ARCHIVESPARK

Andrej Galad
12/6/2016
CS-5974 – Independent Study
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061
Professor Edward A. Fox

AGENDA

- ArchiveSpark Overview/Recap
- Benchmarking
- Demo

ARCHIVESPARK

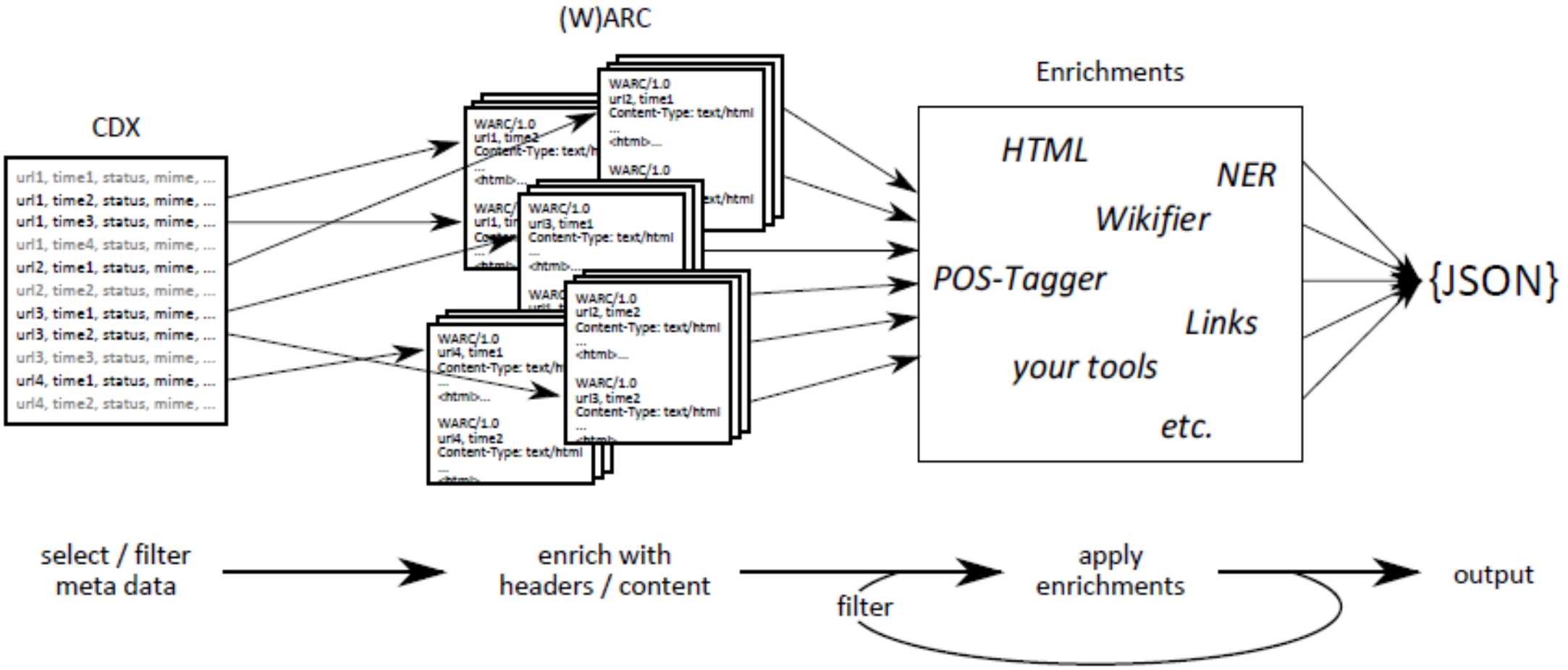
- Framework - efficient data access, extraction and derivation on Web archive data
- ArchiveSpark: Efficient Web Archive Access, Extraction and Derivation
  - Helge Holzmann, Vinay Goel, Avishek Anand
  - Published in JCDL 2016
  - Nominated for the Best Paper Award
- OS project - https://github.com/helgeho/ArchiveSpark

TWO TECHNIQUES

1. **Pre-generated CDX metadata index**
   - Smaller dataset
   - Reduction based on Web archive metadata
2. **Incremental filtering workflow**
   - “Extract only what you need”
   - Augment -> Filter -> Repeat

- Concept – **Enrichments**
  - ArchiveSpark Record extension
  - Featured – StringContent.scala, Html.scala, Json.scala, Entities.scala, Prefix.scala, …
  - Custom – mapEnrich[Source, Target](sourceField, targetField) (f: Source => Target)

WORKFLOW


(W)ARC
CDX
url1, time1, status, mime, ...
url1, time2, status, mime, ...
url1, time3, status, mime, ...
url1, time4, status, mime, ...
url2, time1, status, mime, ...
url2, time2, status, mime, ...
url3, time1, status, mime, ...
url3, time2, status, mime, ...
url3, time3, status, mime, ...
url4, time1, status, mime, ...
url4, time2, status, mime, ...
WARC/1.0
Content-Type: text/html
Enrichments
HTML
NER
Wikifier
POS-Tagger
Links
your tools
etc.
{JSON}
select / filter meta data
enrich with headers / content
filter
apply enrichments
output

