


The Download: Community Tech Talks Episode 1

January 12, 2017



Welcome!

- Please share: Let others know you are here with #HPCCTechTalks 
- Ask questions! We will answer as many questions as we can following each speaker.
- Look for polls at the bottom of your screen. Exit full-screen mode or refresh your screen if you don't see them.
- We welcome your feedback - please rate us before you leave today and visit our [blog](#) for information after the event.
- Want to be one of our featured speakers? Let us know! techtalks@hpccsystems.com

Today's Presenters



Dr. Flavio Villanustre

VP Technology, LexisNexis® Risk Solutions

Flavio.Villanustre@lexisnexis.com

Dr. Flavio Villanustre leads HPCC Systems®, and is also VP, Technology for LexisNexis Risk Solutions. In this position, he is responsible for Information and Physical Security, overall HPCC Systems® platform strategy and new product development.

Flavio has been involved with the open source community for over 15 years through multiple initiatives. Some of these include founding the first Linux User Group in Buenos Aires (BALUG) in 1994, releasing several pieces of software under different open source licenses, and evangelizing open source to different audiences through conferences, training and education. Prior to Flavio's technology career, he was a neurosurgeon.



Anirudh Shah

Founder & CTO, 3LOQ Labs

anirudh@3loq.com

Anirudh Shah is the Founder & CTO, 3LOQ Labs. He is currently working on his second startup and has more than a decade of experience in machine learning, natural language processing, mobile software development and management. Anirudh has been using HPCC Systems for the past four years.

3LOQ meshes proprietary Advanced Computing techniques with Human Wisdom to produce Actionable Insights at speed. These insights allow your brand to partner with the Right customer, at the right time, to fulfill their purchase intention. We create an UNIQUE CUSTOMER View by contextualizing billions of transaction data points. The customer can be viewed individually or in relation to other customers.

Today's Presenters



Allan Wrobel

Sr Software Engineer, LexisNexis® Risk Solutions

allan.wrobel@lexisnexis.com

Allan has spent his career working in the technology industry, (that's 1976!) and he has been working with Databases since the mid-eighties.

Allan has worked with LexisNexis Risk Solutions since 2011 and the inception of LexisNexis Risk Solutions in the UK. Initially working with Data Operations, Allan is now serves as an ECL developer on both Thor and ROXIE.



Lorraine Chapman

Consulting Business Analyst, LexisNexis® Risk Solutions

Lorraine.chapman@lexisnexis.com

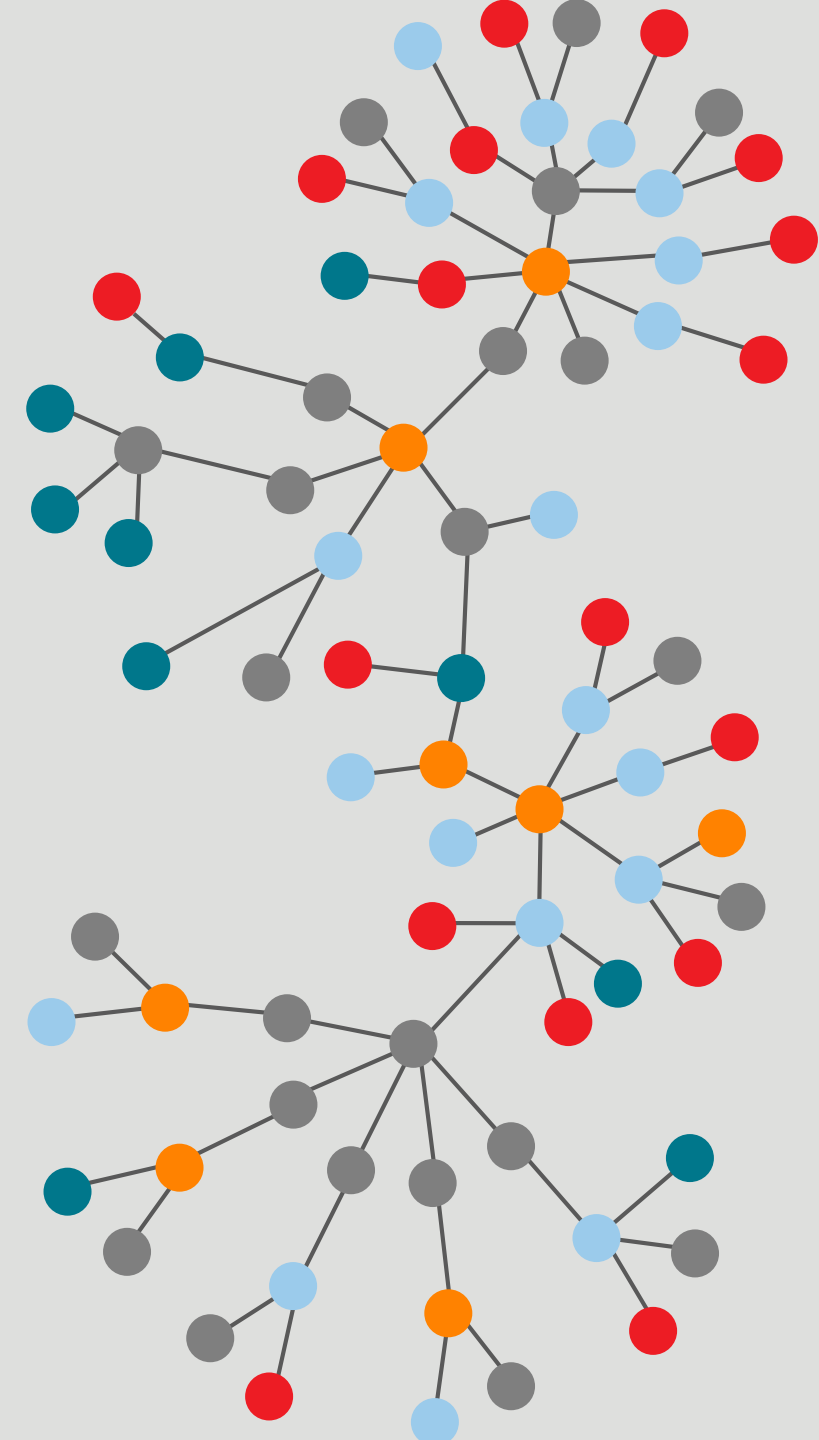
Lorraine has worked alongside software developers for over 20 years in a supportive role which has ranged from producing documentation including developing on-line help systems to software testing and release management.

