Applications of the YarcData Urika in Drug Discovery and Healthcare

Robert Henschel – henschel@iu.edu
Manager, Scientific Applications and Performance Tuning

Research Technologies, UITS
Indiana University

May 7th, 2014

RESEARCH TECHNOLOGIES
INDIANA UNIVERSITY
University Information Technology Services

PERVASIVE TECHNOLOGY INSTITUTE
INDIANA UNIVERSITY

Contents

- Motivation and Introduction
- Use Case
- Hardware and Software
- Methodology and Results
- Conclusion

Acknowledgements

- Jim Maltby and Matt Gianni at YarcData
- Mario Vale at CSCS

Motivation and Introduction

- I am not a Urika or Semantic Web expert
- Learn more about the Urika
  - Other than marketing material
  - How can the machine be used
- Great partners
  - CCRG for a use case
  - YarcData and CSCS for testing

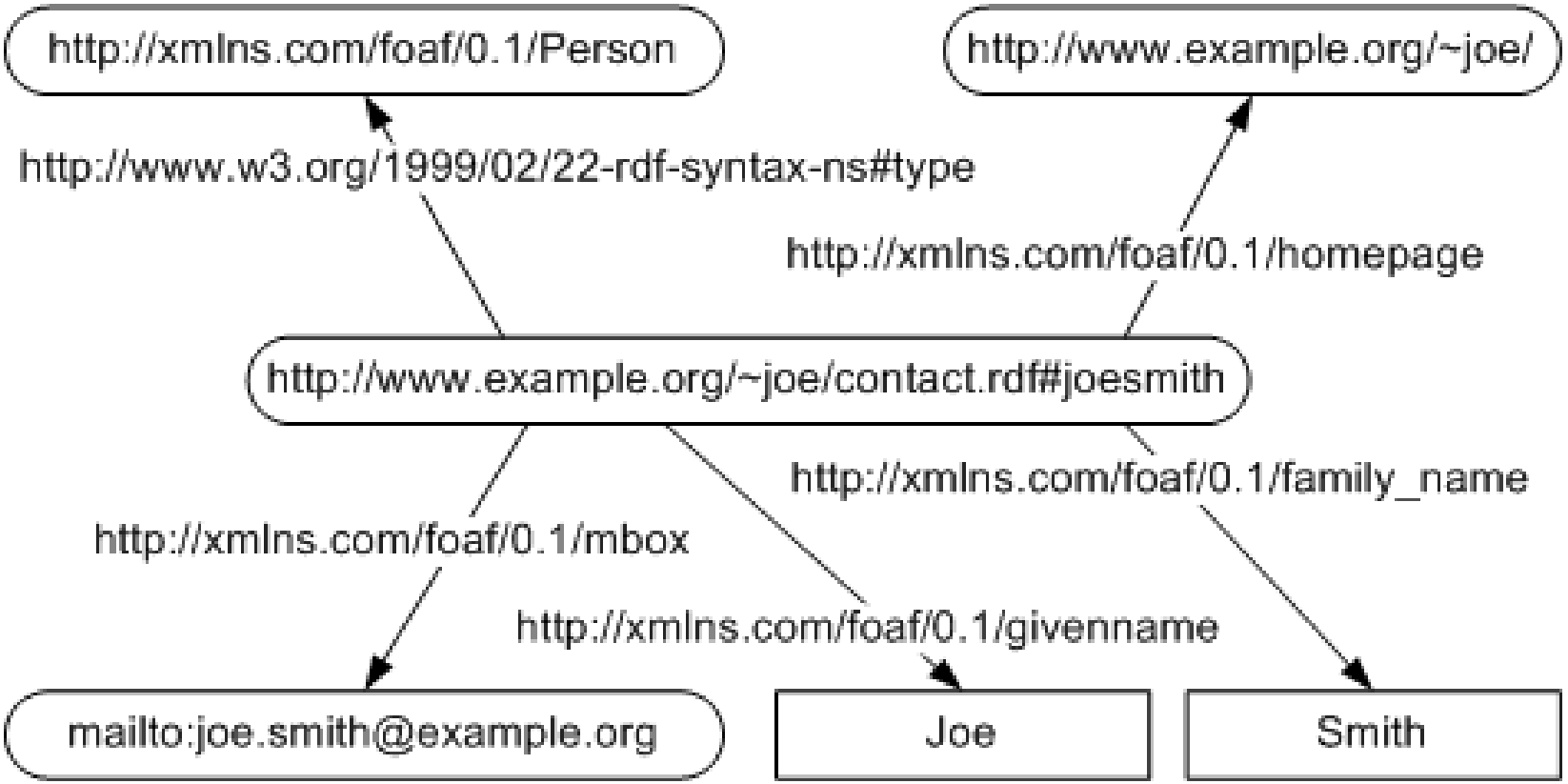
Motivation and Introduction

- Forget all about writing your own programs and get used to formulating your questions in SPARQL
- Data must be available in RDF triple form
- This is not a relational database, joins are not a problem
- Think of the Urika as an accelerator for SPARQL

Motivation and Introduction

- Semantic Web technologies
  - RDF
  - SPARQL
  - RDFS and OWL

RDF



(http://www.example.org/~joe/contact.rdf#joesmith, http://xmlns.com/foaf/0.1/family_name, "Smith")

Motivation and Introduction

- Semantic Web technologies
  - RDF
  - SPARQL
  - RDFS and OWL

SPARQL

Query:

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT *
WHERE {
     ?person foaf:family_name ?name .
}
```

Result:

```
http://www.example.org/~joe/contact.rdf#joesmith  Smith
```

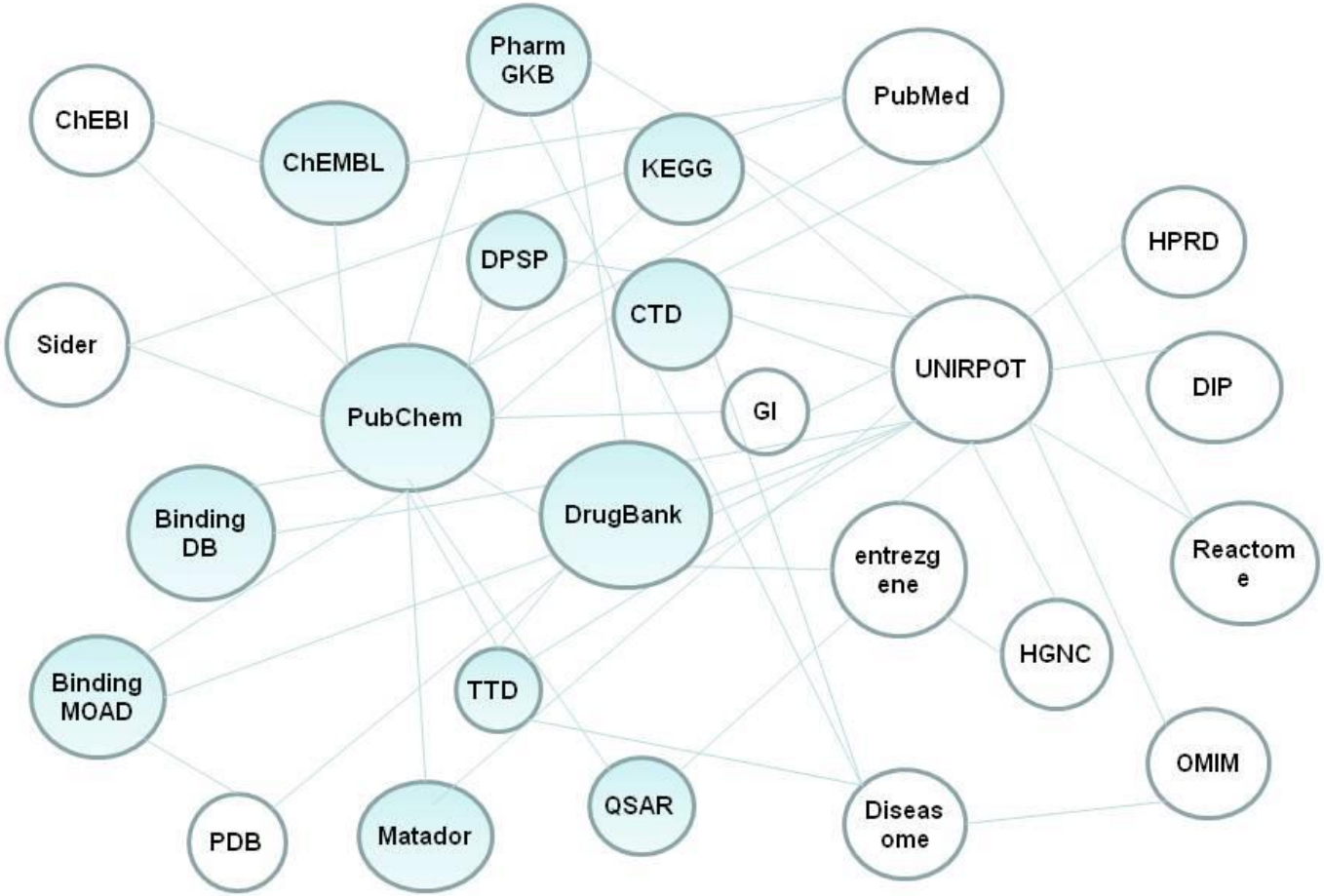
Motivation and Introduction

- Semantic Web technologies
  - RDF
  - SPARQL
