Ten years of dumps at Wikimedia

About the speaker

- Dumps wrangler for 10 years at WMF
- SRE team member for $reasons
- IRC: apergos or atglenn
- Wikis: ArielGlenn
- Gender: nonbinary
- Age: git offa my lawn!

What are these dumps?

- TL;DR: dumps of content and metadata of Wikipedia and all other Wikimedia projects
- Long version: some db tables in sql format, page/revision content and metadata in xml format, edit logs in xml format, short abstracts of articles in xml format, site information in json format, checksums of everything
- Cost: free to you
- License: CC-BY-SA, reuse and share everything please!
- More details: meta.wikimedia.org/wiki/Data_dumps

Sample xml output

```
  <page>
    <title>Δεκέμβριος</title>
    <ns>0</ns>
    <id>2006</id>
    <revision>
      <id>2783885</id>
      <parentid>2763226</parentid>
      <timestamp>2012-11-12T21:36:00Z</timestamp>
      <contributor>
        <username>Flubot</username>
        <id>448</id>
      </contributor>
      <minor />
      <comment>ενημέρωση των interwikis, προσθήκη bg</comment>
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text xml:space="preserve">=={{-el-}}==
{{ΟιΜήνεςΤουΧρόνου}}
{{el-κλίσ-'Φίλιππος'|Δεκέμβρι|Δεκεμβρί}}
==={{ετυμολογία}}===
: '''{{PAGENAME}}''' &lt; {{ετυμ la}} [[December]] &lt; [[decem]] [[δέκα]] + επίθημα ''-ber'' (π
 ημερολόγιου)
```

Who uses these?

- Researchers
- Editors
- Sites with mirrors
- Colleagues at WMF
- Folks working in NLP
- Offline wiki reader projects

(Some) derived datasets

- DBpedia (2016): NIF-annotated article texts, 6.6 million entities containing variously abstracts, geo coordinates and depictions, as RDF triples
- T-REx (2017): 11 million alignments from 3.09 million DBpedia abstracts and Wikidata entity triples, in ttf or json format
- HotpotQA (2018): 113k Wikipedia-based “multi-hop” question-answer pairs
- Wikitext-103 (2016): 28,595 Wikipedia articles, 103 million tokens, 267,735 unique tokens, WikiText-2 (2016) 2 million tokens
- DAWT (2018) 13.5 million articles in 6 languages densely annotated with Freebase IDs
- SEW (2016): sense-annotated Wikipedia corpus with over 200 million annotations
- Wiki-Talk (2016): 95 million user and article talk diffs, 1 million crowd-sourced annotations of 100k diffs
- LTS2 NRC (2019): queryable graph dataset containing 5.7 million article and 1.7 million category nodes, along with a timeseries dataset containing page views information
- ...and many more

Generating datasets from latest dumps

- https://github.com/dbpedia/extraction-framework
- https://github.com/hadyelsahar/RE-NLG-Dataset
- https://github.com/epfl-lts2/sparkwiki
- https://github.com/yfiua/wiki-talk-parser
- https://github.com/idio/wiki2vec
- https://github.com/LuminosoInsight/wikiparsec
- https://github.com/JonathanRaiman/wikipedia_ner
- https://github.com/attardi/wikiextractor
- https://github.com/facebookresearch/fastText/
- …and many more

Machine learning from the dumps

- Building up Ontologies with Property Axioms from Wikipedia (2018)
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018)
- Universal Neural Machine Translation for Extremely Low Resource Languages (2018)
- Automated Speech Generation from UN General Assembly Statements:Mapping Risks in AI Generated Texts (2019)
- Learning to Interpret Satellite Images Using Wikipedia (2018)
- A Cost Efficient Approach to Correct OCR Errors in Large Document Collections (2019)
- ERNIE: Enhanced Language Representation with Informative Entities (2019)
- ...and many more

Dump use by bots/editors

- AutoWikiBrowser – script-assisted bulk editing
- Wikipedia:WikiProject Check Wikipedia
- Wikidata Navel Gazer (statement addition counts)
- Wikipedia:Typo Team corrections of typos, grammatical errors, and so on
- PyWikiBot (via replace.py)
- Many uses of the wikidata entity dumps, but someone else should give that talk!