Lorraine joined LexisNexis in 2004 and as well as continuing to work alongside the HPCC Systems platform development team, also administers the HPCC Systems Intern Program and manages our application to be an accepted organization for Google Summer of Code.

Lorraine is an active blogger on our website covering a wide range of subjects from new release information, features and improvements and the work students have completed during their internships.

Quick poll: What frequency would you like for these Tech Talks?

See poll on bottom of presentation screen





Automation using ECL+Jinja2

Anirudh Shah
Founder & CTO, 3LOQ Labs

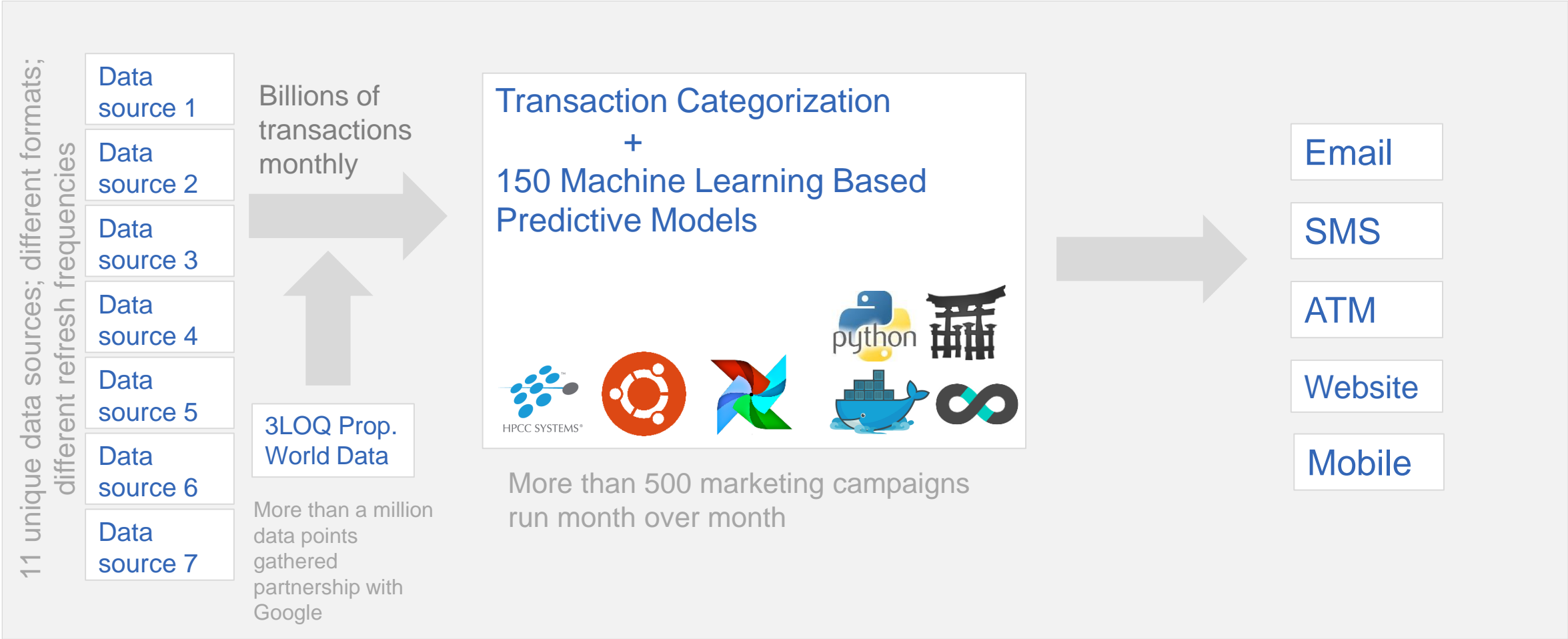


Quick poll: Have you used Jinja2?

