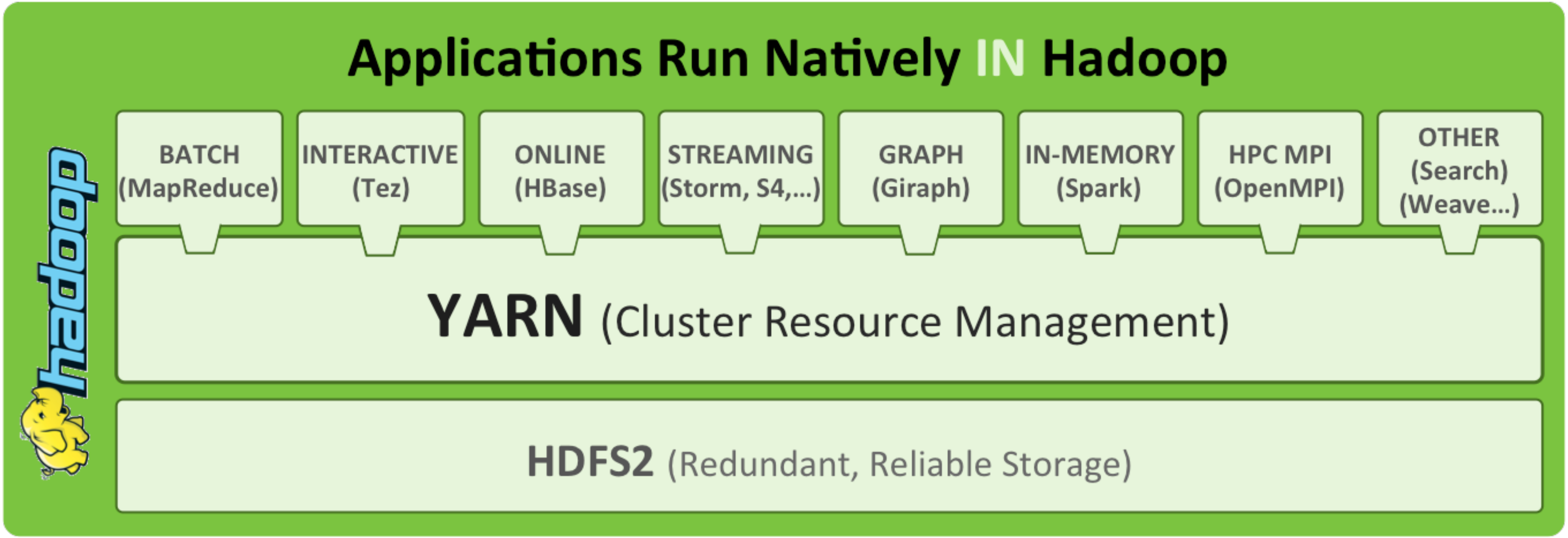
Data Munging and Analysis for Scientific Applications

Raminder Singh
Science Gateways Group
Indiana University, Bloomington
raminder@apache.org

Overview

- Evaluate the Apache Big data tools
- Understand the execution patterns of Analysis applications
- Solutions using Airavata
- Build a gateways solution with HPC and Big Data requirements

Hadoop 2 Ecosystem


Applications Run Natively IN Hadoop
hadoop
BATCH (MapReduce)
INTERACTIVE (Tez)
ONLINE (HBase)
STREAMING (Storm, S4,...)
GRAPH (Giraph)
IN-MEMORY (Spark)
HPC MPI (OpenMPI)
OTHER (Search) (Weave...)
YARN (Cluster Resource Management)
HDFS2 (Redundant, Reliable Storage)

http://hortonworks.com/hadoop/yarn/

Motivation to explore

- Heterogeneous data
- Data Munging (parsing, scraping, formatting data)
- Visualization or Analyze
- Preservation of data

Analysis Applications

- Behavior Tracking - medical
- Situational Awareness - weather
- Time Series Data -Patient monitoring, weather data to help farmers
- Resource consumption Monitoring - Smart grid
- Process optimization

Scientific applications Data Types

Observational Data – uncontrolled events happen and we record data about them.

Examples include astronomy, earth observation, geophysics, medicine, commerce, social data, the internet of things.

Experimental Data – we design controlled events for the purpose of recording data about them.

Examples include particle physics, photon sources, neutron sources, bioinformatics, product development.

Simulation Data – we create a model, simulate something, and record the resulting data.