Other wiki-related dumps projects

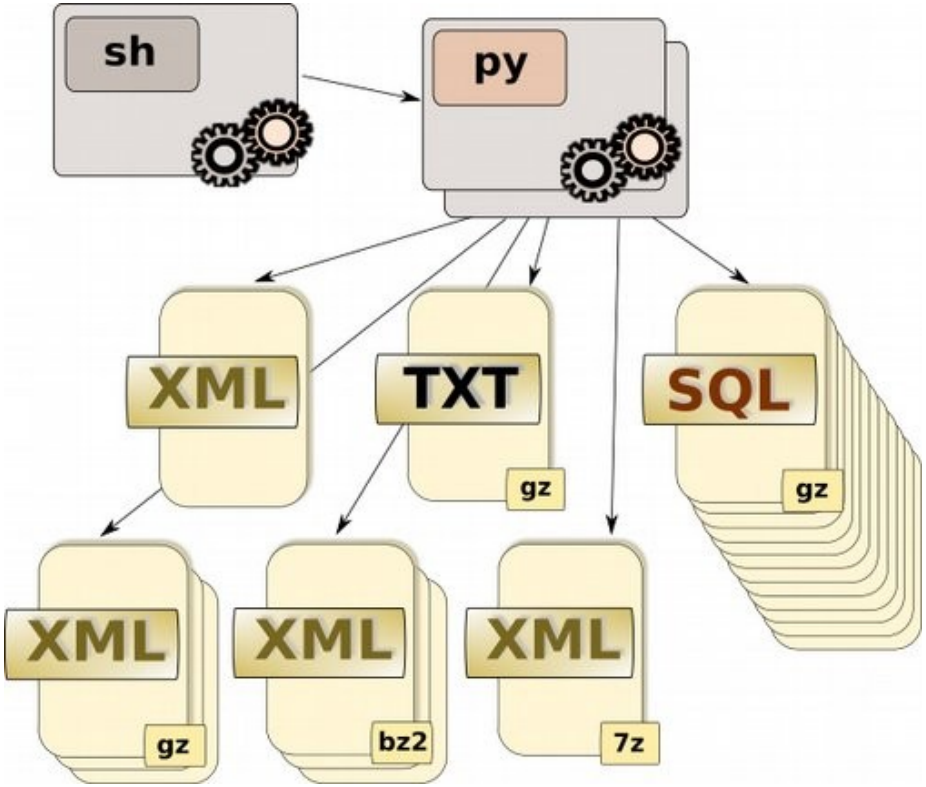
- Kiwix Wikivoyage travel app
- OpenZIM offline Wikipedia
- XOWA local Wiki project mirror
- Wikipedia-based cultural knowledge quiz (“Test Your Culture”)
- Analysis of Wikipedia edit networks
- WikiEd Error Corpus of corrected edits
- ...and, you know the drill: many more!

Dump file converters and parsers

- Xml to csv: https://github.com/Grasia/wiki-scripts/tree/master/wiki_dump_parser
- Load into Mongo: https://github.com/spencermountain/dumpster-dive/blob/master/README.md
- Xml to JSON: https://radimrehurek.com/gensim/scripts/segment_wiki.html
- Xml to SQL: https://github.com/wikimedia/mediawiki-tools-mwdumper
- Load into Hadoop: https://github.com/weikaolun/wikipedia-map-reduce
- Parser/querier: https://github.com/mediawiki-utilities/python-mwviews
- Load into ElasticSearch: https://github.com/AlonEirew/wikipedia-to-elastic
- Many Xml to text converters and extractors...

