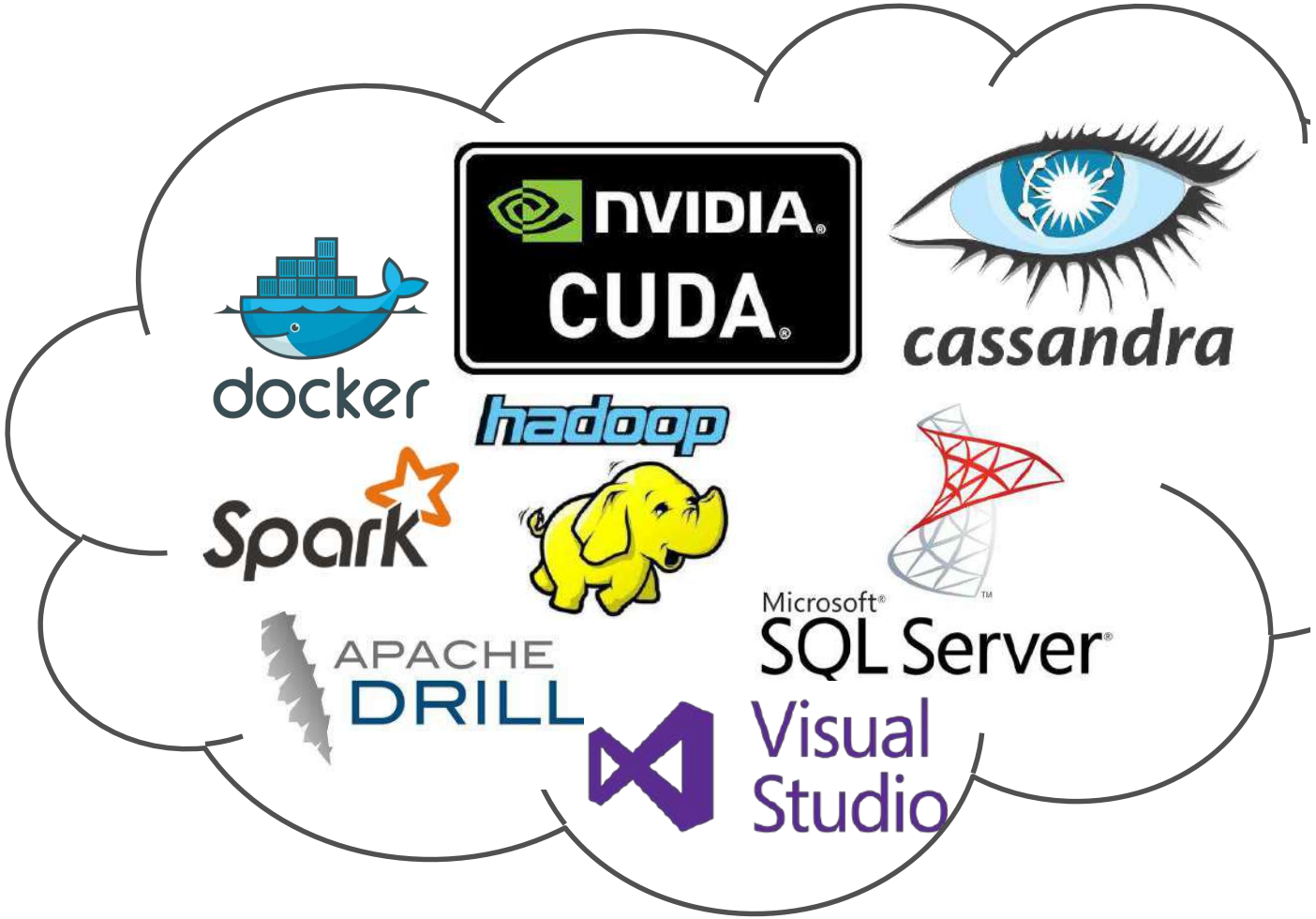
- Before practicing with the state-of-the-art technologies for your R analytical work, you probably need to
  1. Scale up/out cluster.
  2. Set up Hadoop/Spark.
  3. Install and configure software/runtime.
  4. …



Why cloud computing



Why cloud computing



Why cloud computing

Cloud computing with R

- R on cloud instances
  - Administration and operation of cloud resources.
  - Run large-scale analytical jobs.
  - Interactively prototype and develop data science or AI solutions.
  - Deploy applications or services.