FLEXIBLE DEPLOYMENT

- Ultimately a Scala/Spark library
- Environments
  - Standalone solitary Spark instance
  - Local HDFS-backed Spark cluster
  - Large-scale YARN/Mesos-orchestrated cluster running Cloudera/Hortonworks
- Quickstart – Docker (latest version)
- ArchiveSpark version – 2.1.0
  - Spark 2.0.2
  - Scala 2.11.7 -> Java 8

WARC FILES

- Standard Web archiving format - ISO 28500
- Single capture of web resource at a particular time
  - Header section – Metadata (URL, timestamp, content length…)
  - Payload section – Actual response body (HTML, JSON, binary data)
    - HTTP Response – HTTP headers (origin, status code)

```
WARC/1.0
WARC-Type: response
WARC-Record-ID: <urn:uuid:9e6f625c-74f6-4f9b-bec0-cbebc1102f4f>
WARC-Date: 2015-06-13T18:07:56Z
WARC-Target-URI: http://netbootcamp.s3.amazonaws.com/wp-content/uploads/netbootcamp-logo-internet-training.png
WARC-IP-Address: 54.231.65.41
Content-Type: application/http;msgtype=response
WARC-Payload-Digest: sha1:IS672GHFV5GLGOWBVVEN6BLF3DTLS7NB
Content-Length: 18196
WARC-Block-Digest: sha1:LI2ZUQH5PH2UDHGOODXHSGBEHC3UXDG4

HTTP/1.1 200 OK
x-amz-id-2: Bw5mQr6nHrJs+CtzJ0QImShahazsS7hqEgNswjI1jOsZ/+tByVayRWVUjL4ggMht
x-amz-request-id: D774CC57D37483F3
Date: Sat, 13 Jun 2015 18:08:35 GMT
Last-Modified: Wed, 24 Dec 2014 14:38:19 GMT
ETag: "1563196cca285cd1aa801a7a1ee91519"
Accept-Ranges: bytes
Content-Type: image/png
Content-Length: 17850
Server: AmazonS3
```

CDX INDEX

- "Reduced" WARC file
  - WARC metadata
  - Pointers to WARC records – offsets in WARC file
- CDX
  - Header – specifies metadata fields contained in the index
  - Body – typically 9 – 11 fields
    - Original URL, SURT, date, filename, MIME type, response code, checksum, redirect, meta tags, compressed offset

```
CDX N b a m s k r M S V g
com,example)/?example=1 20140103030321 http://example.com?example=1 text/html 200 B2LTWWPUOYAH7UIPQ7ZUPQ4VMBSVC36A - - 1043
333 example.warc.gz
com,example)/?example=1 20140103030341 http://example.com?example=1 warc/revisit - B2LTWWPUOYAH7UIPQ7ZUPQ4VMBSVC36A - - 553
1864 example.warc.gz
org,iana)/domains/example 20140128051539 http://www.iana.org/domains/example text/html 302 JZ622UA23G5ZU6Y3XAKH4LINONUEICEG
- - 577 2907 example.warc.gz
```

TOOLS

- CDX Writer
  - Python script for CDX extraction
  - Alternative to Internet Archive's Wayback Machine - http://archive.org
- Jupyter Notebook
  - Web application for code sharing/results visualization
- Warcbase (only benchmarking)
  - "State-of-the-art" platform for managing and analyzing Web archives
  - Hadoop/HBase ecosystem – CDH
  - Archive-specific Scala/Java objects for Apache Spark and HBase
  - HBase command-line utilities - IngestFiles

BENCHMARKING

- Evaluation of 3 approaches
  1. ArchiveSpark
  2. Pure Spark using Warcbase library
  3. HBase using Warcbase library

- Preprocessing
  - ArchiveSpark – CDX index files extraction
  - HBase – WARC ingestion

- ArchiveSpark Benchmark subproject
  - Requirements:
    - Built and included Warcbase
    - sbt assemblyPackageDependency -> sbt assembly