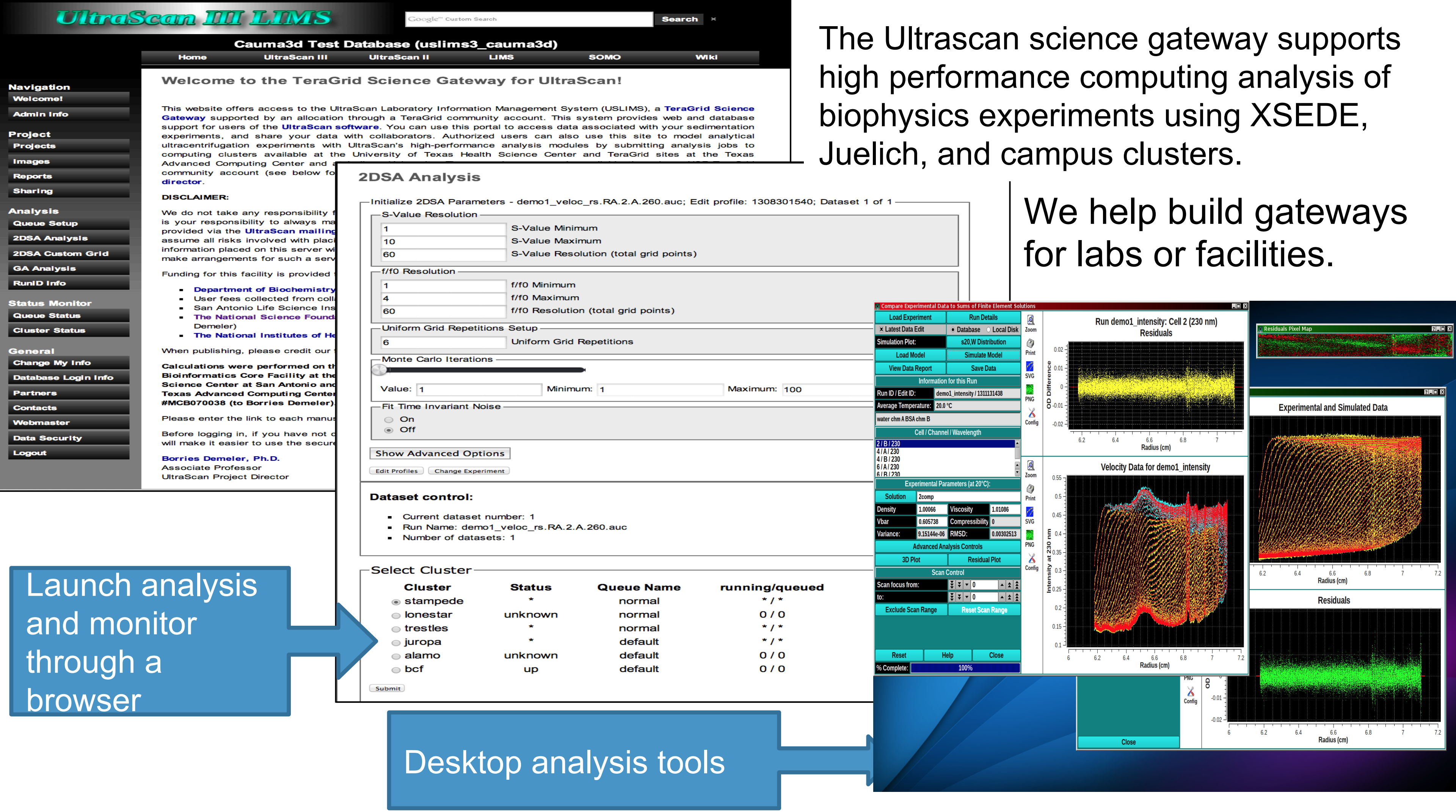
Examples include weather & climate, nuclear & fusion energy, high-energy physics, materials, chemistry, biology, fluid dynamics.

What is Science Gateway?

- Community portal or desktop tools
- Common science theme
- Collaborative environment

The Ultrascan science gateway supports high performance computing analysis of biophysics experiments using XSEDE, Juelich, and campus clusters.

We help build gateways for labs or facilities.



UltraScan III LIMS

Google™ Custom Search | Search

Cauma3d Test Database (uslims3_cauma3d)

Home | UltraScan III | UltraScan II | LIMS | SOMO | Wiki

Navigation
- Welcome!
- Admin Info

Project
- Projects
- Images
- Reports
- Sharing

Analysis
- Queue Setup
- 2DSA Analysis
- 2DSA Custom Grid
- GA Analysis
- RunID Info

Status Monitor
- Queue Status
- Cluster Status

General
- Change My Info
- Database Login Info
- Partners
- Contacts
- Webmaster
- Data Security
- Logout

Welcome to the TeraGrid Science Gateway for UltraScan!

This website offers access to the UltraScan Laboratory Information Management System (USLIMS), a TeraGrid Science Gateway supported by an allocation through a TeraGrid community account. This system provides web and database support for users of the UltraScan software. You can use this portal to access data associated with your sedimentation experiments, and share your data with collaborators. Authorized users can also use this site to model analytical ultracentrifugation experiments with UltraScan's high-performance analysis modules by submitting analysis jobs to computing clusters available at the University of Texas Health Science Center and TeraGrid sites at the Texas Advanced Computing Center and a
community account (see below fo
director.