See poll on bottom of presentation screen



What Does 3LOQ Do?



Challenges That We Faced With Campaign Automation

- Use case stats:
 - 4TB of data processed (450 datasets)
 - 400K lines of ECL
 - 500 campaigns (150 ML models)
 - HW setup: 9+1 commodity desktops
 - Turn Around Time: 48 hrs
- Large number of datasets have to be checked and sanitized monthly
- Application of long cleaning and ETL pipelines to all datasets
- Automation of all activities

Why Use Jinja2?

- Jinja2 is a Python based templating library
- ECL code generation leveraging the full power of Python
- Uses:
 - Iterating during feature generation
 - Hyper-parameter tuning
 - Freezing ECL code and allowing for parametrization
 - Enable automation by reading parameters from a database
 - Avoid use of To/From fields in ECL for very large tables

How To Use Jinja2

1. Create a template file using ECL syntax
2. Replace all the 'parameters' with placeholders.
3. Populate the template using a python script. (From CSV or from DB or from a configuration file)
4. Run the generated ECL code.

Jinja 2 Example1

Base ECL

```
import STD;  
STD.File.VerifyFile('file1', TRUE)
```



After Parametrization

```
/*A template to call STD.File.VerifyFile() for a list of files.*/  
import STD;  
{% for row in paramDict %}  
  STD.File.VerifyFile('{{ row['name'] }}', TRUE)  
{% endfor %}
```



After Populating (Given a list of files)

```
/*A template to call STD.File.VerifyFile() for a list of files.*/  
import STD;  
STD.File.VerifyFile('~test::file1', TRUE)  
STD.File.VerifyFile('~test::file2', TRUE)  
STD.File.VerifyFile('~test::file3', TRUE)  
STD.File.VerifyFile('~test::file4', TRUE)  
.  
.  
.
```

Python Code For Template Inflation

```
import os
import sys
import csv
import jinja2

def apply_template_fromCSV(templateFilename, templateParamsCSVFilename, outputFilename, generatorScript='expander'):
    with open(templateParamsCSVFilename, mode='r') as paramFile:
        paramDict = csv.DictReader(paramFile)
        print paramDict
        inflatedTemplate = apply_template(templateFilename, params={'paramDict':sorted(paramDict), 'generatorScript':generatorScript})
        print inflatedTemplate
        with open(outputFilename, mode='w') as outfile:
            outfile.write(inflatedTemplate)

if __name__ == '__main__':
    if len(sys.argv) <> 3:
        print 'Usage: python expand.py <template_file> <csv_param_file>'
        sys.exit(-1)
    template_file = sys.argv[1]
    param_file = sys.argv[2]
    out_file = os.path.splitext(template_file)[0]
    apply_template_fromCSV(template_file, param_file, out_file)
```

Jinja Example 2: Parameterization

Base ECL

```
import master_date_set;
import DatasetCompare;

DatasetCompare.column_frequency(row_1 , recordof(row_1), OutDsFreq_row_1);

OUTPUT(OutDsFreq_row_1, NAMED('row_1'));
OUTPUT(OutDsFreq_table1,, '~datasetsanity::frequency::set1_'+master_date_set.end_date1, csv(heading(single), separator('|')), overwrite);
```



After Parametrization

```
import master_date_set;
import DatasetCompare;

{% for row in outputTableParams %}
    DatasetCompare.column_frequency({{ row.OutputTableHandle }} , recordof({{ row.OutputTableHandle }}), OutDsFreq_{{ row.OutputTableHandle }});
    OUTPUT(OutDsFreq_{{ row.OutputTableHandle }}, NAMED('{{ row.OutputTableHandle }}'));
    OUTPUT(OutDsFreq_{{ row.OutputTableHandle }}, '~datasetsanity::frequency::{{ row.OutputTableClusterName }}'+master_date_set.end_date1, csv(heading(single), separator('|')), overwrite);
{% endfor %}
```