Simpler times

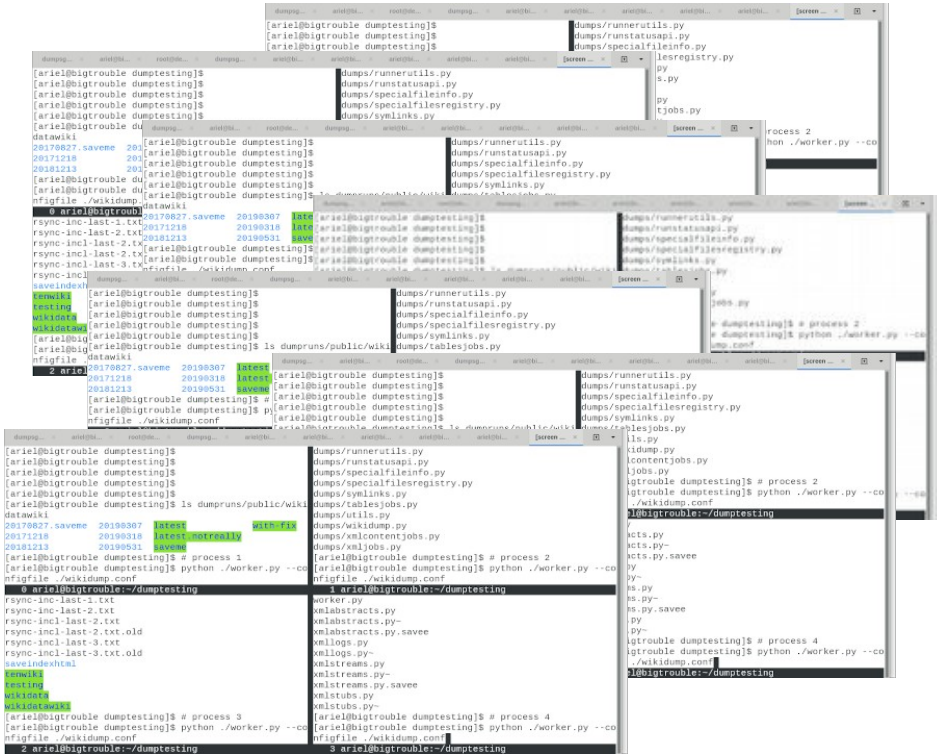
- 2 bash scripts
- 3 python scripts
- 1513 lines of code
- WAT? That's all?
- Yep, that's all!



Note: one of the two bash scripts and one of the three python scripts are just for producing the main index.html file and are not shown in the diagram.

Multiple processes, one host

- One process in screen session
- A second process? A second window in screen session
- A third process? 3 windows
- 27 usable cores? 27 windows…
- Split each window 4 ways, still 7 windows total