DISCLAIMER:

We do not take any responsibility f
is your responsibility to always ma
provided via the UltraScan mailing
assume all risks involved with placi
information placed on this server wi
make arrangements for such a serv

Funding for this facility is provided

- Department of Biochemistry
- User fees collected from coll
- San Antonio Life Science Ins
- The National Science Found
Demeler)
- The National Institutes of He

When publishing, please credit our

**Calculations were performed on th
Bioinformatics Core Facility at the
Science Center at San Antonio and
Texas Advanced Computing Center
#MCB070038 (to Borries Demeler).**

Please enter the link to each manus

Before logging in, if you have not c
will make it easier to use the secure

Borries Demeler, Ph.D.
Associate Professor
UltraScan Project Director

2DSA Analysis

Initialize 2DSA Parameters - demo1_veloc_rs.RA.2.A.260.auc; Edit profile: 1308301540; Dataset 1 of 1

S-Value Resolution

1	S-Value Minimum
10	S-Value Maximum
60	S-Value Resolution (total grid points)

f/f0 Resolution

1	f/f0 Minimum
4	f/f0 Maximum
60	f/f0 Resolution (total grid points)

Uniform Grid Repetitions Setup

6	Uniform Grid Repetitions

Monte Carlo Iterations

Value: 1 Minimum: 1 Maximum: 100

Fit Time Invariant Noise
○ On
● Off

Show Advanced Options

Edit Profiles | Change Experiment

Dataset control:

- Current dataset number: 1
- Run Name: demo1_veloc_rs.RA.2.A.260.auc
- Number of datasets: 1

Select Cluster

	Cluster	Status	Queue Name	running/queued
●	stampede	*	normal	* / *
○	lonestar	unknown	normal	0 / 0
○	trestles	*	normal	* / *
○	juropa	*	default	* / *
○	alamo	unknown	default	0 / 0
○	bcf	up	default	0 / 0

Submit

Compare Experimental Data to Sums of Finite Element Solutions

Load Experiment | Run Details

☒ Latest Data Edit | ● Database ○ Local Disk

Simulation Plot: s20,W Distribution

Load Model | Simulate Model

View Data Report | Save Data

Information for this Run

Run ID / Edit ID: demo1_intensity / 1311131438

Average Temperature: 20.0 °C

water chm A BSA chm B

Cell / Channel / Wavelength

2 / B / 230
4 / A / 230
4 / B / 230
6 / A / 230
6 / B / 230

Experimental Parameters (at 20°C):

Solution	2comp		
Density	1.00066	Viscosity	1.01086
Vbar	0.605738	Compressibility	0
Variance:	9.15144e-06	RMSD:	0.00302513