Jinja Example 2: After Template Inflation

```
import master_date_set;
import DatasetCompare;

DatasetCompare.column_frequency(row_1 , recordof(row_1), OutDsFreq_row_1);
OUTPUT(OutDsFreq_row_1, NAMED('row_1'));
output(OutDsFreq_table1,, '~datasetsanity::frequency::set1_'+master_date_set.end_date1, csv(heading(single), separator('|')), overwrite);

DatasetCompare.column_frequency(row_2 , recordof(row_2), OutDsFreq_row_2);
OUTPUT(OutDsFreq_row_2, NAMED('row_2'));
OUTPUT(OutDsFreq_table2,, '~datasetsanity::frequency::set1_'+master_date_set.end_date1, csv(heading(single), separator('|')), overwrite);

DatasetCompare.column_frequency(row_3 , recordof(row_3), OutDsFreq_row_3);
OUTPUT(OutDsFreq_row_3, NAMED('row_3'));
OUTPUT(OutDsFreq_table3,, '~datasetsanity::frequency::set2_'+master_date_set.end_date1, csv(heading(single), separator('|')), overwrite);
.
.
.
```


Benefits

- Reduction in ECL complexity by avoiding nested MACRO loops
- Ability to automate complex workflows
- Leverage Python's vast ecosystem of libraries for data/file manipulation
- Easy to use

Jinja2 Reference Links

Home Page: <http://jinja.pocoo.org/docs/2.9/>

Wiki: [https://en.wikipedia.org/wiki/Jinja_\(template_engine\)](https://en.wikipedia.org/wiki/Jinja_(template_engine))

Python package index: <https://pypi.python.org/pypi/Jinja2>

Using filters: http://docs.ansible.com/ansible/playbooks_filters.html

Quick poll: Would using Jinja2 and ECL together help you?

See poll on bottom of presentation screen

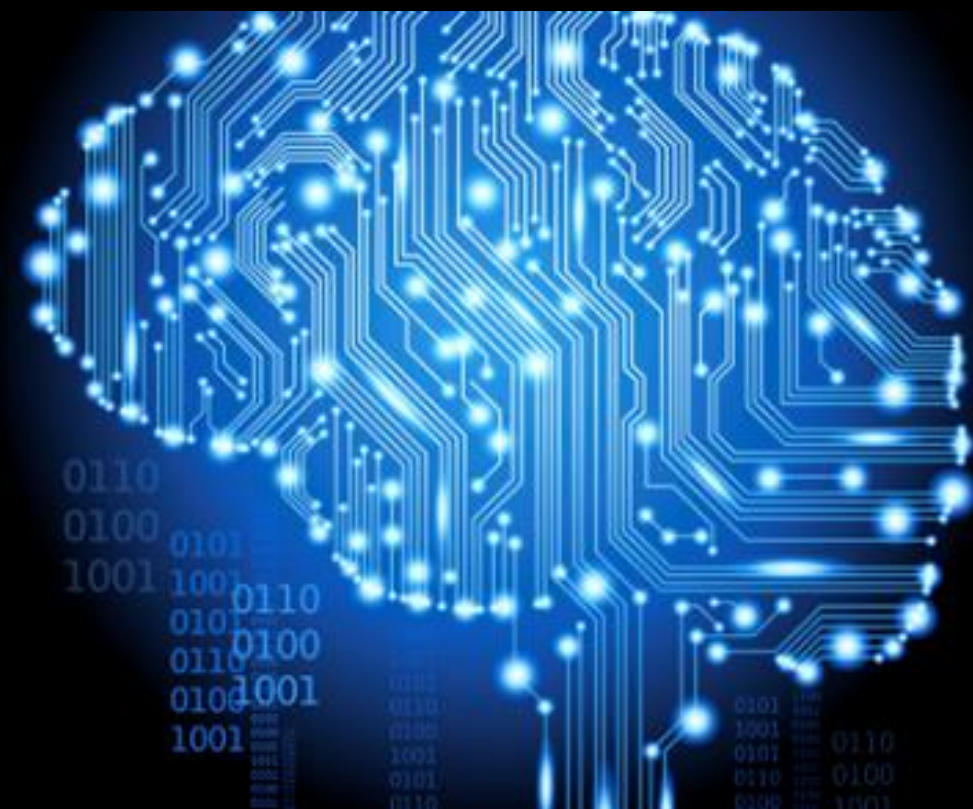


Questions?



Anirudh Shah
Founder & CTO, 3LOQ Labs
anirudh@3loq.com





Leveraging Superfiles on Thor

Allan Wrobel
Sr Software Engineer, LexisNexis







Monolithic Logical Files

- Daily updates mean daily copies of data that is virtually identical day in day out.
- Date bounded Queries read the entire dataset, when many such queries are for recent data, or tightly bounded data.
- Archiving off historic data requires a read and write of the entire dataset.