  - RDFS and OWL

Use Case

- Drug – Drug interaction and side effects
- Chem2Bio2RDF
- Using public data sets

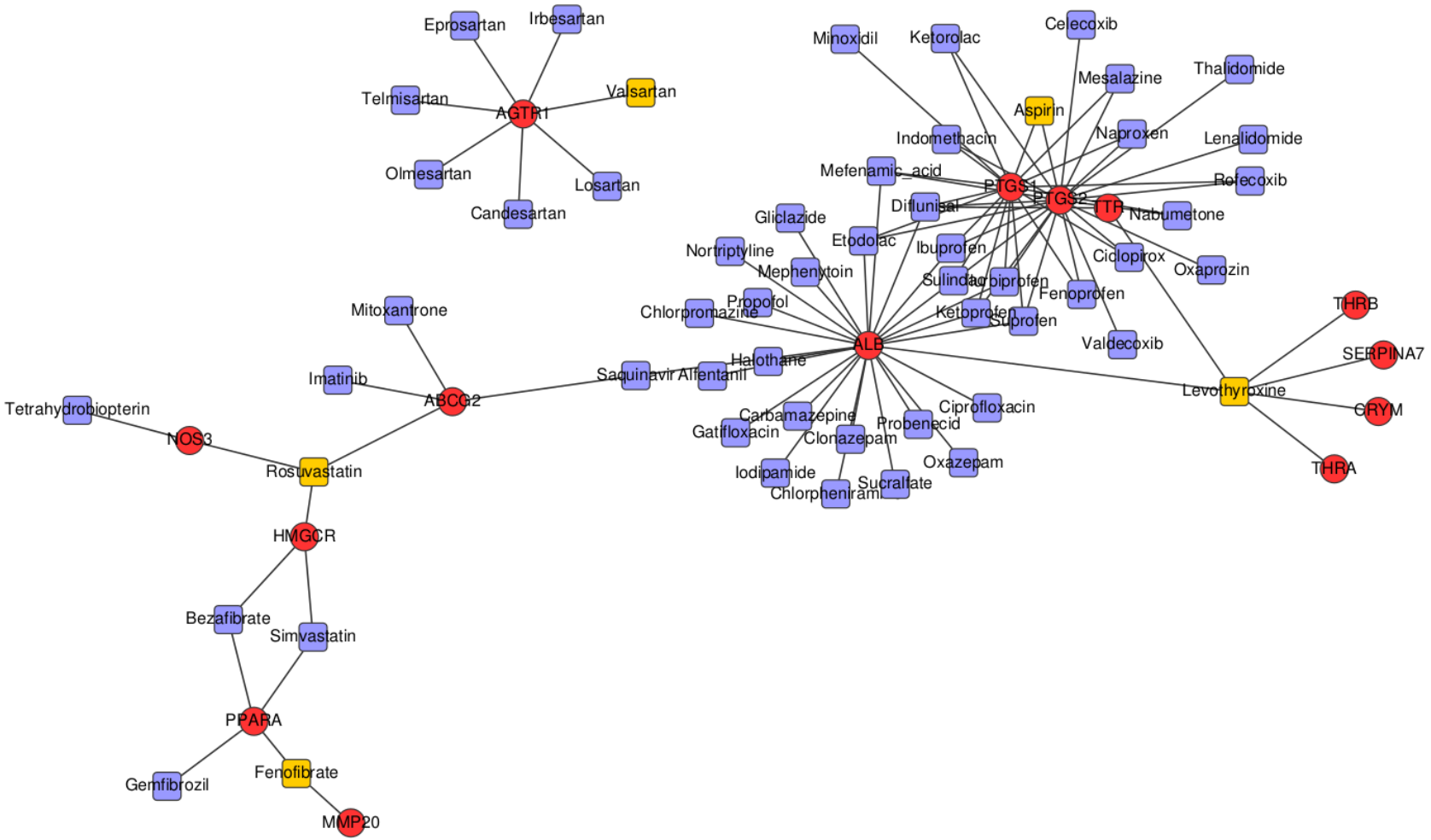
Chem2Bio2RDF Data Set



Example SPARQL Query