ENVIRONMENT

- Development
  - Cloudera Quickstart VM – CDH 5.8.2

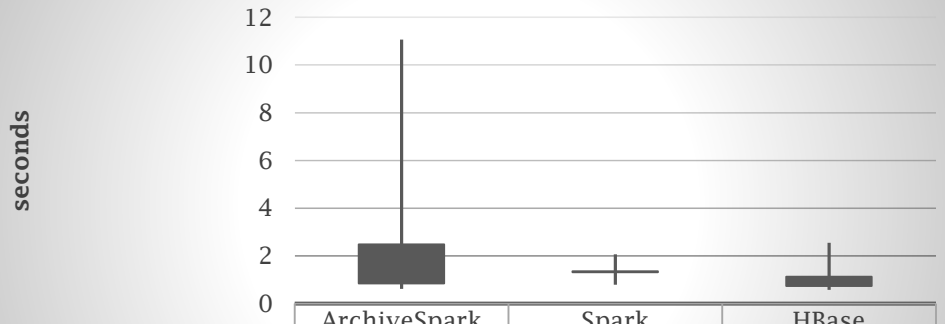
- Benchmarking
  - Cloudera CDH 5.8.2 cluster hosted on AWS (courtesy of Dr. Zhiwu Xie)
  - 5-node cluster consisting of m4.xlarge AWS EC2 instances
    - 4 vCPUs
    - 16 GiB RAM
    - 30 GB EBS storage
    - 750 Mbps network

BENCHMARK 1 – SMALL SCALE

- Filtering & corpus extraction
- 4 scenarios
  - Filtering of the dataset for a specific URL (one URL benchmark)
  - Filtering of the dataset for a specific domain (one domain benchmark)
  - Filtering of the dataset for a date range of records (one month benchmark)
  - Filtering of the dataset for a specific active (200 OK) domain (one active domain benchmark)
- Dataset - example.warc.gz
  - One capture of archive.it domain
  - 261 records
  - 2.49 MB

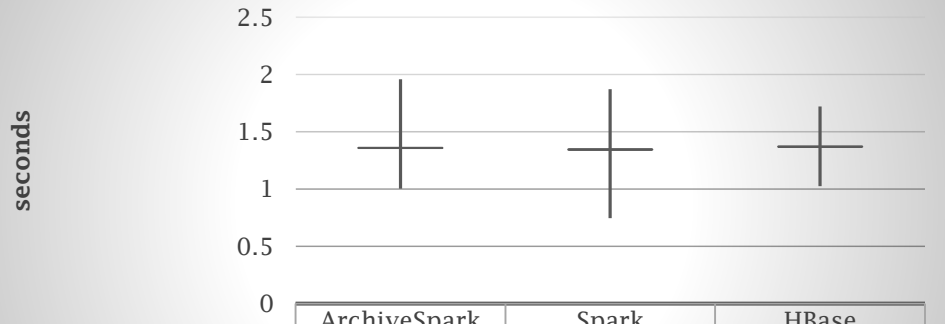
BENCHMARK 1 – RESULTS

One Url



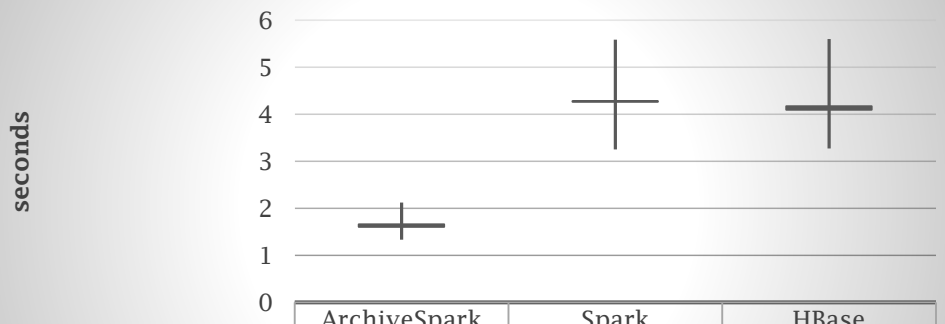
	ArchiveSpark	Spark	HBase
Average	2.477421343	1.359588305	1.127581107
Max	11.06495323	2.070165325	2.556766825
Min	0.620396273	0.787934256	0.583918894
Average/No outliers	0.850671088	1.316165381	0.74553106

One domain (text/html)



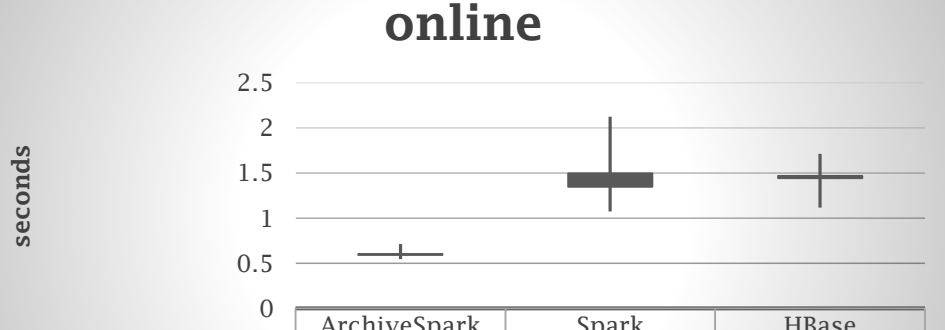
	ArchiveSpark	Spark	HBase
Average	1.362194425	1.347103491	1.371052337
Max	1.959518501	1.873476908	1.722339119
Min	1.003479605	0.745562134	1.026370613
Average/No outliers	1.362194425	1.347103491	1.371052337

