DATA CLOUD 2011
Data Intensive Computing In the Clouds


By
Peter Wittek, Thierry Jacquin, Hervé D’jean
Jean-Pierre Chanod, Sándor Darányi
Digital Preservation

- Retaining digital information & its meaning over time to ensure availability.

Focus

- Accessibility
  - New Formats evolve, legacy should be accessible.
- Migration & Transformation
  - Older formats could be transformed to a persistent format.
- Scalability
  - Should scale when new documents are added.
- Reusability
  - Depends on proper descriptions in the archive
WORKFLOWS AND DIGITAL PRESERVATION

- Important to preserve the intent behind the document processes and document content.
- Production of data and reuse of data may not share the same objective.
- Xeproc Domain Specific Language to store the intent and transformations used.

XML Processing Pipelines

- XML to represent logical structure of documents.
- XSLT to perform the transformation
- XML Schemas and DTD to check the validity.
Document Processing Pipeline

- Steps Involved in Extracting Table of Content.
  - Convert PDF to XML.
  - XSLT transformation for XML files.
  - Page header & footer recognition.
  - Text reading order.
  - Caption detection.
  - Table of content Analysis.

- Xeproc used for modelling XML processing pipelines & validation checks.
  - The pipeline produces metadata containing:
    - Logical organization like Chapters, Contents.
    - Page numbering.
    - Illustrations.
Issues with Digital Preservation

- XML Processing is compute intensive for large documents.
- Natural Language Processing tasks are compute intensive. Deep tree parsing, Named Entity recognition.
- Small organization do not focus on Data Conservation and cannot afford the infrastructure.
- Data Conservation is a ad hoc process.
- Cloud Computing provides an ideal platform for performing Digital preservation.
Architecture

- MapReduce Framework for processing documents.
  - Could use Low Cost commodity servers instead of HPCs.
  - Inherently provides redundancy.
  - Move processing to the data.

- Cloud Infrastructure to host MapReduce framework.
  - Ease of Resource Availability & scalability

- Mapper takes single XML document & starts the workflow involved with the document.

- Reducer emits the XML documents in METS format. (Metadata Encoding and Transmission Standard)

- Same task is replicated at different nodes.
Implementation

- Local Experiments on workstation
  - 24 GB Memory, Quadro-core Intel Xeon
  - 2 TB Storage

- For Cloud Computations.
  - Amazon Web Services.
  - Small standard instances
    - 1.7GB memory, 2 EC2 compute units
    - 150 GB instance storage
  - Large instance
    - 7.5 GB memory, four EC2 compute units
    - 850 GB memory
Data Sets Used

- DNB collection of electronic PhD theses.
  100000 Files are stored in PDF format.
- Digitized Collections from the Gottinger
  Digital-sierungszentrum.
  5 mn Documents related to various projects.
- INEX (Initiative for the Evaluation
  of XML)
  Contains over 50,000 digitized out of copyright
  books.
- Pipelin has been customized for each of these data sets.
MapReduce Performance

For small collections MapReduce does not outperform because of the initialization phase.
Performance on Cloud

![Graph showing running time (in minutes) vs. number of processing cores for different data sizes (100, 1000, 10000)].
Cost for Computation & Storage

Figure 6. Comparison of average cost of computations with different collection sizes