Resource management	Scalable analytics	Interaction with remote	Application deployment
AzureSMR AzureDSVM doAzureParallel …	RevoScaleR MicrosoftML doAzureParallel dplyrXdf parallel doSNOW sparklyr SparkR …	IDE (Rstudio Server, Jupyter Notebook, etc.) Remote Desktop SSH mrsdeploy remoter …	mrsdeploy AzureML. Shiny Docker container …

Cloud computing with R – resource admin

- AzureSMR
  - Managing a selection of Azure resources such as storage blobs, HDInsight, etc.
- AzureDSVM
  - Deployment and operation of Azure Data Science Virtual Machine (DSVM).
  - Remote execution of script and file transfer with a Linux DSVM.
  - Retrieval of cost and expense information of using DSVM.

```
# Authentication.
sc <- createAzureContext(tenant="{TID}",
                         clientID="{CID}",
                         authKey="{KEY}")
azureAuthenticate()

azureListSubscriptions()
```

```
# DSVM operation – deployment.
deployDSVM(sc,
           resource.group="{resource group}",
           location="{location}",
           hostname="{DSVM name}",
           username="{user name}",
           size="{DSVM size}",
           authen="{authentication method}",
           password="{password}")
```

Cloud computing with R – scalable analytics

- RevoScaleR – collection of functions for practicing data science in scale
  - Stream data into RAM in blocks – "big data" in any size.
  - XDF file format – optimized file format to speed up iterative algorithm processing. Used with dplyrXdf for convenient manipulation.
  - Support for various computing contexts, local multi-core, multi-node across cluster, SQL server, and Hadoop/Spark.

```
### LOCAL COMPUTING CONTEXT ###
# Set up local environment variables.
rxSetComputeContext("localpar")

# Create Linux directory and file objects.
linuxFS <- RxNativeFileSystem( )
AirlineDataSet <- RxXdfData("airline_20MM.xdf",
                            fileSystem = linuxFS)
```

```
### LOCAL COMPUTING CONTEXT ###
# Setup of Spark environment variables.
mySparkCC <- RxSpark()

# Spark compute context.
rxSetComputeContext(mySparkCC)

# Create HDFS directory and file objects.
hdfsFS <- RxHdfsFileSystem()
AirlineDataSet <- RxXdfData("airline_20MM.xdf",
                            fileSystem = hdfsFS)
```

Cloud computing with R – scalable analytics

- sparklyr – R interface to Apache Spark with a complete backend of dplyr
  - Filter and aggregate Spark datasets and then bring them into R for analysis and visualization.
  - Use Spark's distributed machine learning library from R.
  - Create extensions to call Spark API and provide interfaces to Spark packages.
- SparkR – light-weight frontend to use Apache Spark in R
  - Operations like filtering, grouping, selecting, etc., and MLib model training.

```
# sparklyr for data manipulation.
sc <- spark_connect(master="local")

iris_tbl <- copy_to(sc, iris)
flights_tbl <- copy_to(sc, flights, "flights")

iris_preview <-
   dbGetQuery(sc, "SELECT * FROM iris LIMIT 10")
flights_tbl <- filter(dep_delay == 2)
```