```
PREFIX sider:<http://chem2bio2rdf.org/sider/resource/>
SELECT * FROM <http://chem2bio2rdf.org/sider>
WHERE
{
?sider sider:drug_name ?drug_name .
FILTER
regex(?drug_name,"Fenofibrate|Aspirin|Rosuvastatin|Levothyroxine|Valsartan","i" ) .
?sider sider:side_effect ?side_effect .
}
```

Use Case


Eprosartan
Irbesartan
Telmisartan
AGTR1
Valsartan
Olmesartan
Losartan
Candesartan
Minoxidil
Ketorolac
Celecoxib
Mesalazine
Thalidomide
Aspirin
Indomethacin
Naproxen
Lenalidomide
Mefenamic_acid
PTGS1
PTGS2
TTR
Rofecoxib
Diflunisal
Nabumetone
Gliclazide
Etodolac
Ibuprofen
Nortriptyline
Cidlopirox
Oxaprozin
Mephenytoin
Sulindac
Ibuprofen
Fenoprofen
Propofol
Chlorpromazine
Ketoprofen
Suprofen
Valdecoxib
Mitoxantrone
THRB
SERPINA7
Halothane
Saquinavir
Alfentanil
Imatinib
ABCG2
Levothyroxine
GRYM
Tetrahydrobiopterin
Carbamazepine
Ciprofloxacin
Gatifloxacin
Probenecid
Clonazepam
NOS3
Rosuvastatin
Oxazepam
Iodipamide
Sucralfate
Chlorpheniramine
THRA
HMGCR
Bezafibrate
Simvastatin
PPARA
Fenofibrate