Multiple processes, multiple hosts

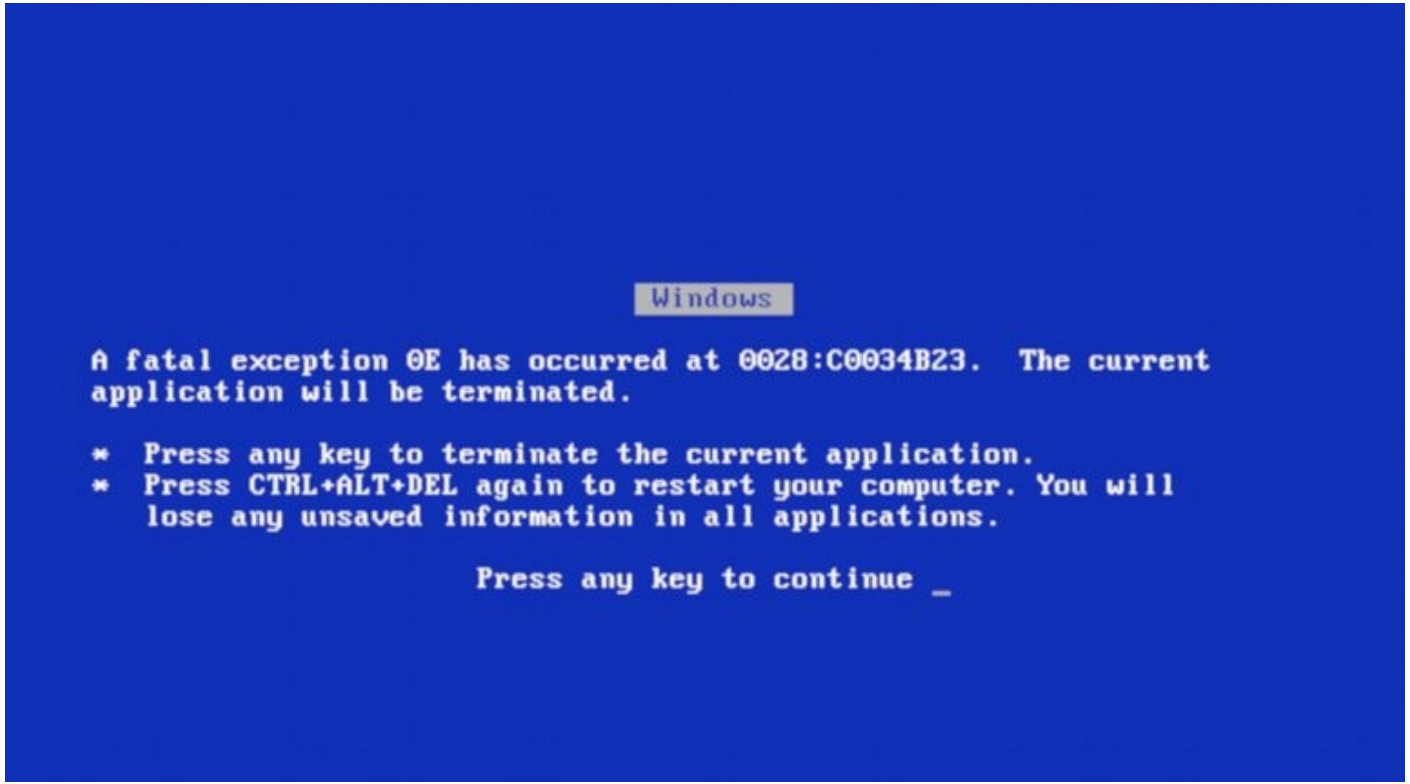
- One host? 7 screen windows on that host
- 2 hosts? 14 screen windows on 2 hosts
- 3 hosts? You know the drill

When does a dump run complete?

- Never. It just keeps running...and running...and running
- Wiki waiting longest for a dump starts next

What if something breaks?

It’s broken.