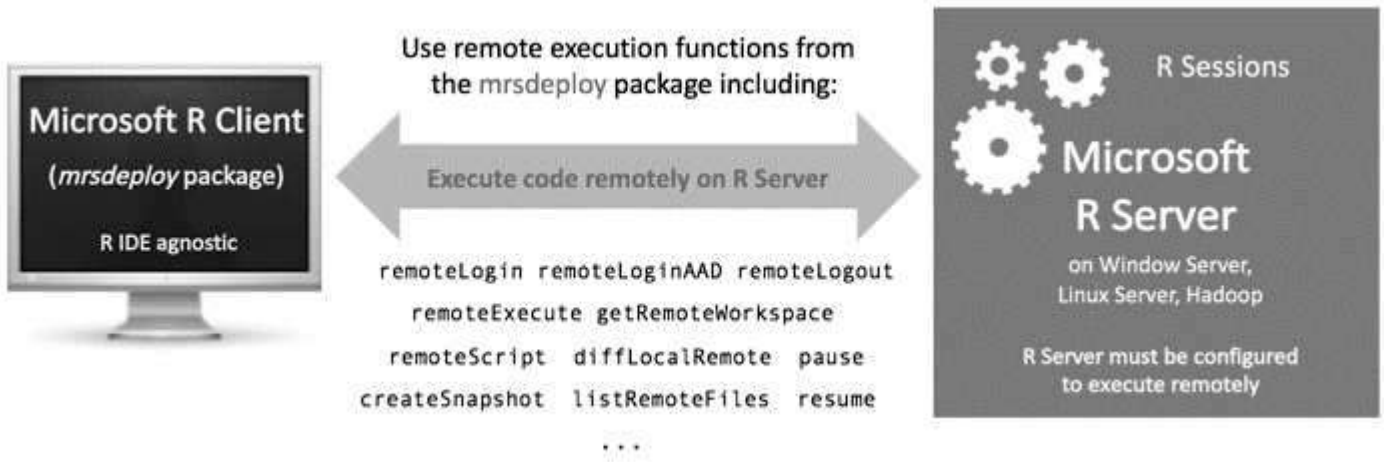
https://spark.rstudio.com/

```
# SparkR for data manipulation
sparkR.session(master="local[*]",
                  sparkConfig=list(spark.driver.memory="2g")

df <- as.DataFrame(faithful)

head(select(df, df$eruptions))
head(select(df, "eruptions"))
```

https://spark.apache.org/docs/latest/sparkr.html

Cloud computing with R – remote interaction

- IDEs
  - Rstudio Server.
  - Jupyter Notebook.
- Remote Desktop (Windows RDC or X2Go)
- R packages
  - Microsoft mrsdeploy.




```
# Login to the remote R Server.
remoteLogin(deployr_endpoint, session, ...)

# Remote execution.
remoteExecute(rcode, script, ...)

# Remote interaction if session is TRUE.
# R session at remote machine will be created.
# REMOTE> x <- rnorm(100)
```

Cloud computing with R – app deployment

- Shiny
  - A web application framework for R.
  - Interactive web application with no prior knowledge of JS, HTML, or CSS required.
- mrsdeploy
  - Easy to deploy and consume web based service in R session.
  - Compatible with Swagger.

```
# Publish a standard service 'mtService' version 'v1.0.0'. Assign service to 'api' variable.
api <- publishService( "mtService",
                       code = manualTransmission,
                       model = carsModel,
                       inputs = list(hp = "numeric", wt = "numeric"),
                       outputs = list(answer = "numeric"),
                       v = "v1.0.0" )
result <- api$manualTransmission(120, 2.8)
```

Cloud computing with R – app deployment

- AzureML
  - Interaction with Azure Machine Learning (AML) Studio.
  - Deployment of AML service in R session.

```
# Publish a web service on AzureML.
add <- function(x, y) {x +y}
ws <- workspace()
api <- publishWebService(ws, fun=add, name="AzureML-vignette-silly",
                                      inputSchema=list(x="numeric", y="numeric"),
                                      outputSchema=list(ans="numeric"))
```

