Quickstart Big Data

Dr. Olaf Flebbe
of ät oflebbe.de

Fosdem 4.2.2017
HPC, Big Data and Data Science devroom

About me

PhD in computational physics

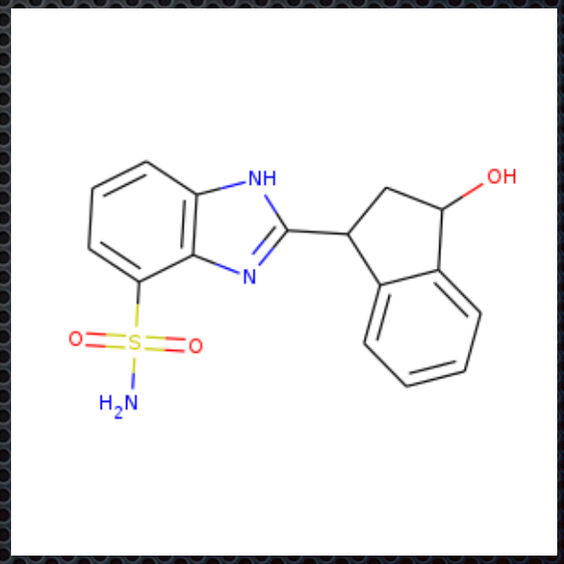
Former projects: Minix68k (68k FP Emulation), Linux libm.so.5 (High Precision FP), perl and python for epoc, flightgear, msktutil…

Member of PMC of Apache Bigtop

Software Architect for connected E-Bikes

Problem from ChemInformatics

- Database
  For instance:
  Synthetically Accessible Virtual Inventory (SAVI) project



- SMILES Notation

OC1CC(C2=CC=CC=C12)C4=NC3=C(C=CC=C3[S](=O)(=O)N)[NH]4

Problem: Substructure Search

- Commeratial solution: Enterprise DB and a Cartridge
- Look out for a Big Data Approach
- Use python rdkit

```
from rdkit import Chem

m = Chem.MolFromSmiles('OC1CC(C2=CC=CC=C12)C4=NC3=' +
'C(C=CC=C3[S](=O)(=O)N)[NH]4')

m.HasSubstructMatch(Chem.MolFromSmiles('C[C@H](F)Cl'))
```

Problem at Hand

- Ingredients
  - Time consuming python code
  - Large Datasets: Even more time consuming
- Environment
  - Big Data Cluster
  - HPC Cluster

How (not) to scale out

- Most of the runtime is needed for constructing the molecule object
  - AFAIK One of the problem is to identify ring structures
    (to aromatize)
- Can be improved by pickling (serializing) Molecule Objects ...
- A huge gain but not scaling effect

How to scale out

- Problem at hand is EP (embarrasingly parallel)
- Can be solved by "Distribute the Algorithm to the data"
  - Use Apache Spark-Core python binding
    - Use the RDD paradigm
    - Runs in HPC and BigData Environments

How to scale out

- Read the Instructions at Spark.a.o

```
def matchmol( smile) :
        return
Chem.MolFromSmiles( smile).\

HasSubstructMatch('C[C@H](F)Cl')

input = sc.textFile("file:///directory/
smiles.txt")
input.filter( matchmol).count()
```

HPC Mode

- HPC Cluster of Machines with
  - Use Cluster Filesystem for
    - Deploying Apache Spark
    - Distribute Data
  - Does not use locality of data, but works.
- Use for instance Standalone Mode

Big Data Setup

- Well ....

Big Data Distro

Apache Bigtop

- Apache Bigtop is the „Debian“ of the Big Data Distributions
- reused by Google for their Managed Hadoop Service
- reused within Cloudera and Hortonworks
- used by Canonicals Hadoop Offering
- reused by the ODPI.org

Some components of Apache Bigtop

Components

- Compile Environment (based on docker)
- Convenience artifacts (i.e. repositories for Centos7, Centos6, Debian 8, Ubuntu 16.04, Ubuntu 14.04, Fedora 20, opensuse 42.1)
- Provisioning with docker compose (openstack)
- Deployment Templates based on puppet
- Orchestration with Juju Charms
- Automatic Testing Environment: iTests
- And … non intel architectures (ppc64le, aarch64)

Example

Konfigurationsmatrix	centos-6	centos-7	debian-8	fedora-25	fedora-25-ppc64le	opensuse-42.1	ubuntu-14.04-aarch64	ubuntu-16.04	ubuntu-16.04-ppc64le
apex	●	●	●	●	●	●	●	●	●
alluxio	●	●	●	●	●	●	●	●	●
bigtop-groovy	●	●	●	●	●	●	●	●	●
bigtop-jsvc	●	●	●	●	●	●	●	●	●
bigtop-tomcat	●	●	●	●	●	●	●	●	●
bigtop-utils	●	●	●	●	●	●	●	●	●
crunch	●	●	●	●	●	●	●	●	●
datafu	●	●	●	●	●	●	●	●	●
flume	●	●	●	●	●	●	●	●	●
flink	●	●	●	●	●	●	●	●	●
giraph	●	●	●	●	●	●	●	●	●
gpdb	●	●	●	●	●	●	●	●	●
hadoop	●	●	●	●	●	●	●	●	●
hama	●	●	●	●	●	●	●	●	●
hbase	●	●	●	●	●	●	●	●	●
hive	●	●	●	●	●	●	●	●	●
hue	●	●	●	●	●	●	●	●	●
ignite-hadoop	●	●	●	●	●	●	●	●	●
kafka	●	●	●	●	●	●	●	●	●
kite	●	●	●	●	●	●	●	●	●
mahout	●	●	●	●	●	●	●	●	●
oozie	●	●	●	●	●	●	●	●	●
phoenix	●	●	●	●	●	●	●	●	●
pig	●	●	●	●	●	●	●	●	●
qfs	●	●	●	●	●	●	●	●	●
solr	●	●	●	●	●	●	●	●	●
spark1	●	●	●	●	●	●	●	●	●
spark	●	●	●	●	●	●	●	●	●
sqoop	●	●	●	●	●	●	●	●	●
sqoop2	●	●	●	●	●	●	●	●	●
tajo	●	●	●	●	●	●	●	●	●
tez	●	●	●	●	●	●	●	●	●
ycsb	●	●	●	●	●	●	●	●	●
zookeeper	●	●	●	●	●	●	●	●	●
zeppelin	●	●	●	●	●	●	●	●	●

Use with Bigtop 

- Use puppet scripts to deploy Apache Hadoop
  - Use HDFS
  - Use YARN Mode
  - Download and unpack Spark 2.1.x

Bigtop 1.2 (preview)

- Will have Spark 2.1
- OpenJDK 8 Support
- Not finished ;-(
- We need help ...
- join at bigtop.apache.org

Conclusion

- Problem runs much better in HPC environment
  - because it is compute intensive
  - not limited by data pipeline bandwith
- Scales really well O(n)
- HW amount needed to challange the existing solution is too large... need to investigate further

Thanks

of ät oflebbe dot de