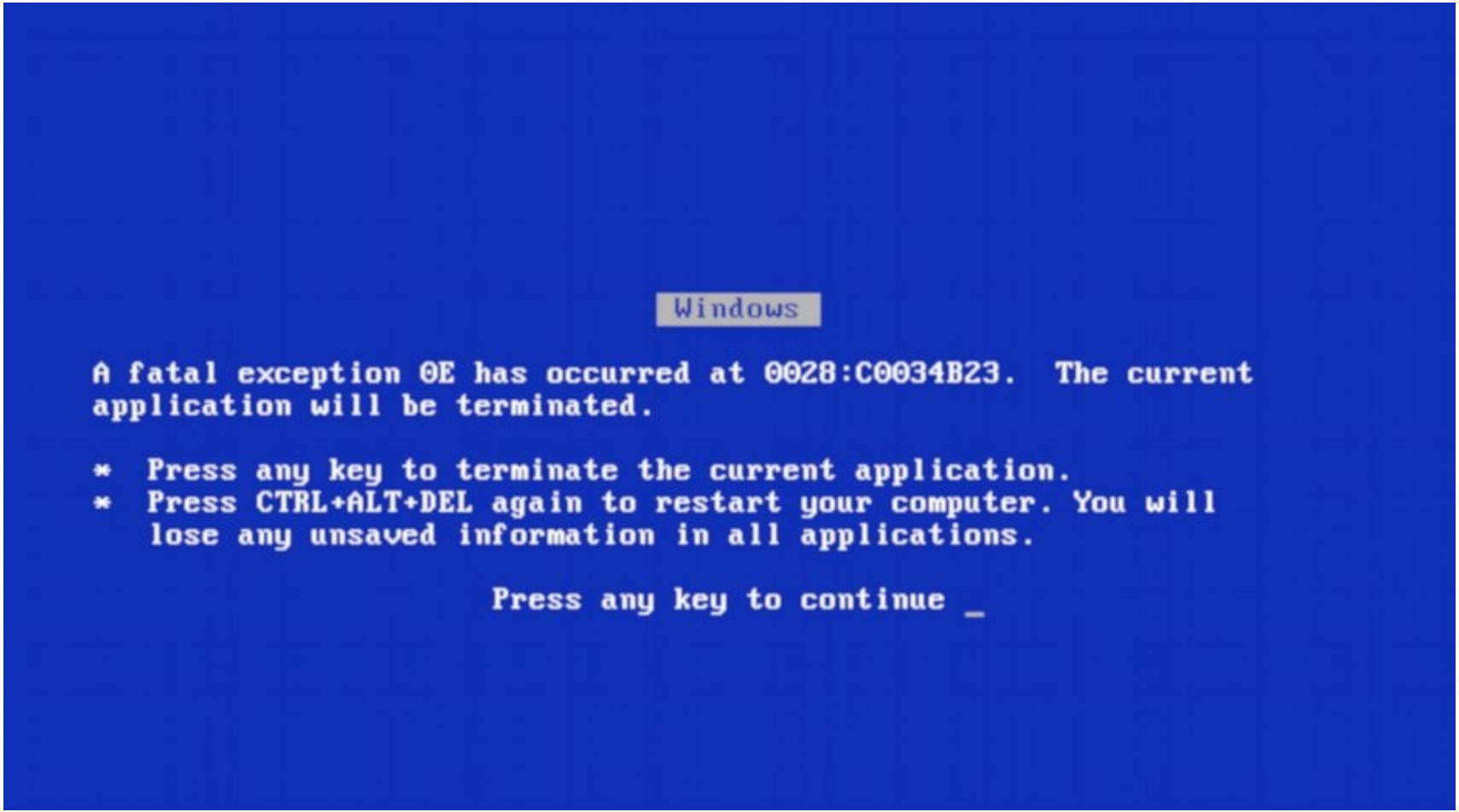

Windows
A fatal exception 0E has occurred at 0028:C0034B23. The current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.
Press any key to continue _

Other wiki dumps run but...

- One dump run = sql tables, revision metadata, revision content, logging data, site info
- Days to complete
- Must start over from the beginning!

Cascading failures?

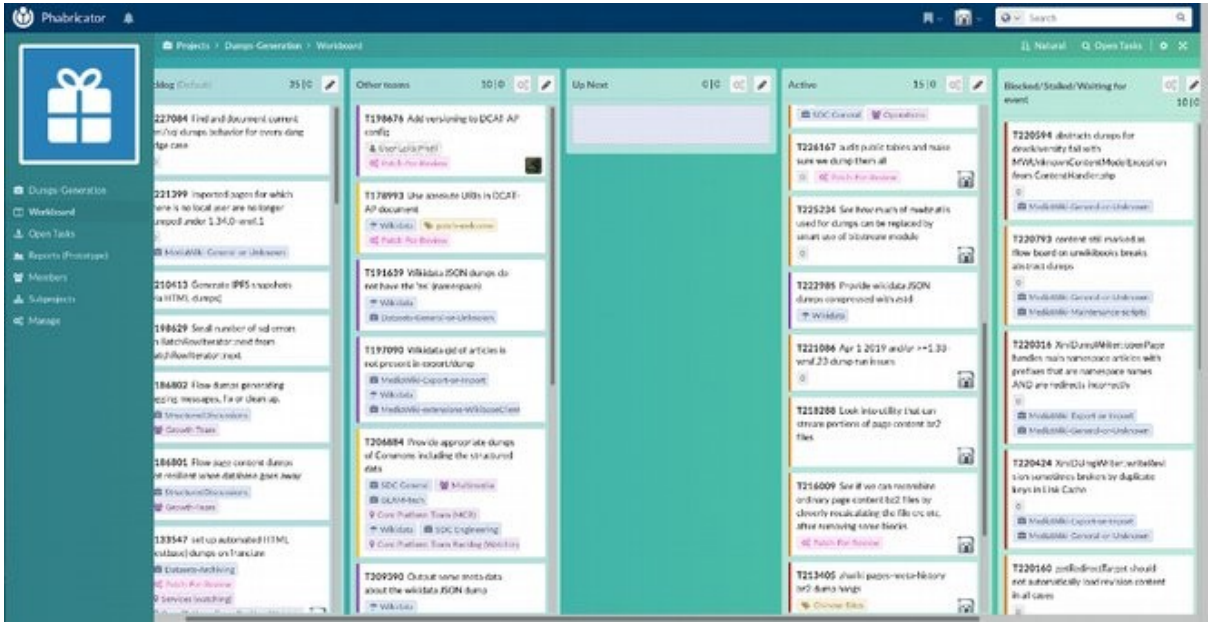
- We got ‘em right here.
- MediaWiki deploys, bad data, one dump job dependent on the next dependent on the next, etc.