Coordinating Multiple Logical Files

- Perhaps best illustrated by example










Day 1

Logical Name	Owner	Description	Cluster	Records	Size	Parts
 publicrecords::base::current	awrobel			5	10	
 publicrecords::base::current:20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::father						
 publicrecords::base::grandfather						













Day 2

Logical Name	Owner	Description	Cluster	Records	Size	Parts
 publicrecords::base::current	awrobel			13	26	
 publicrecords::base::current::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::current::20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::father	awrobel			5	10	
 publicrecords::base::father::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::grandfather						
















Day 3

Logical Name	Owner	Description	Cluster	Records	Size	Parts
 publicrecords::base::current	awrobel			24	48	
 publicrecords::base::current::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::current::20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::current::20161129::001	awrobel		Cluster_1	11	22	20
 publicrecords::base::father	awrobel			13	26	
 publicrecords::base::father::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::father::20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::grandfather	awrobel			5	10	
 publicrecords::base::grandfather::20161127::001	awrobel		Cluster_1	5	10	20












Day 3 : Second Build of the Day

Logical Name	Owner	Description	Cluster	Records	Size	Parts
 publicrecords::base::current	awrobel			28	56	
 publicrecords::base::current::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::current::20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::current::20161129::001	awrobel		Cluster_1	11	22	20
 publicrecords::base::current::20161129::002	awrobel		Cluster_1	4	8	20
 publicrecords::base::father	awrobel			24	48	
 publicrecords::base::father::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::father::20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::father::20161129::001	awrobel		Cluster_1	11	22	20
 publicrecords::base::grandfather	awrobel			13	26	
 publicrecords::base::grandfather::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::grandfather::20161128::001	awrobel		Cluster_1	8	16	20











Day 4

Logical Name	Owner	Description	Cluster	Records	Size	Parts
 publicrecords::base::current	awrobel			33	66	
 publicrecords::base::current:20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::current:20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::current:20161129::001	awrobel		Cluster_1	11	22	20
 publicrecords::base::current:20161129::002	awrobel		Cluster_1	4	8	20
 publicrecords::base::current:20161130::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::father	awrobel			28	56	
 publicrecords::base::father:20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::father:20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::father:20161129::001	awrobel		Cluster_1	11	22	20
 publicrecords::base::father:20161129::002	awrobel		Cluster_1	4	8	20
 publicrecords::base::grandfather	awrobel			24	48	
 publicrecords::base::grandfather:20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::grandfather:20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::grandfather:20161129::001	awrobel		Cluster_1	11	22	20










Day 4 : End of Month Process

Logical Name	Owner	Description	Cluster	Records	Size	Parts
 publicrecords::base::current	awrobel			33	66	
 publicrecords::base::current::201611	awrobel		Cluster_1	33	66	20
 publicrecords::base::father	awrobel			28	56	
 publicrecords::base::father::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::father::20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::father::20161129::001	awrobel		Cluster_1	11	22	20
 publicrecords::base::father::20161129::002	awrobel		Cluster_1	4	8	20
 publicrecords::base::grandfather	awrobel			24	48	
 publicrecords::base::grandfather::20161127::001	awrobel		Cluster_1	5	10	20
 publicrecords::base::grandfather::20161128::001	awrobel		Cluster_1	8	16	20
 publicrecords::base::grandfather::20161129::001	awrobel		Cluster_1	11	22	20

Day 5 : First Day of Month

Logical Name	Owner	Description	Cluster	Records	Size
 publicrecords::base::current	awrobel			40	80
 publicrecords::base::current::201611	awrobel		Cluster_1	33	66
 publicrecords::base::current::20161201::001	awrobel		Cluster_1	7	14
 publicrecords::base::father	awrobel			33	66
 publicrecords::base::father::201611	awrobel		Cluster_1	33	66
 publicrecords::base::grandfather	awrobel			28	56
 publicrecords::base::grandfather::20161127::001	awrobel		Cluster_1	5	10
 publicrecords::base::grandfather::20161128::001	awrobel		Cluster_1	8	16
 publicrecords::base::grandfather::20161129::001	awrobel		Cluster_1	11	22
 publicrecords::base::grandfather::20161129::002	awrobel		Cluster_1	4	8

Day 6

Logical Name	Owner	Description	Cluster	Records	Size	Parts
 publicrecords::base::current	awrobel			49	98	
 publicrecords::base::current::201611	awrobel		Cluster_1	33	66	20
 publicrecords::base::current::20161201::001	awrobel		Cluster_1	7	14	20
 publicrecords::base::current::20161202::001	awrobel		Cluster_1	9	18	20
 publicrecords::base::father	awrobel			40	80	
 publicrecords::base::father::201611	awrobel		Cluster_1	33	66	20
 publicrecords::base::father::20161201::001	awrobel		Cluster_1	7	14	20
 publicrecords::base::grandfather	awrobel			33	66	
 publicrecords::base::grandfather::201611	awrobel		Cluster_1	33	66	20

Logical File Selection

- LogicalFileSubSet(STRING8 dateFrom,STRING8 dateTo);
- LogicalFileSubSetDelta(INTEGER daysback,STRING8 dateTo);
- LogicalFileSubSetN(UNSIGNED n);
- ReverseLFSubSetN(UNSIGNED n);