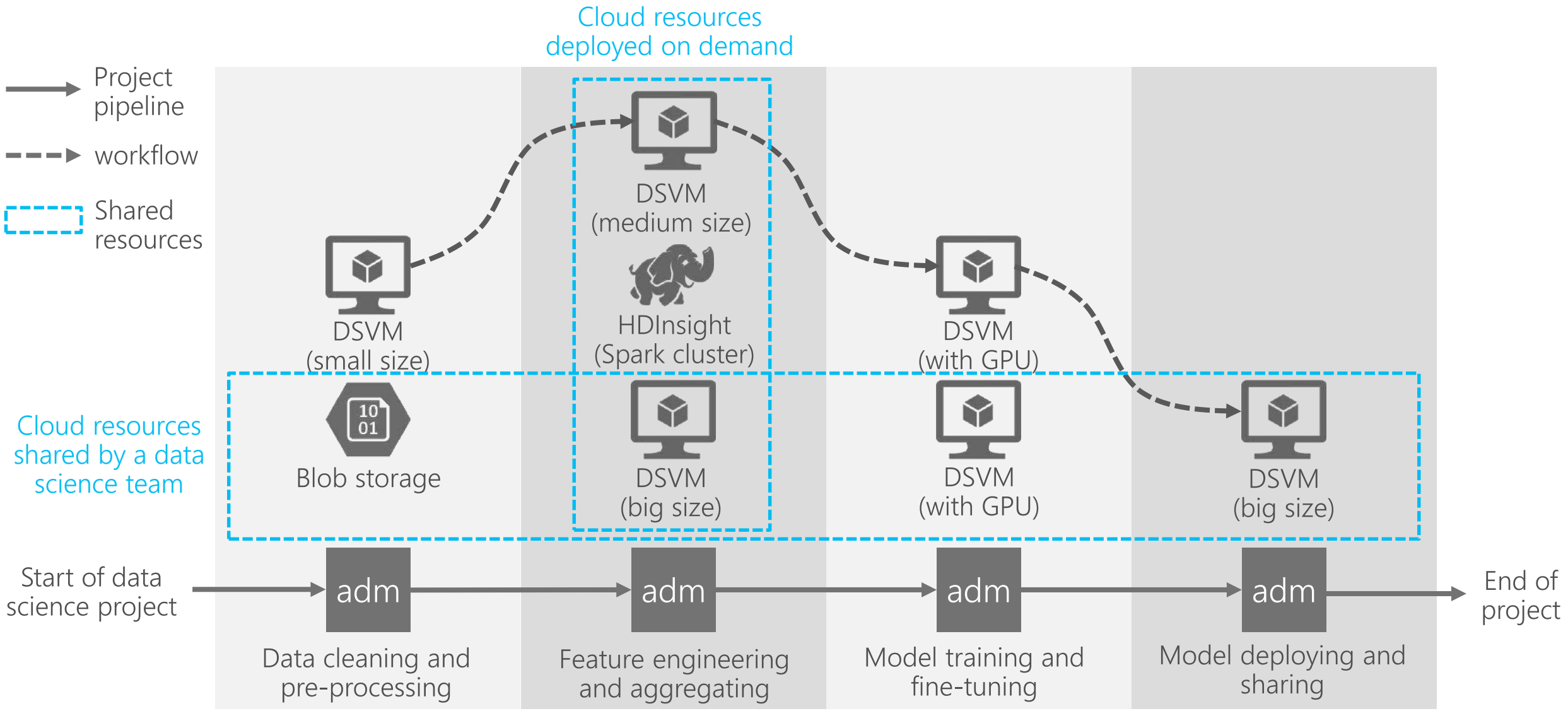
https://cran.r-project.org/web/packages/AzureML/vignettes/getting_started.html

- Docker container
  - Docker hub/Azure Container Registry.
  - Azure Container Services/Azure Container Instances.

Data science and AI pipeline on cloud

- Operationalizing, managing, and administrating an enterprise-grade data science development/production pipeline ALL BY USING R.
- Benefits inherited from Azure cloud platform
  - Elasticity of resource deployment.
  - Security and role access control.
  - Computational efficiency of on-demand high-performance and distributed data analytics.
  - Ease of modularization.
  - Cost effectiveness.
- Collaborative development SW/HW environment for R-user data scientist, data engineer, and solution architect teams – Microsoft R Server, Hadoop/Spark, GPU-incorporated virtual machine, Docker container, etc.

Operationalization of a data science pipeline



Operationalization of a data science pipeline

Elastic deployment and use of cloud resources ranging from data/file storage, single virtual machine node, to large-scale high-performance computing cluster.

Azure cloud resources

Local R packages:
- AzureSMR
- AzureDSVM

Key functions:
- deployDSVM
- deployDSVMCluster
- azureCreateHDI
- azureCreateStorageAccount
- azureDeleteResourceGroup

Azure AD Application

adm

Management and operation of Azure cloud resources.