Windows
A fatal exception 0E has occurred at 0028:C0034B23. The current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.
Press any key to continue _

Broken

Fixing bugs is interrupt-driven

- Never time to do it right
- Always time to do it over
- And over…and over… and over
- Cheapest approach: restart broken jobs by hand, check results periodically, rest of the time write and test code



Breaking up the monolith

- Each job got a separate status entry
- On rerun, successful jobs skipped
- Failed/waiting jobs rerun

Long-running jobs are long

- English language Wikipedia revision history content dump job: 16 days (2008!!)
- Proof: see the March 2010 archives on https://dumps.wikimedia.org

Break long jobs into pieces

- Revision history dumps… bz2 and 7z
- Current revision dumps
- Articles dumps
- Metadata dumps

But long-running jobs are still long

- 27 pieces, 10 days (2011!!)
- Proof: Aug 11 2011 archive.org dumps index.html for enwiki 20110722 run

Need moar pieces!

- ‘Checkpoint’ files: close a file after so many hours, open a new one
- Later: dump by page range, no checkpoints needed

Even moar pieces

- Abstracts and page logs split up
- Output for each stubs piece done in retryable batches
- Later: same for abstracts, logs

Reassembling the pieces

- Dd is your friend
- Gz multistream files
- Bz2 multistream files
- More in progress

On to the next problem: fairness

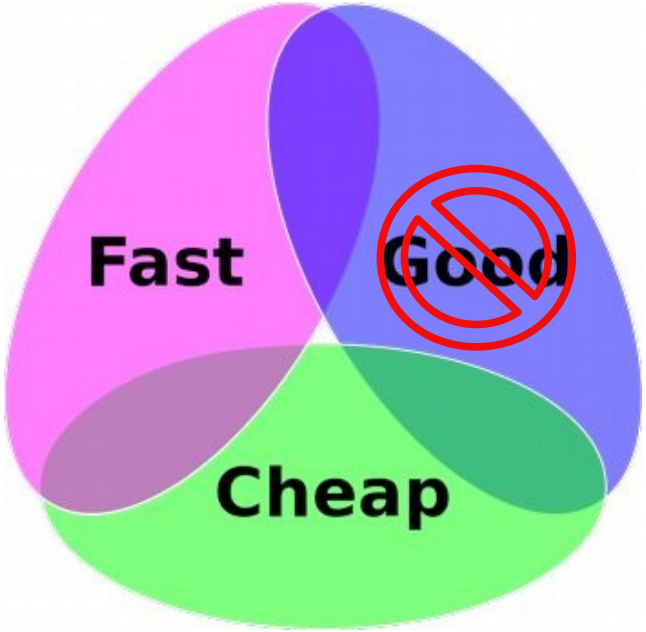
- Big wikis still take a long time
- Small wikis get blocked waiting for big wikis to finish
- Sadness!