One month online



	ArchiveSpark	Spark	HBase
Average	1.661291172	4.280218437	4.173602424
Max	2.124806716	5.588506367	5.602307053
Min	1.336782951	3.251106002	3.271654471
Average/No outliers	1.609789445	4.269318795	4.09674533

One domain (text/html) online



	ArchiveSpark	Spark	HBase
Average	0.60295292	1.497542806	1.468800563
Max	0.712397826	2.122264735	1.710818983
Min	0.547894969	1.073288654	1.11752257
Average/No outliers	0.590792374	1.347723263	1.441909627

BENCHMARK 2 – MEDIUM SCALE

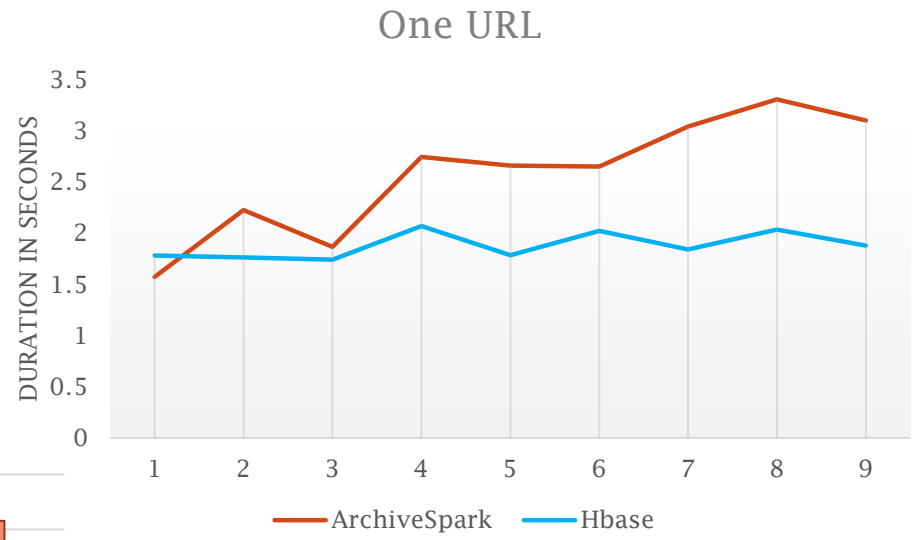
- Filtering & corpus extraction
- 4 scenarios
  - Filtering of the dataset for a specific url (one url benchmark)
  - Filtering of the dataset for a specific domain (one domain benchmark)
  - Filtering of the dataset for a specific active (200 OK) domain (one active domain benchmark)
  - Filtering of the dataset for pages containing scripts (pages with scripts benchmark)

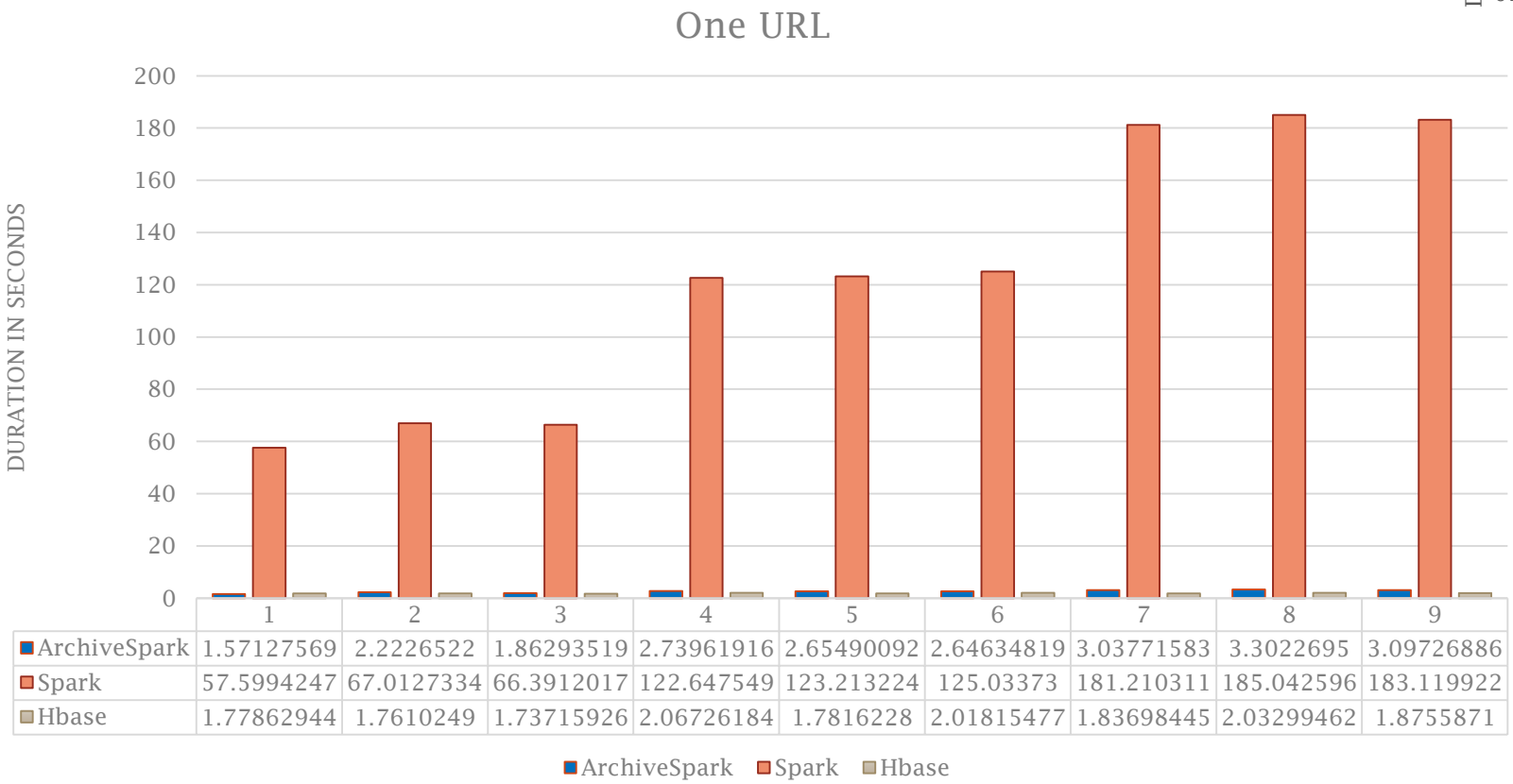
- Dataset – WIDE collection
  - Internet Archive crawl data from Webwide Crawl (02/25/2011)
  - 214470 records
  - 9064 MB
  - 9 files – approx. 1 GB