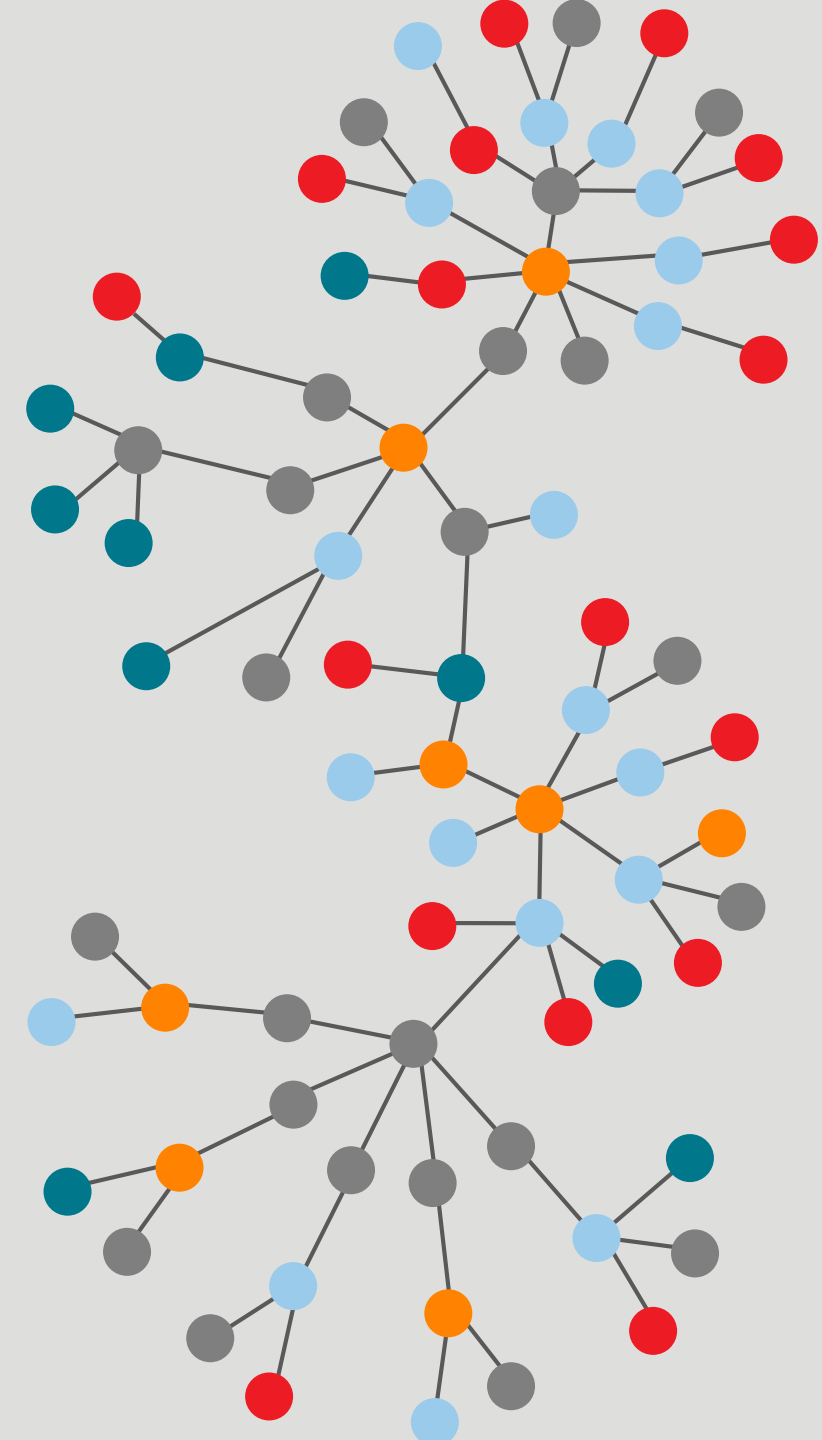
Benefits

- Build times reduced by orders of magnitude. (The bigger the data the greater the improvement)
- Allowed ad-hoc searches of very large un-indexed data. (Where the search is date bounded)
- Archiving historic data becomes a trivial exercise that only uses the DFU server.

Quick poll: Do you already use
functionality like this?

If not, do you think this
functionality would be useful in
your business?

See poll on bottom of presentation screen



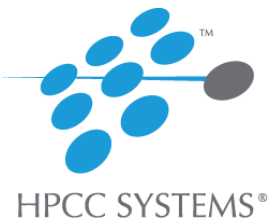
Questions?