Advanced Analysis Controls

3D Plot | Residual Plot

Scan Control

Scan focus from: 0

to: 0

Exclude Scan Range | Reset Scan Range

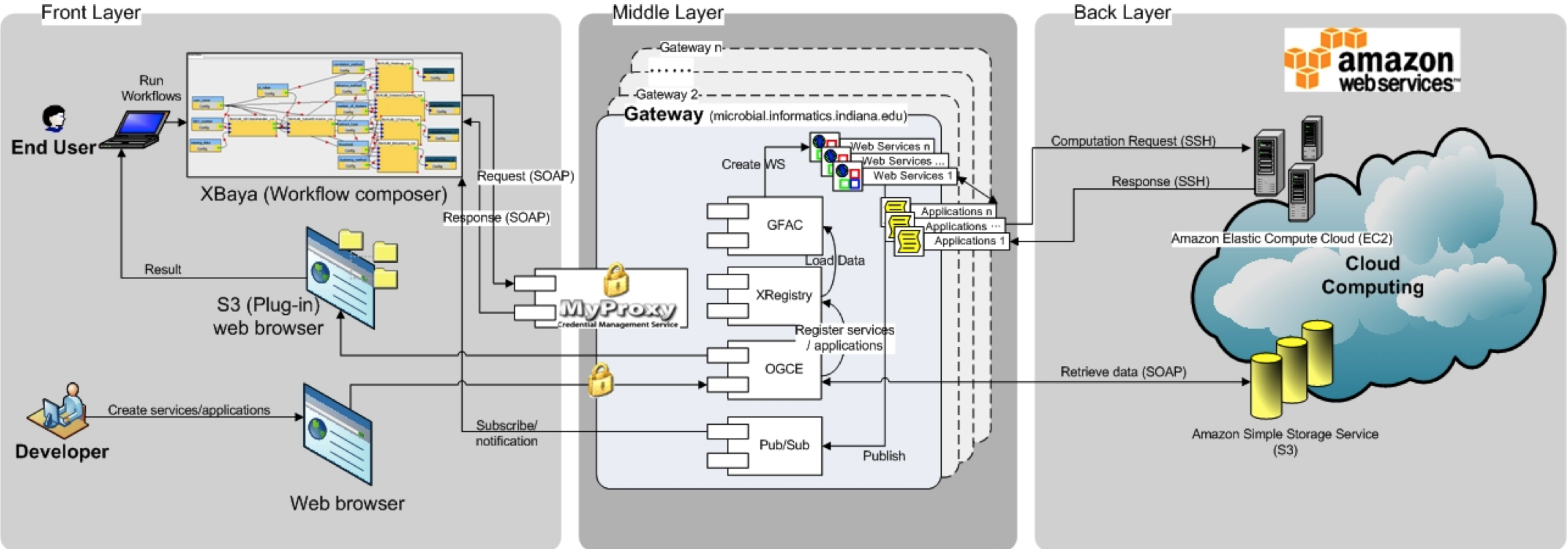
Reset | Help | Close

% Complete: 100%

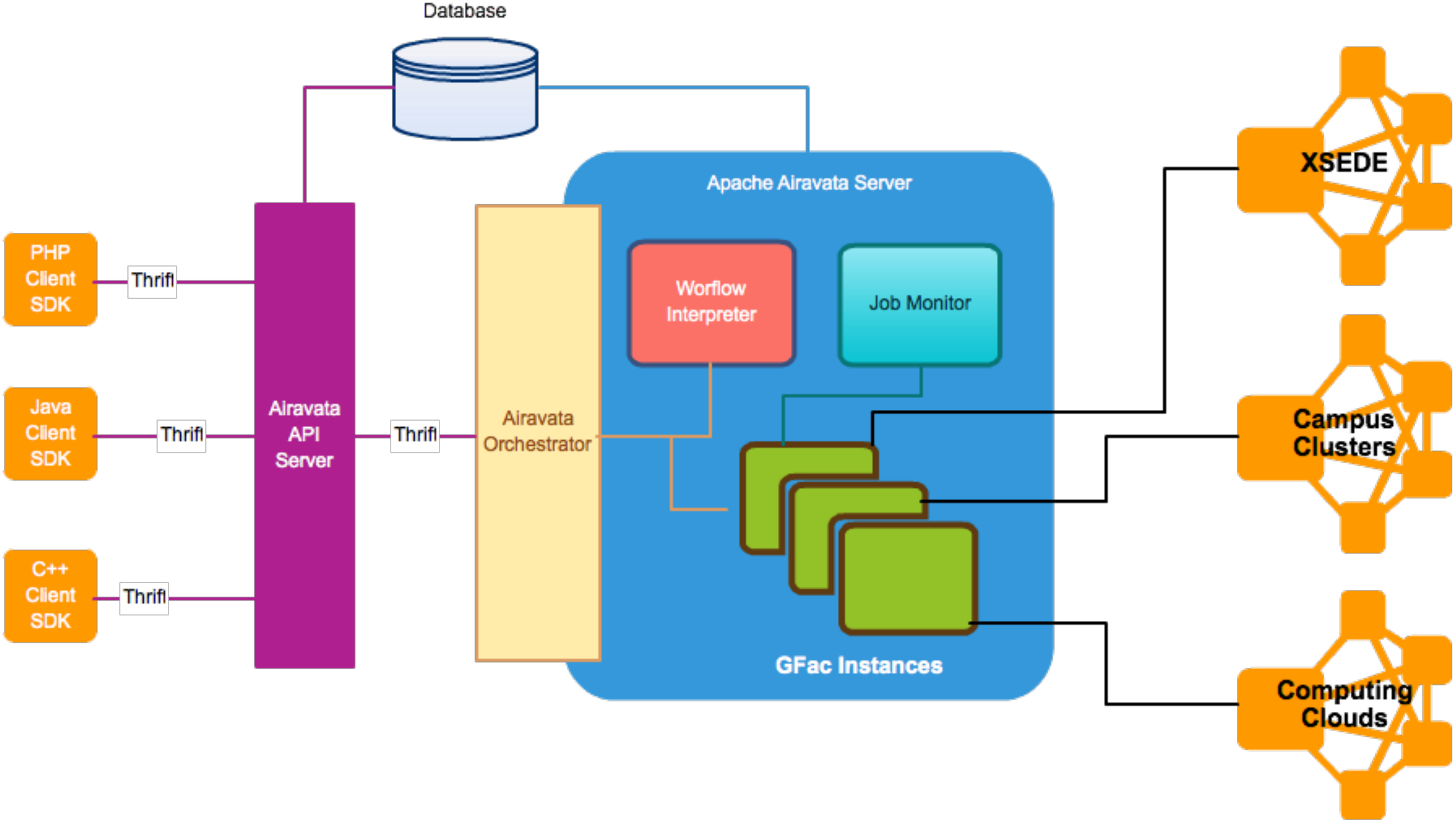
Launch analysis and monitor through a browser

Desktop analysis tools

BioVLAB


Front Layer
Run Workflows
End User
XBaya (Workflow composer)
Request (SOAP)
Response (SOAP)
Result
S3 (Plug-in) web browser
Developer
Create services/applications
Web browser
Subscribe/ notification
Middle Layer
Gateway n
Gateway 2
Gateway (microbial.informatics.indiana.edu)
Create WS
Web Services n
Web Services ...
Web Services 1
Applications n
Applications ...
Applications 1
GFAC
Load Data
XRegistry
Register services / applications
OGCE
Pub/Sub
Publish
MyProxy
Credential Management Service
Back Layer
amazon webservices
Computation Request (SSH)
Response (SSH)
Amazon Elastic Compute Cloud (EC2)
Cloud Computing
Retrieve data (SOAP)
Amazon Simple Storage Service (S3)