BENCHMARK 2 – PREPROCESSING

- CDX Extraction
  - 4 minutes 41 seconds
- HDFS Upload
  - 2 minutes 46 seconds
- HBase Ingestion x9
  - 1 file – (1 minute 10 seconds <-> 1 minute 32 seconds)
  - Sequential ingestion – approx. 13 minutes 54 seconds

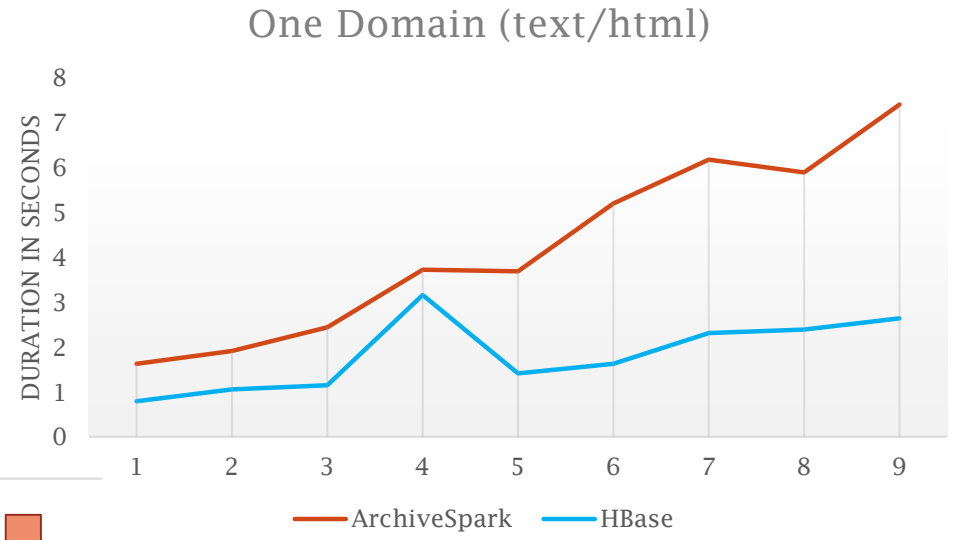
BENCHMARK 2 – RESULTS

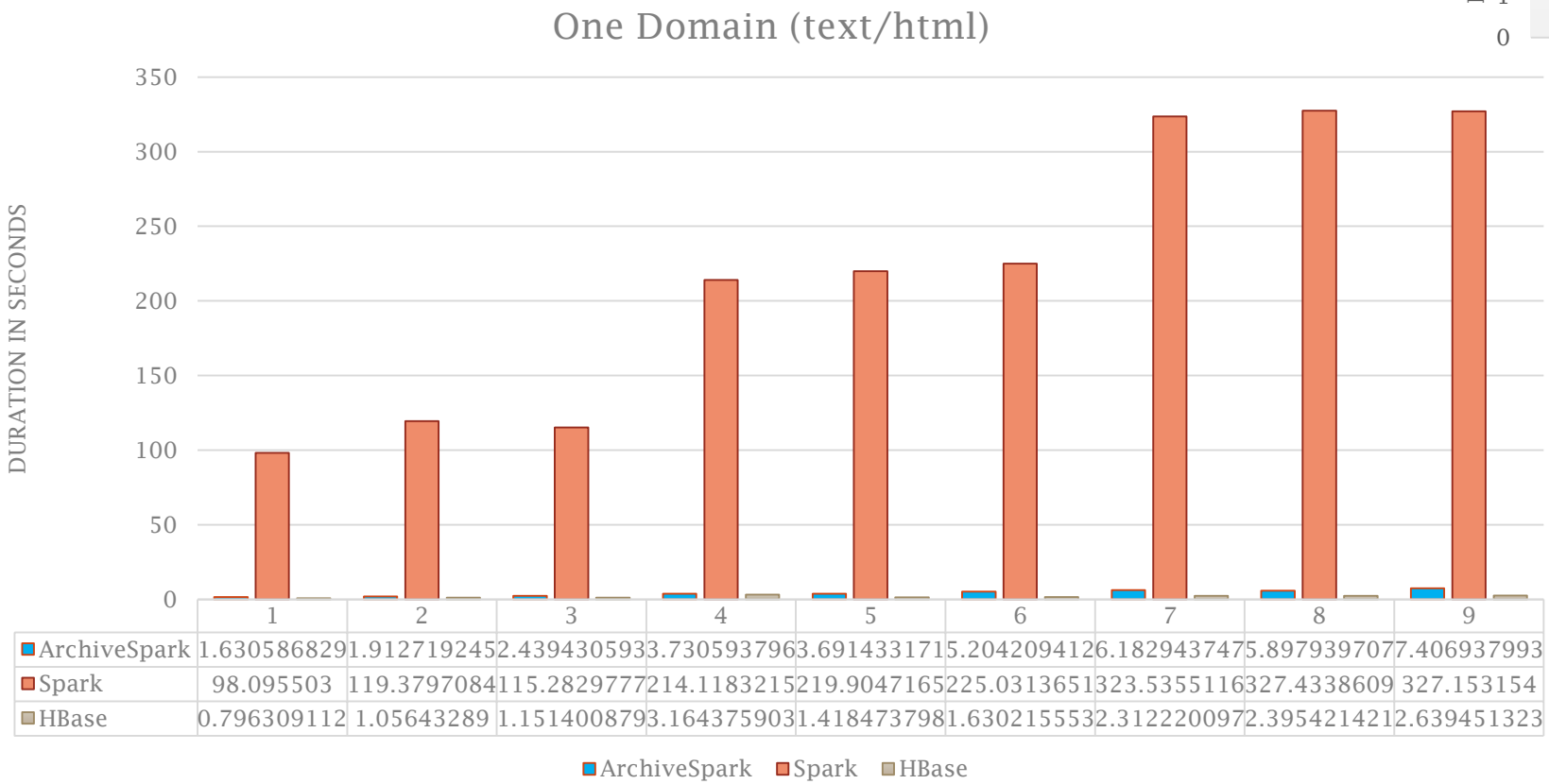