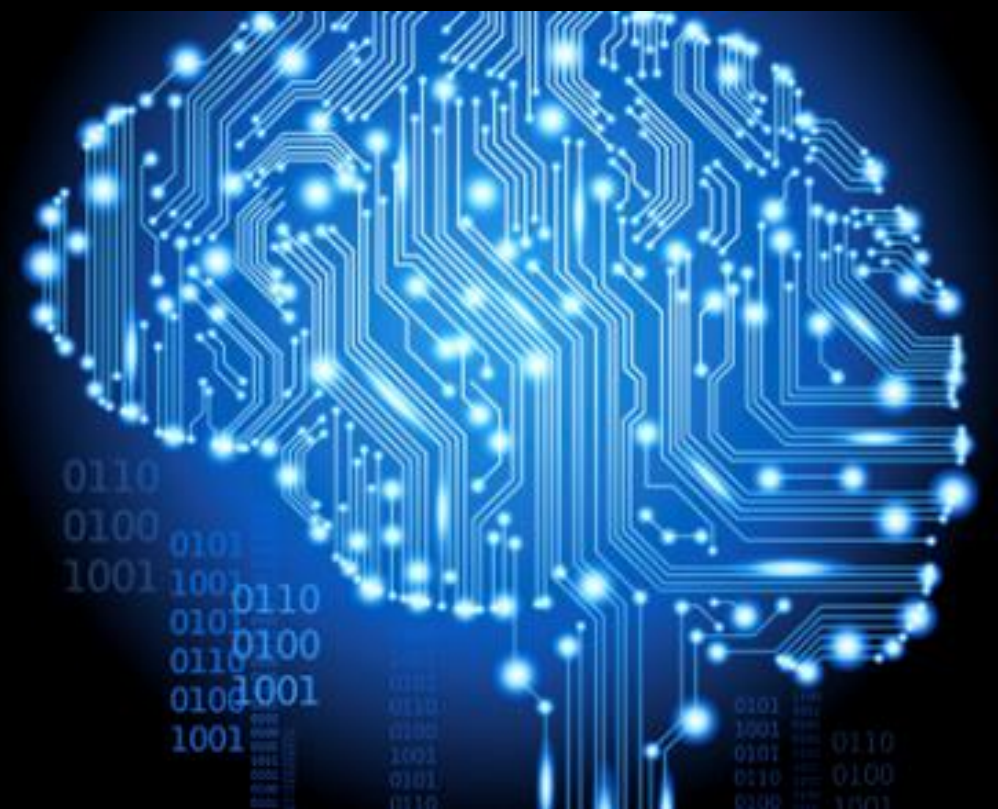


Allan Wrobel

Sr Software Engineer, LexisNexis® Risk Solutions

allan.wrobel@lexisnexis.com





Student Opportunities with HPCC Systems

Lorraine Chapman
Consulting Business Analyst

January 12, 2017



Available Student Opportunities

- Google Summer of Code
- HPCC Systems Summer Intern Program
- LexisNexis Corporate Intern Program

Program Similarities

- 10 week program over the summer recess
- Paid programs
- Evaluations
- Mentoring by a subject/department expert
- Weekly updates
- Final presentation
- Potential for presenting at the HPCC Systems Engineering Summit

Program Differences

- LexisNexis program is office based only
- Contact Renu.Midha@lexisnexis.com

GSoC and HPCC Systems Programs only...

- Proposal based
- Expect coding to start the first day
- Paid in 2 stipends
- Regular communication with mentor is required
- Blog journal

Google Summer of Code

- February 9th 2017 - Deadline for organization applications
- February 27th 2017 - Publication of accepted organizations
- March 20th -April 3rd 2017 - Student proposal period
- May 1st 2017 - Publication of accepted students
- May 1st - May 30th – Community bonding period
- May 30th – August 29th – Coding period

HPCC Systems Summer Intern Program

- Deadline for proposals – Monday 3rd April
- Contact the project mentor for support
- Submit a draft to the project mentor
- Final proposals to Lorraine.Chapman@lexisnexis.com
- Notification by email within 2 weeks
- Introductory meetup before coding begins

