BDP ideas for 17/18

Improve Slides 1:

Introduce concept of the data pipeline earlier:

<https://dzone.com/articles/how-to-build-an-analytic-pipeline>

Go back to Marz book and make better use of it. For example in intro, used characteristics of Big Data from this and mentioned Lambda, but could use material in chapter 1 to explain issues that Lambda addresses and to list types of tool (1.8.3) – got most of the latter somehow but could organise better. First chapter under Batch, on modelling looks useful for intro to NoSQL and to filesystems.

There is a 1718 version of slides\_1 – not completed but contains notes on how to revise.

Consider bringing material on filesystems/formats before material on NoSQL – maybe not, could keep it after NoSQL and as intro to analytics/pipelines

Multi-model databases are becoming more important, expand on these (ArangoDB and particularly Azure Cosmos DB) – maybe integrate with cloud module here.

Mention Thrift as well as Avro

Integrate Hadoop coverage a bit better – maybe use cloud service referred to in IBM Big Data Univ course, and use their lab setup, but create own materials:

<https://my.imdemocloud.com/>

e. uni p. +Ibm+ (last + literal)  
in system username is jamespaterson, pwd is the same

(Note that IBM Big Data University account is [jim@paterson.co.uk](mailto:jim@paterson.co.uk))

Alternatively, negotiate with Nhamo and: use his Hadoop lab (maybe do it in Python?) for those who haven’t done his module; or bring that material into my module instead of his

Look at link in links file for Hadoop with Python or (better) use Hadoop with Python eBook (in books folder and resources folder for this module) – book suggests use of the Snakebite Python client for Hadoop and mrjob, note that Hadoop supports languages other than Java using Hadoop Streaming and these tools make use of this.

Can we make use of <https://datascientistworkbench.com/> - alternative to Databricks if not available

Cover more of ecosystem – Hive/Pig

\*Update on Hadoop thinking – look at what Nhamo does and build on that, could do more on HDFS and YARN, Flume, etc, could cover Hadoop streaming and show examples in Python (Nhamo showed example in Java last time). Get Definitive Guide book (White/O’Reilly), has some good stuff on other things, Hive, Pig, Flume, HBase, ZooKeeper, Avro, Parquet, Sqoop, Crunch

Look at Apache Drill, e.g. with MongoDB and Parquet files (note that Drill installation includes sample Parquet data)

Spark ML – show example, find out what is done in Data Analytics and show how a particular example/ML algorithm can be applied in spark

Show GraphX example too?

Lab 1 (Python) – look at Jupyter notebook as working environment instead of or in addition to PyCharm (install Anaconda!) - keep an eye on what Nhamo does with Python in general, NumPy and pandas, and adjust accordingly to revise/expand on what he has covered

Could do more on cloud deployment, e.g. using Docker on AWS – not sure where this would go, maybe in general NoSQL lecture, or maybe later on.

Explain better why you would use Kafka – more control over parallelism, see http://www.michael-noll.com/blog/2014/10/01/kafka-spark-streaming-integration-example-tutorial/

Assignment – be more specific about what’s wanted, e.g. design proposal to pitch to client, include diagrams illustrating solution. Ask to describe in terms layed out in <https://dzone.com/articles/how-to-build-an-analytic-pipeline>

(and introduced in revised lecture 1)

Provide data source, not as obvious as Twitter so that off-the-shelf solutions are not available

Specify implementation more so that absolutely minimal attempts are not acceptable:

e.g. read data as streaming (as in lecture demo), filter/aggregate and store OR filter/aggregate/combine and visualise in a specified way

Try to devise task which will encourage individual work rather than minor variations on a theme

MongoDB aggregation – need to point out that match before group matches on field, match after group matches on aggregated value, e.g. sum – may need to add an example to Lab 3 for this.

**REQUIRED SOFTWARE:**

Linux VM (see installation notes from this year)

Python 2.7 – install with Anaconda, which enables Jupyter notebooks also and may make other installations easier (e.g with pip)

PyCharm Edu

Scala

IntelliJ with Scala plugin (Community Edition OK)

MongoDB

MongoClient and/or RoboMongo

PyMongo and test with simple Python app

Also test Scala connection using MongoDB Scala driver

Neo4j

Neo4j Python driver (<https://neo4j.com/developer/python/#neo4j-python-driver>) and test with simple Python app

Cassandra

Redis

Hadoop – test HDFS lecture examples

Spark –2.1, and test PySpark and Scala shells

Apache Kafka

Apache Drill – with support for file formats (CSV, Parquet, JSON), filesystems (local, HDFS, S3), and dbs (MongoDB, Hive)

parquet-tools (possibly) - https://github.com/Parquet/parquet-mr/tree/master/parquet-tools

Cloud (see installation notes from this year)

DataBricks community (for students also)

Amazon S3 for storage – look at what I can get for free

MongoDB – Mlab or Atlas (if free tier), use existing credentials if possible

Redis cloud, redismin, use existing credentials if possible