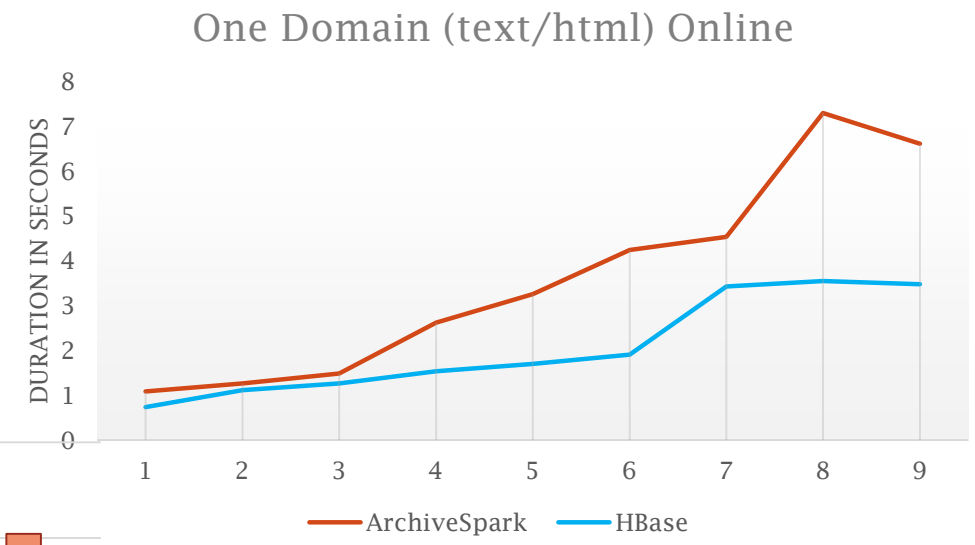
	1	2	3	4	5	6	7	8	9
ArchiveSpark	1.57127569	2.2226522	1.86293519	2.73961916	2.65490092	2.64634819	3.03771583	3.3022695	3.09726886
Spark	57.5994247	67.0127334	66.3912017	122.647549	123.213224	125.03373	181.210311	185.042596	183.119922
Hbase	1.77862944	1.7610249	1.73715926	2.06726184	1.7816228	2.01815477	1.83698445	2.03299462	1.8755871