Serverless-like data analytics execution on varieties of computing instances on cloud.

Azure cloud computing resources

Local R packages:
- MicrosoftML
- mrsdeploy
- doAzureParallel
- dplyrXDF

Key functions:
- remoteLogin
- remoteExecute
- rxExeBy
- rxFastTrees
- rxNeuralNet

adm

High-performance and distributed data analytics

AI service deployment with a micro-service framework.

Azure cloud resources

adm

Model and service deployment

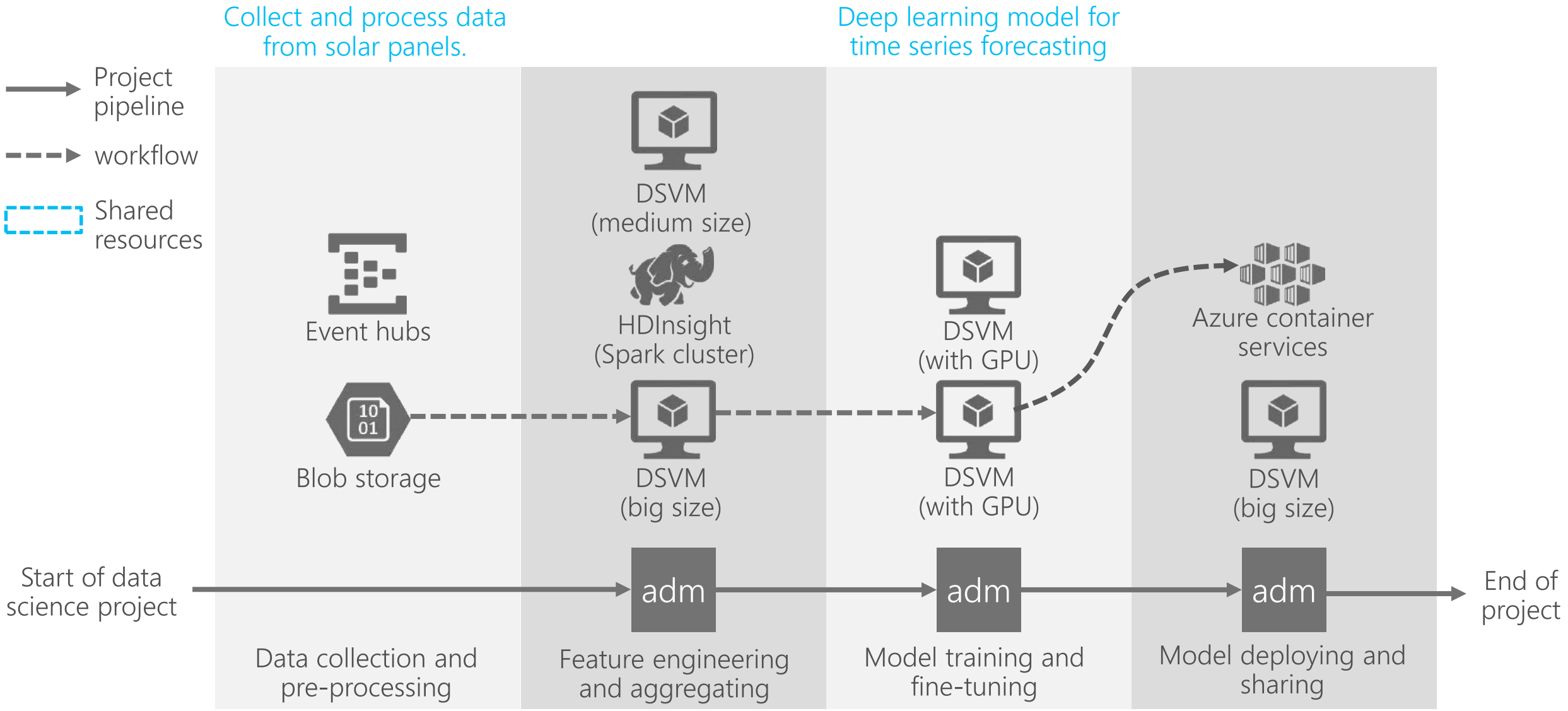
Talk is cheap. Show me the code.