Airavata


Database
PHP Client SDK
Thrifl
Java Client SDK
Thrifl
C++ Client SDK
Thrifl
Airavata API Server
Thrifl
Airavata Orchestrator
Apache Airavata Server
Worflow Interpreter
Job Monitor
GFac Instances
XSEDE
Campus Clusters
Computing Clouds

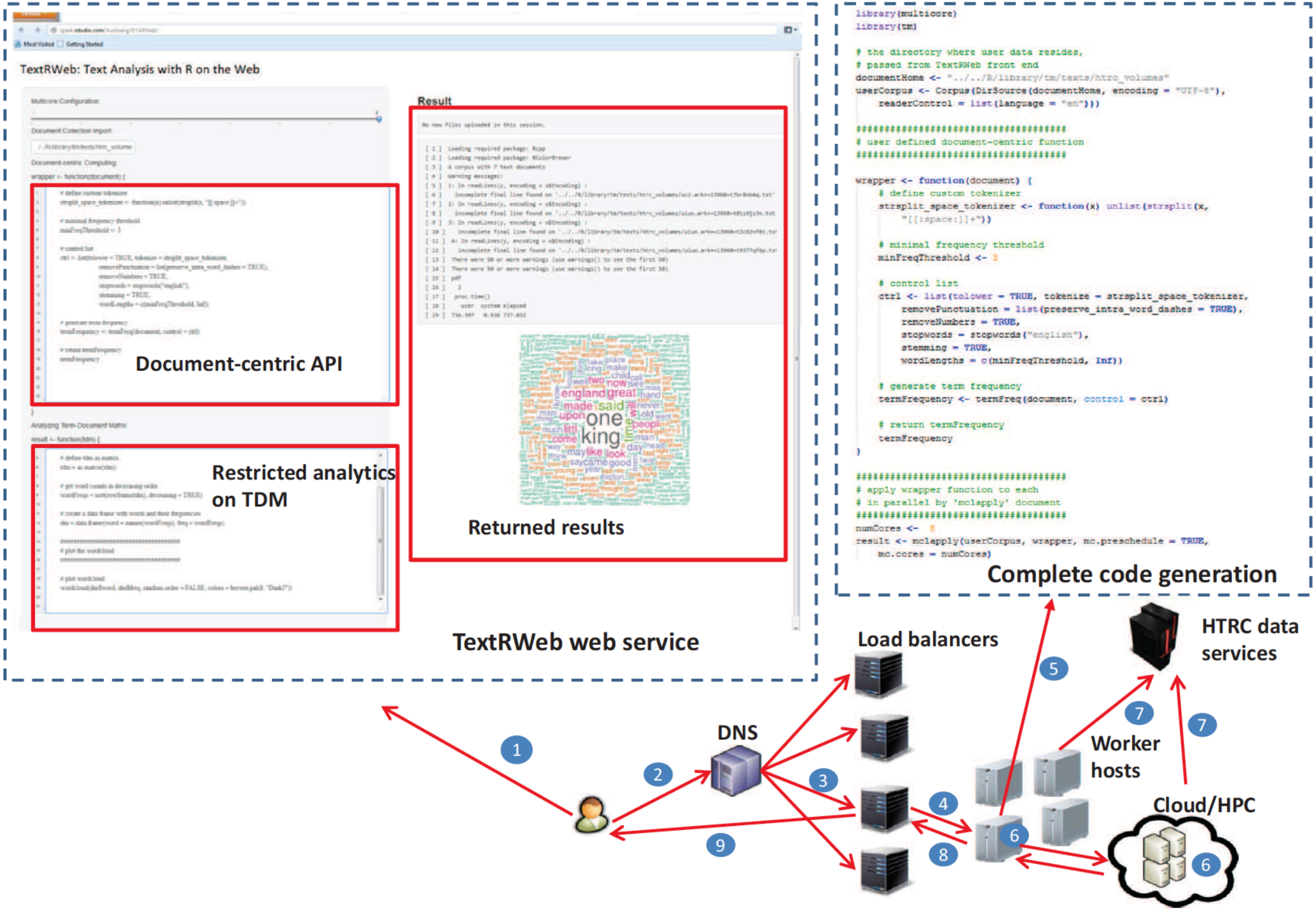
Value of using Airavata

- Enable collection of resources
- Application centric not compute centric
- Meta workflow to enable set of applications

Use-case for Data Analysis

- TextRWeb: Large Scale Text Analytics with R on the web

Collaborator: Hui Zhang, Data Scientist at Indiana University



TextRWeb: Text Analysis with R on the Web

Multicore Configuration:

Document Collection import:

../../R/library/tm/texts/htrc_volume

Document-centric Computing:

```
wrapper <- function(document) {
    # define custom tokenizer
    strsplit_space_tokenizer <- function(x) unlist(strsplit(x, "[[:space:]]+"))

    # minimal frequency threshold
    minFreqThreshold <- 3

    # control list
    ctrl <- list(tolower = TRUE, tokenize = strsplit_space_tokenizer,
                 removePunctuation = list(preserve_intra_word_dashes = TRUE),
                 removeNumbers = TRUE,
                 stopwords = stopwords("english"),
                 stemming = TRUE,
                 wordLengths = c(minFreqThreshold, Inf))

    # generate term frequency
    termFrequency <- termFreq(document, control = ctrl)

    # return termFrequency
    termFrequency
}
```

Analyzing Term-Document Matrix:

```
result <- function(tdm) {
    # define tdm as matrix
    tdm = as.matrix(tdm)

    # get word counts in decreasing order
    wordFreqs = sort(rowSums(tdm), decreasing = TRUE)

    # create a data frame with words and their frequencies
    dm = data.frame(word = names(wordFreqs), freq = wordFreqs)

    #####################################
    # plot the wordcloud
    #####################################

    # plot wordcloud
    wordcloud(dm$word, dm$freq, random.order = FALSE, colors = brewer.pal(8, "Dark2"))
```

Result

```
No new files uploaded in this session.
```

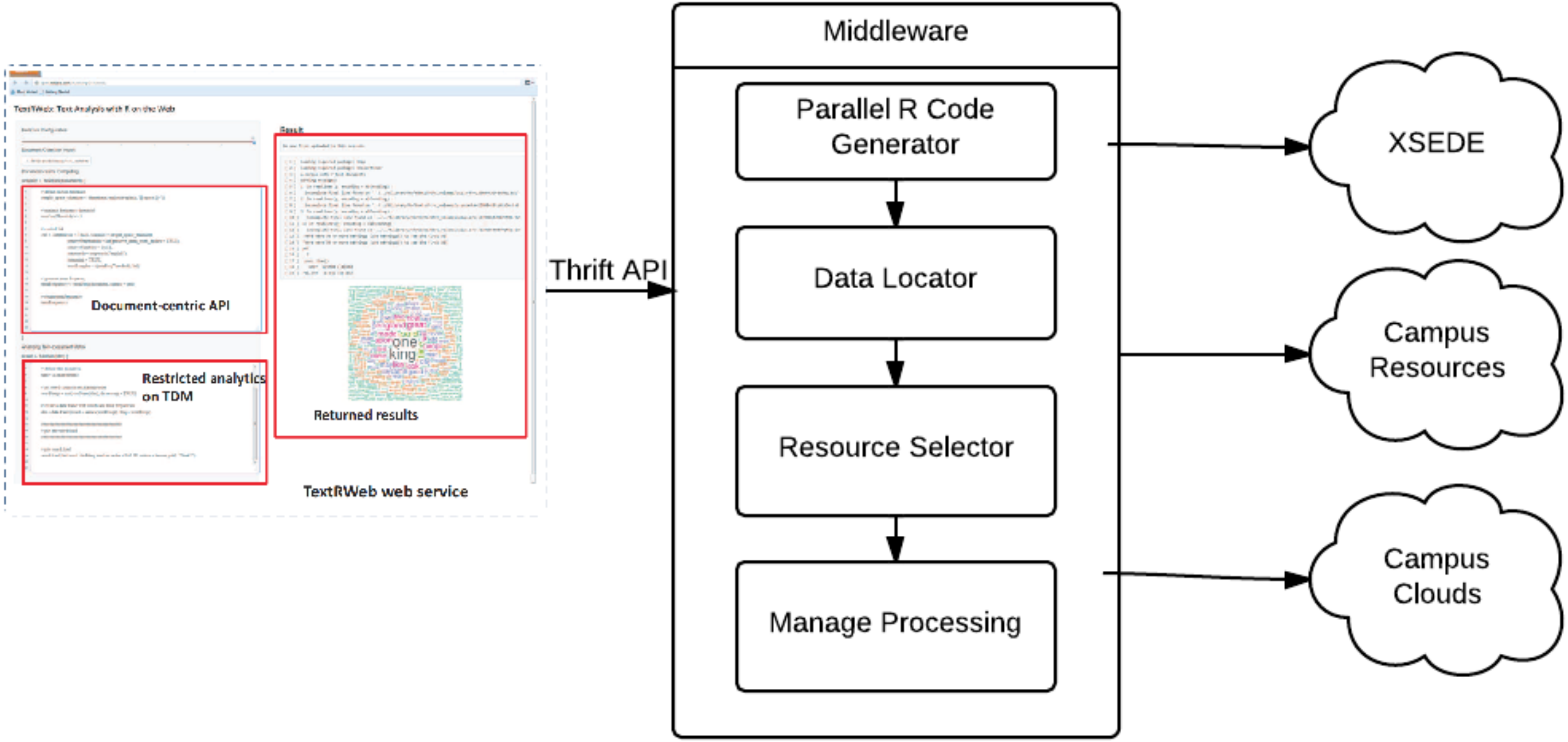
```
[ 1 ]  Loading required package: Rcpp
[ 2 ]  Loading required package: RColorBrewer
[ 3 ]  A corpus with 7 text documents
[ 4 ]  Warning messages:
[ 5 ]  1: In readLines(y, encoding = x$Encoding) :
[ 6 ]    incomplete final line found on '../../R/library/tm/texts/htrc_volumes/uc2.ark+=13960=t7br8nb4q.txt'
[ 7 ]  2: In readLines(y, encoding = x$Encoding) :
[ 8 ]    incomplete final line found on '../../R/library/tm/texts/htrc_volumes/uiuo.ark+=13960=t01z4js3n.txt
[ 9 ]  3: In readLines(y, encoding = x$Encoding) :
[ 10 ]    incomplete final line found on '../../R/library/tm/texts/htrc_volumes/uiuo.ark+=13960=t2c82nf0z.txt
[ 11 ]  4: In readLines(y, encoding = x$Encoding) :
[ 12 ]    incomplete final line found on '../../R/library/tm/texts/htrc_volumes/uiuo.ark+=13960=t9377qf6p.txt
[ 13 ]  There were 50 or more warnings (use warnings() to see the first 50)
[ 14 ]  There were 50 or more warnings (use warnings() to see the first 50)
[ 15 ]  pdf
[ 16 ]    2
[ 17 ]   proc.time()
[ 18 ]     user  system elapsed
[ 19 ]  736.397   0.936 737.032
```

```
library(multicore)
library(tm)

# the directory where user data resides,
# passed from TextRWeb front end
documentHome <- "../../R/library/tm/texts/htrc_volumes"
userCorpus <- Corpus(DirSource(documentHome, encoding = "UTF-8"),
    readerControl = list(language = "en")))

####################################
# user defined document-centric function
####################################

wrapper <- function(document) {
    # define custom tokenizer
    strsplit_space_tokenizer <- function(x) unlist(strsplit(x,
        "[[:space:]]+"))

    # minimal frequency threshold
    minFreqThreshold <- 

    # control list
    ctrl <- list(tolower = TRUE, tokenize = strsplit_space_tokenizer,
        removePunctuation = list(preserve_intra_word_dashes = TRUE),
        removeNumbers = TRUE,
        stopwords = stopwords("english"),
        stemming = TRUE,
        wordLengths = c(minFreqThreshold, Inf))

    # generate term frequency
    termFrequency <- termFreq(document, control = ctrl)

    # return termFrequency
    termFrequency
}

####################################
# apply wrapper function to each
# in parallel by 'mclapply' document
####################################
numCores <- 
result <- mclapply(userCorpus, wrapper, mc.preschedule = TRUE,
    mc.cores = numCores)
```