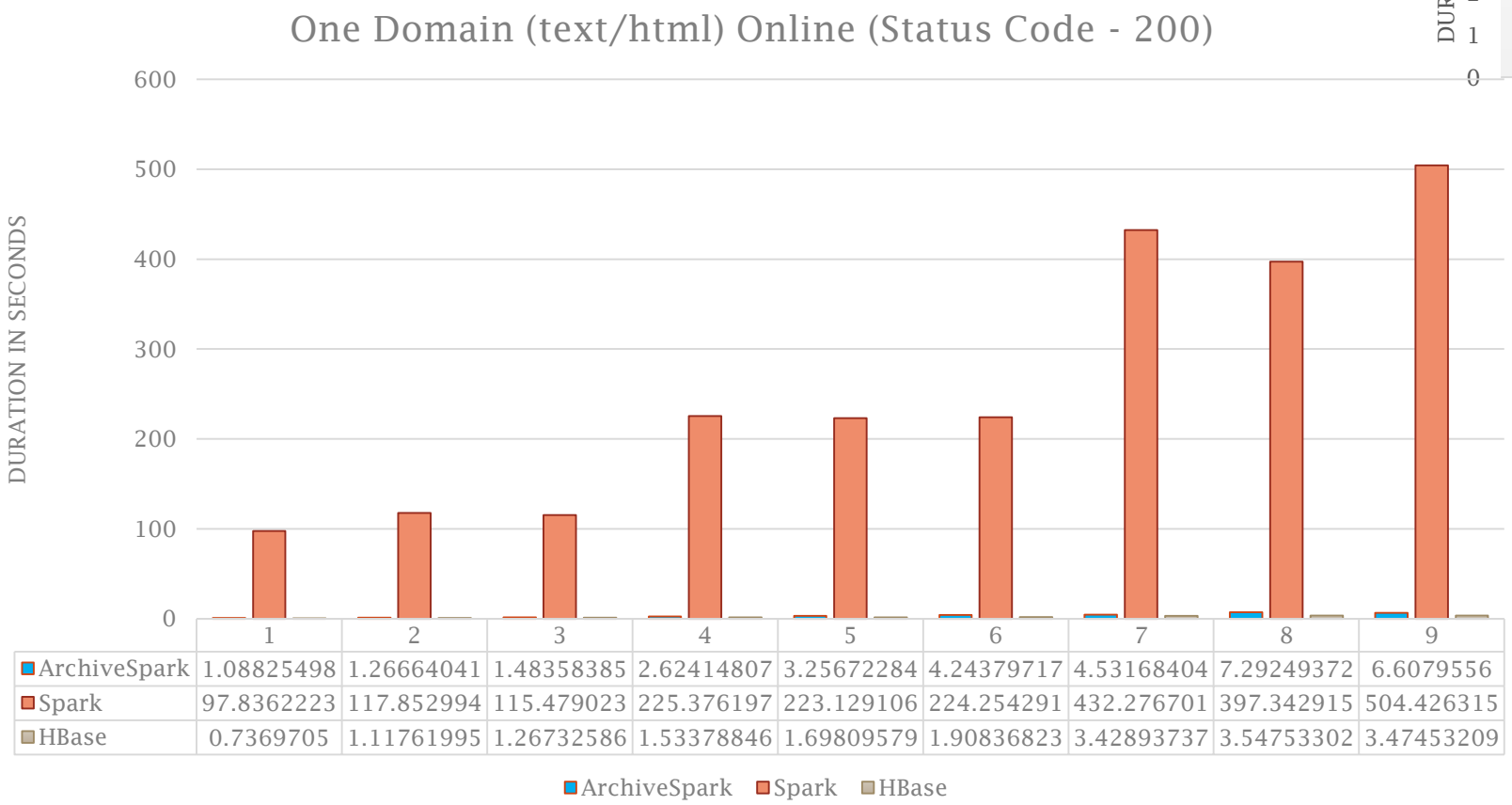
BENCHMARK 2 – RESULTS





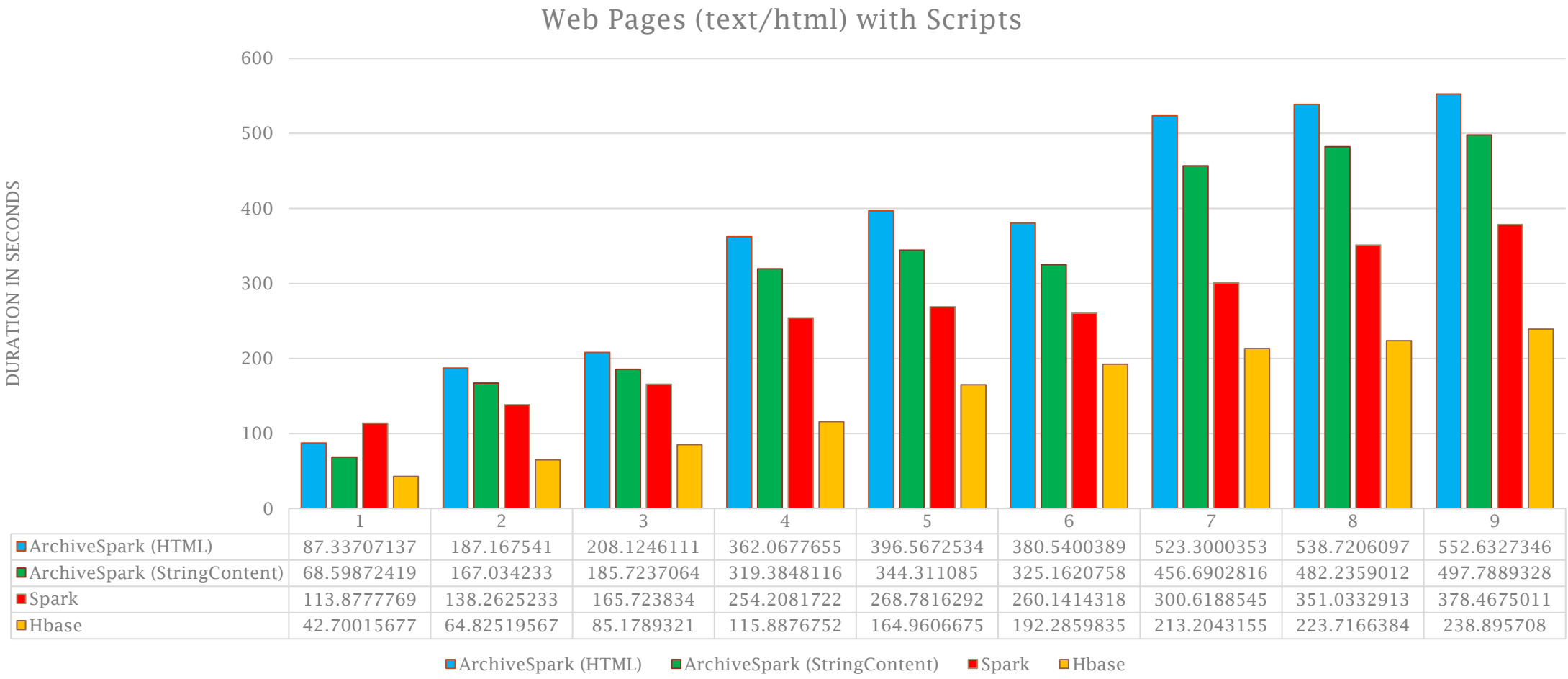
BENCHMARK 2 – RESULTS





	1	2	3	4	5	6	7	8	9
ArchiveSpark	1.08825498	1.26664041	1.48358385	2.62414807	3.25672284	4.24379717	4.53168404	7.29249372	6.6079556
Spark	97.8362223	117.852994	115.479023	225.376197	223.129106	224.254291	432.276701	397.342915	504.426315
HBase	0.7369705	1.11761995	1.26732586	1.53378846	1.69809579	1.90836823	3.42893737	3.54753302	3.47453209

BENCHMARK 2 – RESULTS



	1	2	3	4	5	6	7	8	9
ArchiveSpark (HTML)	87.33707137	187.167541	208.1246111	362.0677655	396.5672534	380.5400389	523.3000353	538.7206097	552.6327346
ArchiveSpark (StringContent)	68.59872419	167.034233	185.7237064	319.3848116	344.311085	325.1620758	456.6902816	482.2359012	497.7889328
Spark	113.8777769	138.2625233	165.723834	254.2081722	268.7816292	260.1414318	300.6188545	351.0332913	378.4675011
Hbase	42.70015677	64.82519567	85.1789321	115.8876752	164.9606675	192.2859835	213.2043155	223.7166384	238.895708

ACKNOWLEDGEMENT & DISCLAIMER

- This material is based upon work supported by following grants:
  - IMLS LG-71-16-0037-16: Developing Library Cyberinfrastructure Strategy for Big Data Sharing and Reuse
  - NSF IIS-1619028, III: Small: Collaborative Research: Global Event and Trend Archive Research (GETAR)
  - NSF IIS - 1319578: III: Small: Integrated Digital Event Archiving and Library (IDEAL)
- Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

THANK YOU

http://tinyurl.com/zejgc9f

TUTORIAL