Gemfibrozil
MMP20

Contents

- Motivation and Introduction
- Use Case
- Hardware and Software
- Methodology and Results
- Conclusion

Hardware and Software

- Standard Server
- Urika-64

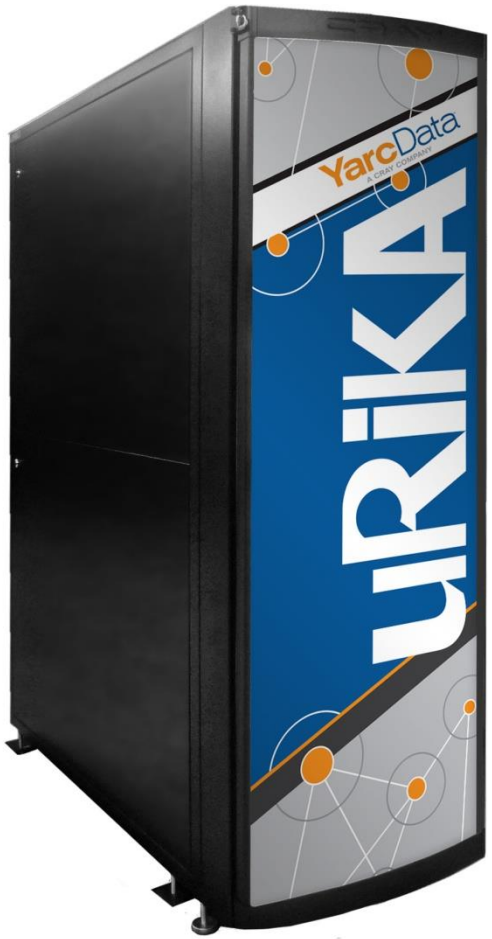
Standard Server – Hardware

- Dell PowerEdge R510
- Two quad-core Intel Xeon X5550, at 2.6 GHz
- 26 GB of memory

Standard Server – Software

- RHEL 5.1
- Virtuoso 06.01.3127-pthreads

Urika-64 – Hardware



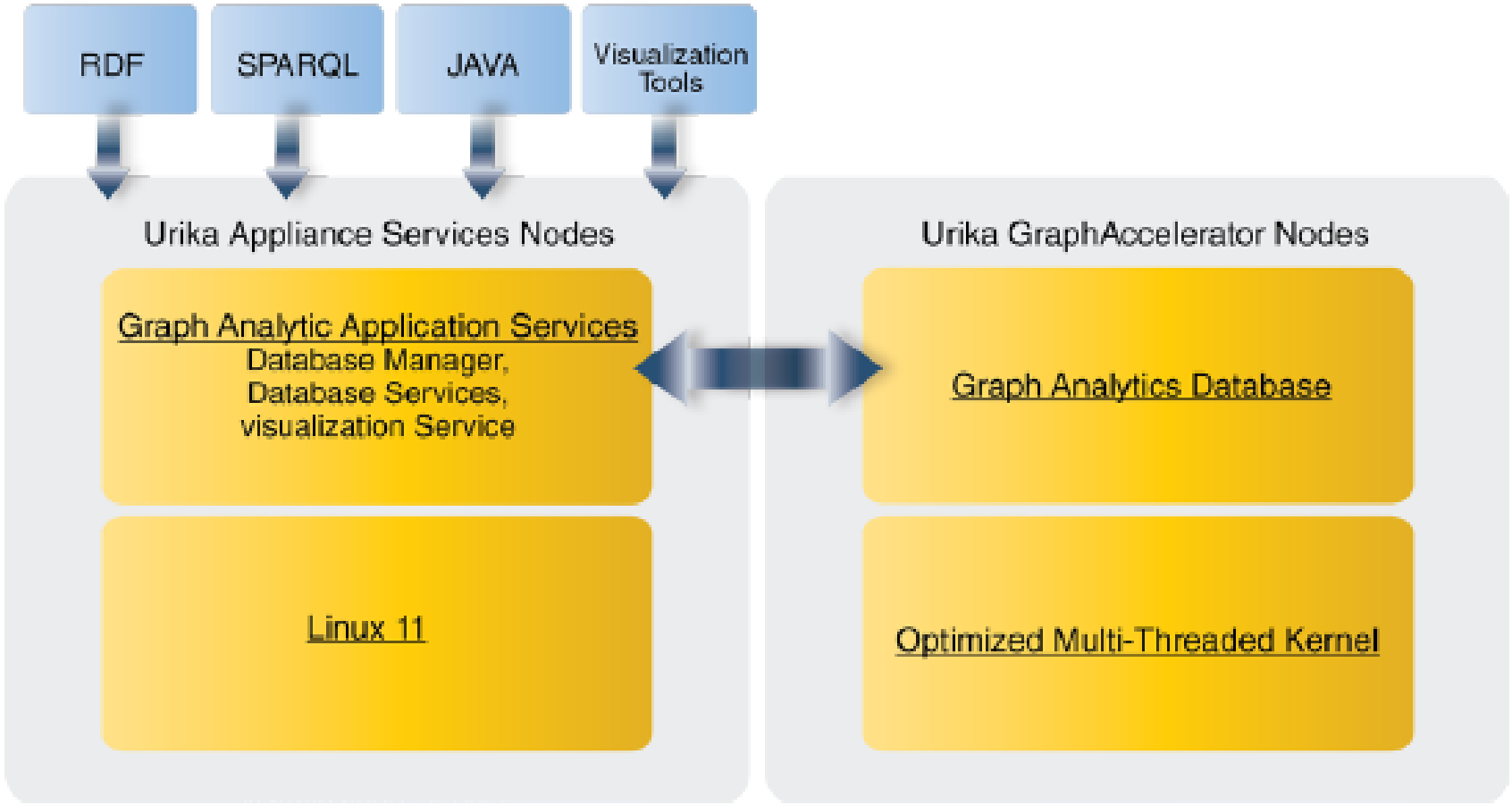
Urika-64 – Hardware

- Cray XT-5 architecture
  - Blade architecture
  - SeaStar2 interconnect
- Opteron based service nodes
- Compute nodes with Cray Threadstorm 4.0 ASIC
  - 64 processors with 128 streams per processor
  - 32 GB of memory per processor

Urika-64 – Software

- Not at all like a Cray XT-5
- Linux, unified view of memory and processors
  - 2 TByte of memory, 64 processors, 8192 streams
- Cray proprietary RDF triple store, exposed via an Apache Jena interface
- Large data import via Lustre possible