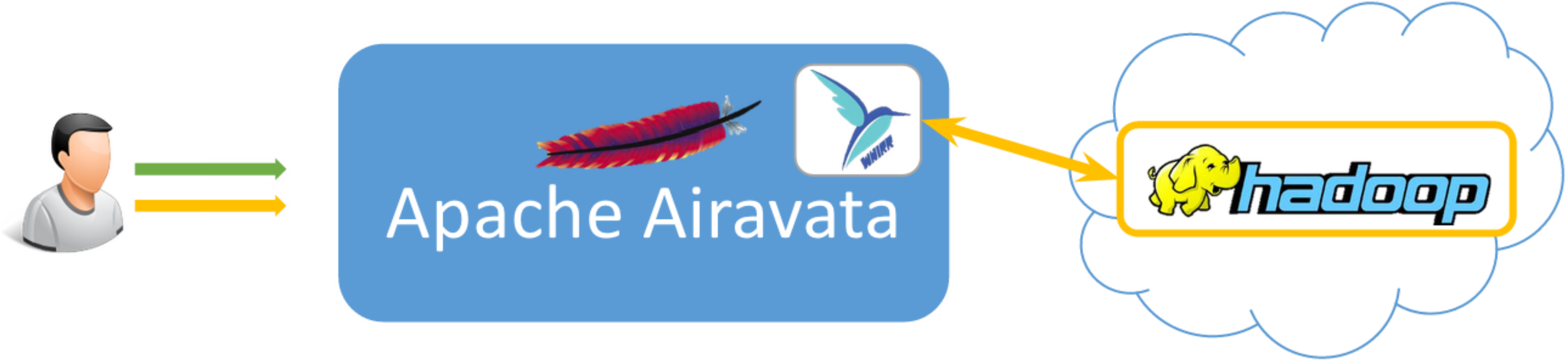
Goals for R on the web project

- Run large scale text analysis using parallel R.
- Hide computational complexity with user interfaces
- Support interactive text analysis
- Support iterative text mining