- *Linus Torvalds*

Use case 1 – Solar power forecasting

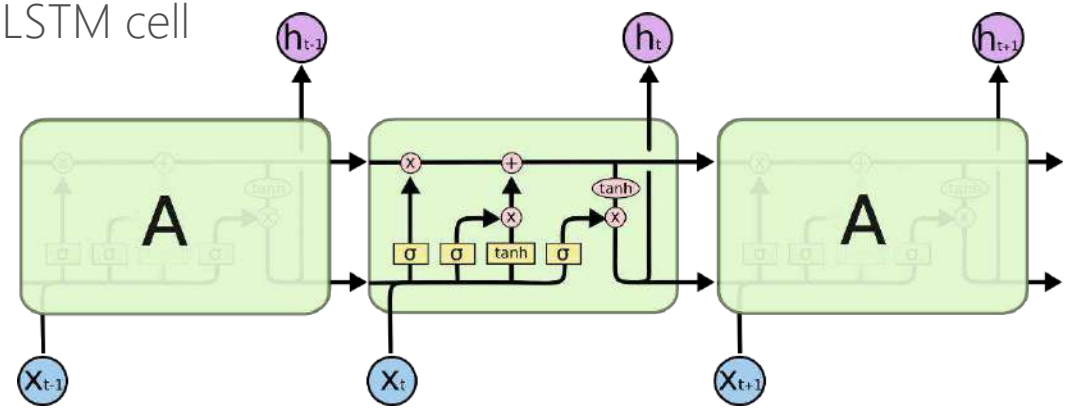
- Solar power forecasting is essential for efficient use, the management of electricity grid, and solar energy trading.
- Stochastically learning method fits in resolving the forecasting problem.
  - Modeling the time series.
  - Forecasting the long-term dependencies on the univariates.
- Deep learning based approach.
  - Algorithm: Long Short-Term Memory (LSTM).
  - Computing engine: DSVM with GPU.
  - Data set: photovoltaic systems device reading from solar panels.

Solar power forecasting – operationalization



Solar power forecasting with LSTM

LSTM cell

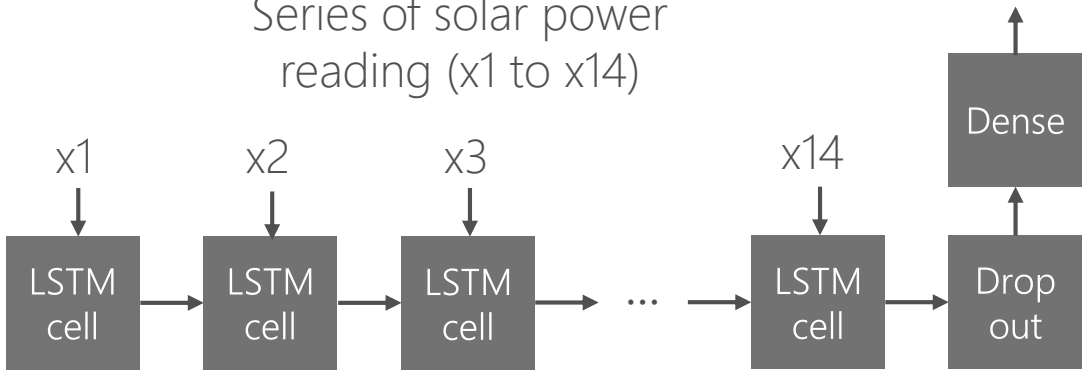


Original data

Time	Current	Total
2013-12-01 07:00:00	6.3	1.69
2013-12-01 07:30:00	44.3	11.36
2013-12-01 08:00:00	208.0	67.50
2013-12-01 08:30:00	482.0	250.5

Predicted total solar power (y)

Series of solar power reading (x1 to x14)



Prediction scenario & results

1.7,11.4 -> 10300
1.7,11.4,67.5 -> 10300
1.7,11.4,67.5,250.5 ... -> 10300