- Web interface for running queries

Urika-64 – Software



Contents

- Motivation and Introduction
- Use Case
- Hardware and Software
- Methodology and Results
- Conclusion

Data Set

Dataset	Triple Count
http://chem2bio2rdf.org/omim	17,251
http://chem2bio2rdf.org/kegg	245,997
http://chem2bio2rdf.org/reactome	15,849
http://chem2bio2rdf.org/ctd	4,933,479
http://chem2bio2rdf.org/chebi	5,812,141
http://chem2bio2rdf.org/dcdb	20,780
http://chem2bio2rdf.org/bindingdb	1,191,201
http://chem2bio2rdf.org/hprd	477,697
http://chem2bio2rdf.org/hgnc	1,720,541
http://chem2bio2rdf.org/medline	480,716,135
http://chem2bio2rdf.org/kidb	744,738
http://chem2bio2rdf.org/pubchem	15,439,873
http://chem2bio2rdf.org/qsar	32,206
http://chem2bio2rdf.org/bindingmoad	252,938
http://chem2bio2rdf.org/matador	269,656
http://chem2bio2rdf.org/pharmgkb	512,361
http://chem2bio2rdf.org/pdb	95,925
http://chem2bio2rdf.org/ttd	116,767
http://chem2bio2rdf.org/chembl	85,156,878
http://chem2bio2rdf.org/medline_lite	56,212,993