TextR Solution Diagram



Current Hadoop Integration



Computational Resources

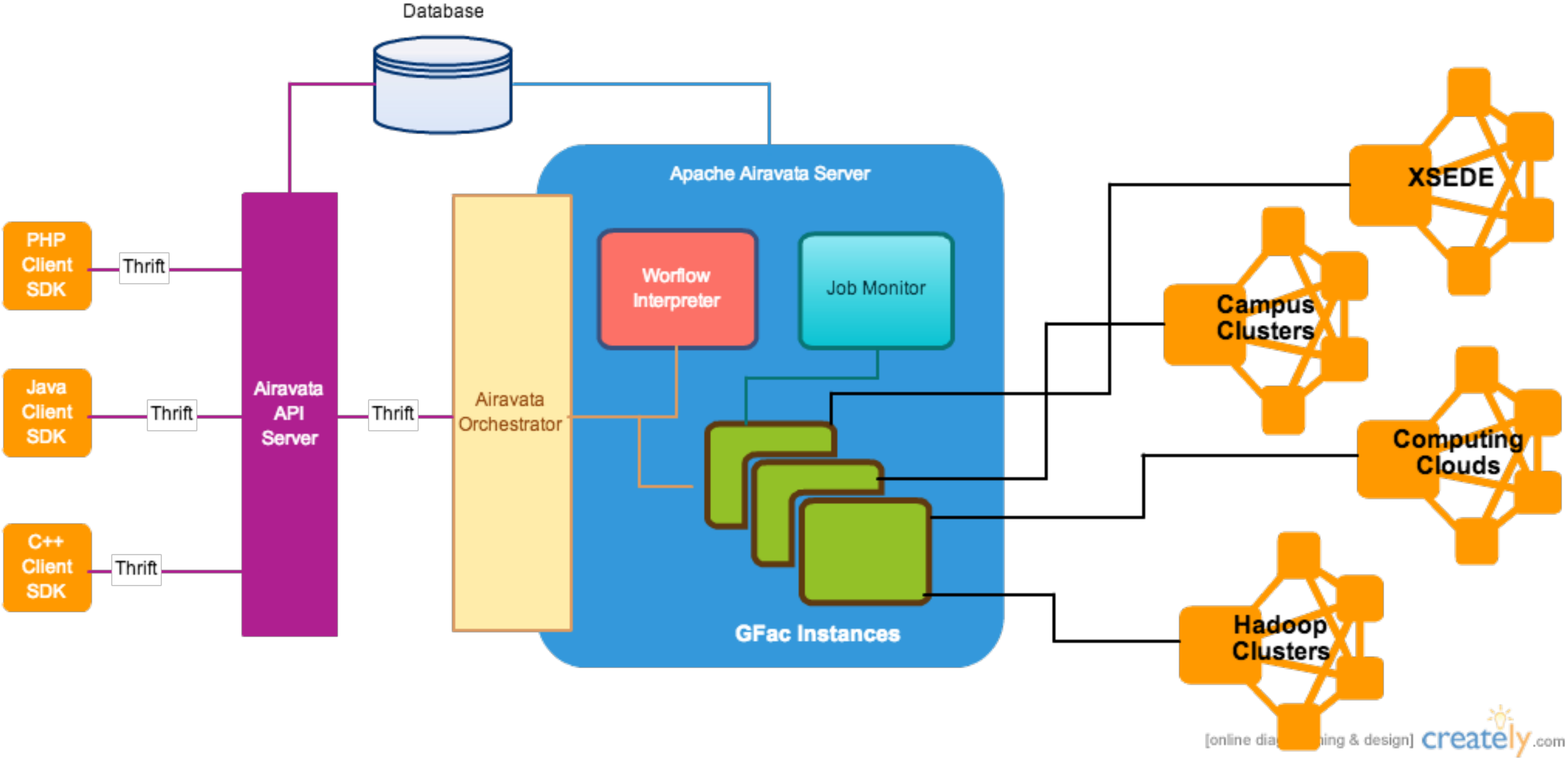
Future Work

- Integrate TextRWeb with Apache Spark
- Explore SparkR [1]
- Develop Apache Thrift interfaces for TextRWeb server
- Integrate with Apache Airavata for HPC job.
- Explore workflow DAGs for Text Analysis
- Keep updated with product offering like Stratosphere

1. https://github.com/amplab-extras/SparkR-pkg

Apache Spark

- In Memory computations
- Machine learning library (MLLib)
- graph engine (GraphX)
- Streaming analytics engine (Spark Streaming)
- Fast interactive query tool (Shark).
- Use Lineage data for fault tolerance
    - Tracking the data path



Database

PHP Client SDK

Java Client SDK

C++ Client SDK

Thrift

Thrift

Thrift

Airavata API Server

Thrift

Airavata Orchestrator

Apache Airavata Server

Worflow Interpreter

Job Monitor

GFac Instances

XSEDE

Campus Clusters

Computing Clouds

Hadoop Clusters

Conclusion

- Value added for the scientific communities
- Value for Apache Big Data Suite

Q & A

airavata.apache.org

Subscribe: users-subscribe@airavata.apache.org
Subscribe: dev-subscribe@airavata.apache.org
Subscribe: architecture-subscribe@airavata.apache.org

Thanks You!