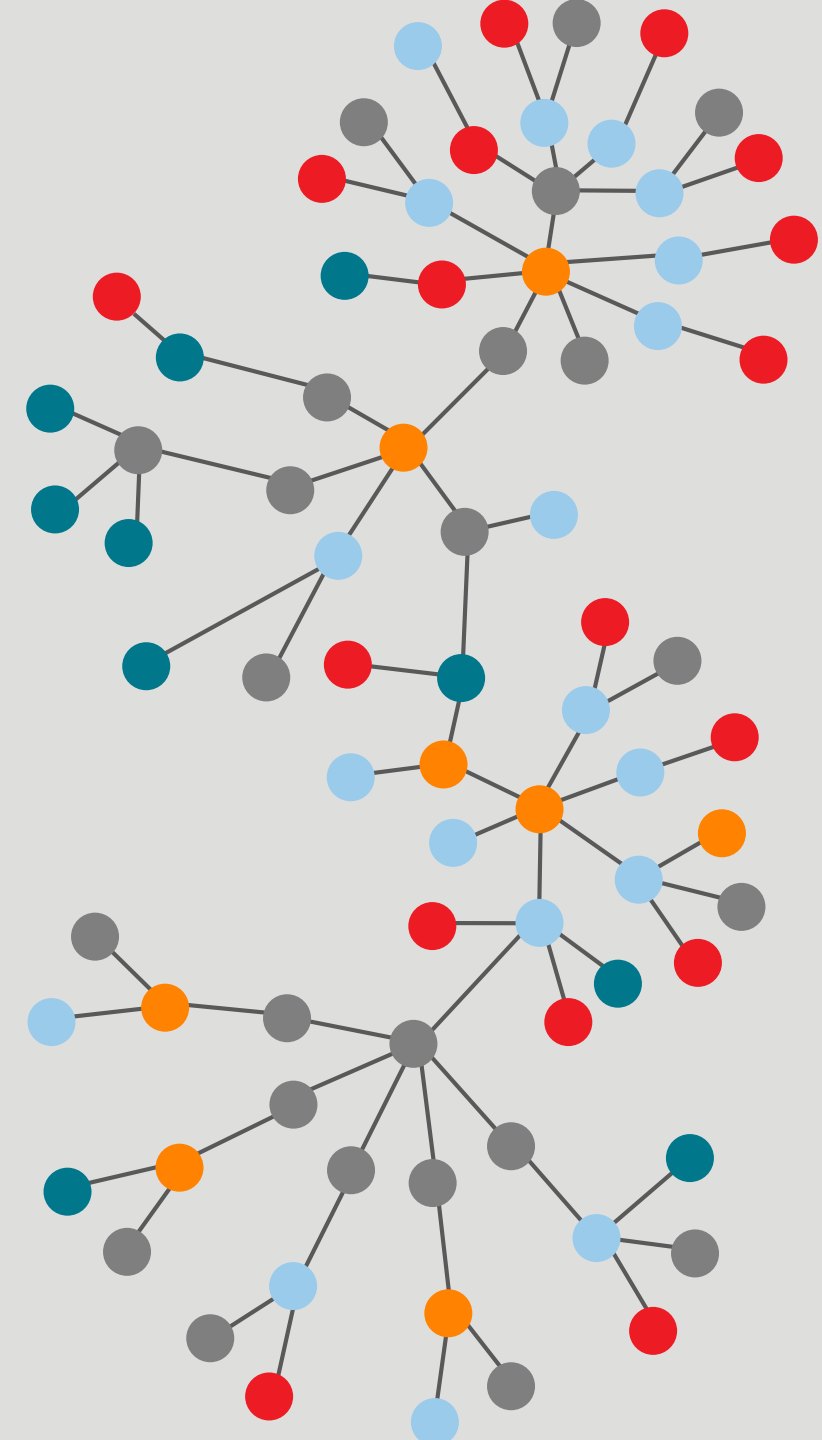
Student Successes

- 11 students over 2 years across all programs
- 4 returning students
- Employment possibilities
- Forward thinking projects
- Learning experience
- Enjoyable experience



Quick poll: Do you have a student project suggestion that would help improve your HPCC Systems experience?

See poll on bottom of presentation screen



Need More Information?

- Available projects: <https://wiki.hpccsystems.com/x/yIBc>
- Proposal help: <https://wiki.hpccsystems.com/x/SQB>
- Blogs: <http://bit.ly/1OJUIBT>
- GSoC 2015 completed projects: <https://wiki.hpccsystems.com/x/g4BR>
- 2016 Completed projects: <https://wiki.hpccsystems.com/x/nACC>
- HPCC Systems student Wikis: <https://wiki.hpccsystems.com>
- GSoC website: <https://developers.google.com/open-source/gsoc/>
- Keep in touch! Visit our Student Programs Forum: <http://bit.ly/2i8yIVh>

Contact Details

- LexisNexis Corporate Intern Program
Renu.Midha@lexisnexisrisk.com
- HPCC Systems Intern Program and Google Summer of Code
Lorraine.Chapman@lexisnexisrisk.com
- Project mentor contact details
See the project descriptions
- HPCC Systems Blog
<https://hpccsystems.com/resources/blog>
- Student and GSoC wikis:
<https://wiki.hpccsystems.com>

Questions?



Lorraine Chapman

Consulting Business Analyst, LexisNexis® Risk Solutions

Lorraine.chapman@lexisnexis.com

Submit a Talk for an Upcoming Episode!

- Have a new success story to share?
- Want to pitch a new use case?
- Have a new HPCC Systems application you want to demo?
- Want to share some helpful ECL tips and sample code?
- Have a new suggestion for the roadmap?
- Be a featured speaker for an upcoming episode! Email your idea to Techtalks@hpccsystems.com

Visit The Download Tech Talks wiki for more information:
<https://wiki.hpccsystems.com/display/hpcc/HPCC+Systems+Tech+Talks>

Thank You!



 **RELX** Group

A copy of this presentation will be made available soon on our blog:
hpccsystems.com/blog