http://chem2bio2rdf.org/dip	1,113,871
http://chem2bio2rdf.org/sider	127,755
http://chem2bio2rdf.org/uniprot	1,994,607
http://chem2bio2rdf.org/drugbank	436,283
http://chem2bio2rdf.org/chemogenomics	7,327,361
Total	664,985,283

Methodology

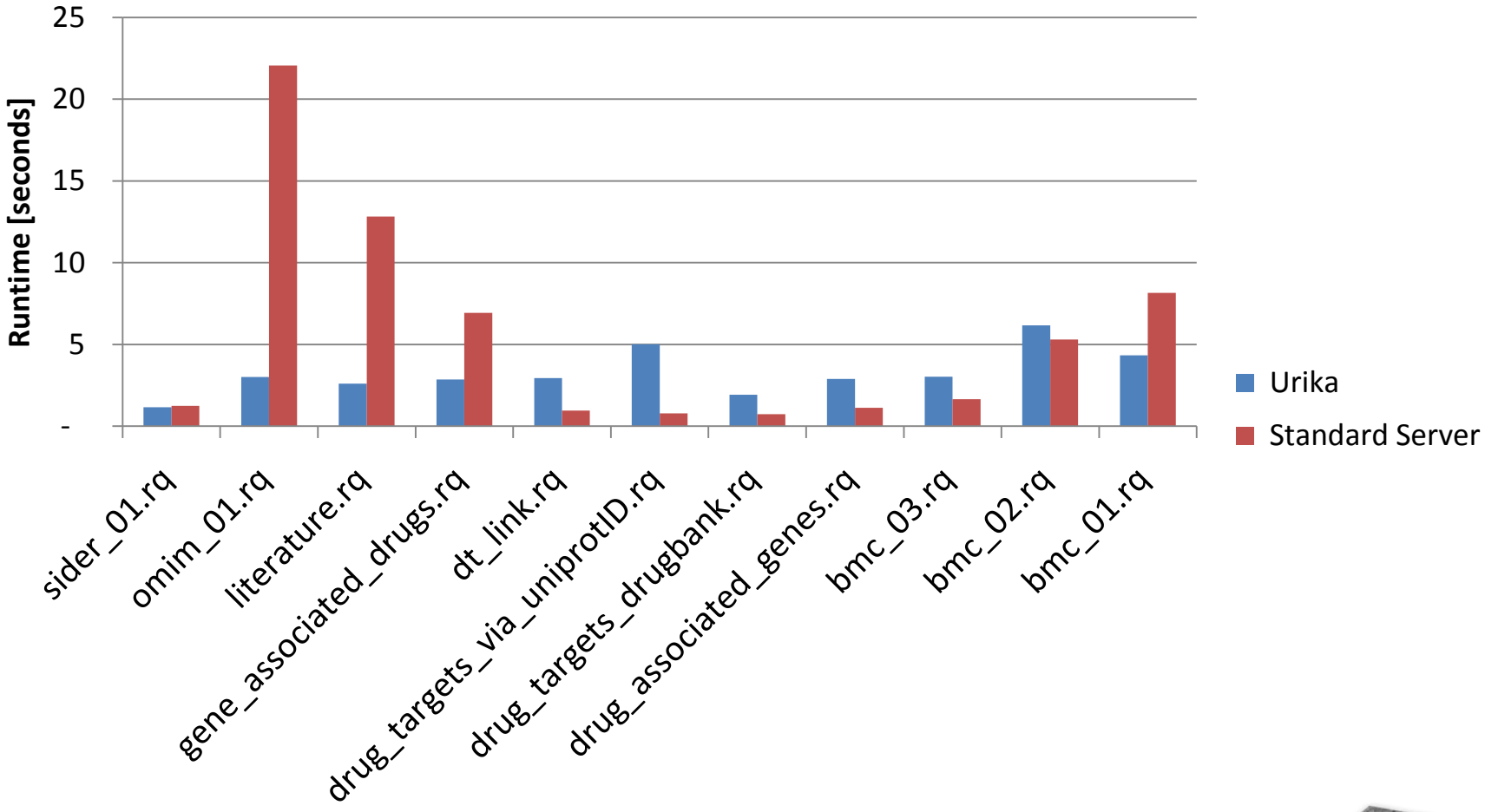
- Import all triples into the Urika
- Create database
- Start database
- Run query through the web interface
- Run query using Java application

Overall Results

- Importing the data and running SPARQL queries worked
- Accessing the SPARQL endpoint was easy
- Custom hardware, proprietary software, exposed via standards compliant interfaces

- Performance results are interesting

Performance Results



Conclusion

- Initial testing of how to use a Urika went fine
- Integrating it into an existing workflow is easy
- Getting researchers to use a Urika could be a challenge
- Not sure about multi user operation
- Our tests do not allow for a performance comparison

Thank You!

Questions?!