HOT LUNCH
ALL DAY.
EXTRA BEER

Queues

Queues on the cheap

- Huge wikis in their own queues (en, wikidata)
- Big wikis in a list (de, ru, zh, fr…)
- Everything else in third queue
- ‘Everything’ jobs run before big wiki jobs, for each stage



Queues ~ hosts

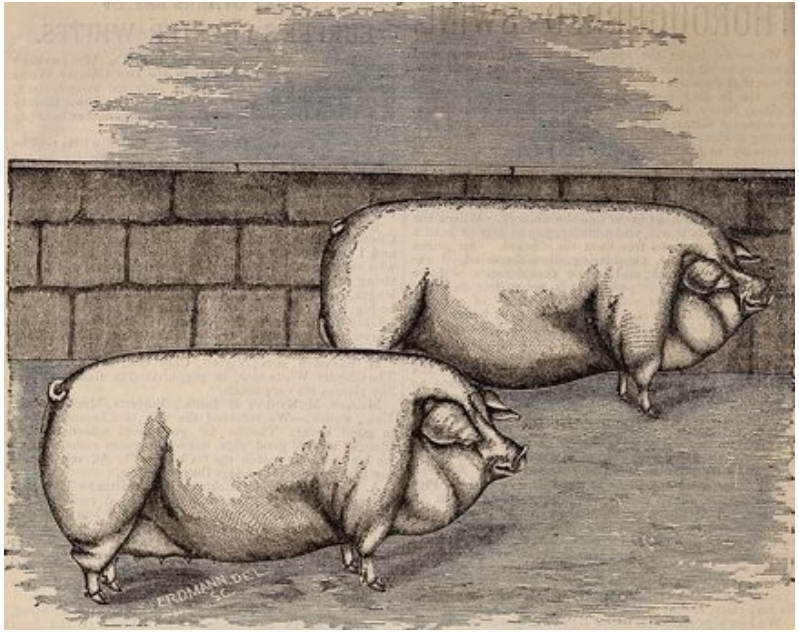
- Dedicated server for enwiki, runs other queue jobs when done
- Dedicated server for wikidata, same
- 3 more hosts that run small/big wiki jobs
- All wikis complete one job (or fail to max retries) before next one started

Dump runs and stages and jobs, oh my!

- Job: produce this set of files with a single command
- Example: stubs (revision metadata) for articles, current revs, all revs = ONE job
- Stage: run these jobs one after another, for a single wiki, then move to the next wiki
- Example: stubs, then sql tables = ONE stage

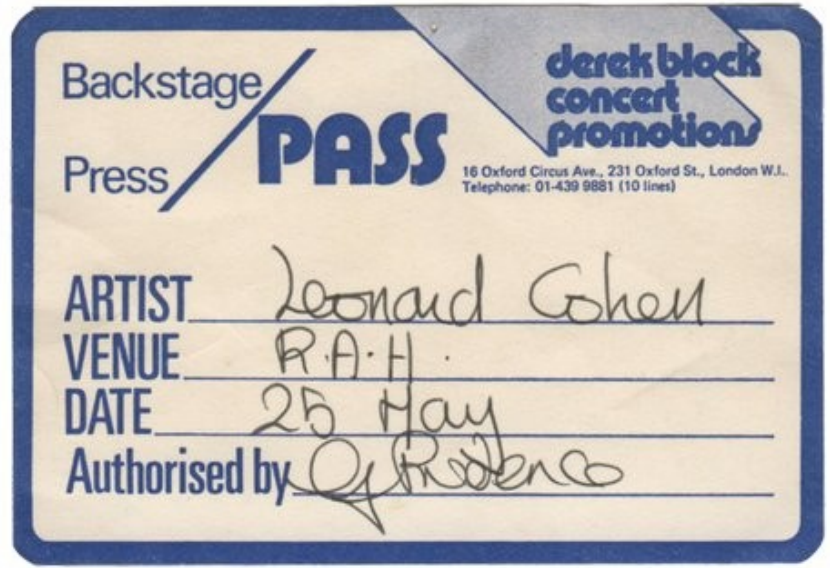
Not all dump runs are created equal

- ‘Full’ runs: Do every job we know about
- ‘Partial’ runs: skip the really long jobs (revision content for every revision ever) but still mark these dumps as ‘done’.
- Rerun from command line without need to know type of run



(Back)Stage details

- Stages are run by a per-server scheduler
- Config: List of possible stages in order, resources (cores) needed per stage, # workers to run per stage



When a worker completes a stage

- Stage finished for one wiki; worker checks for the next wiki.
- No wikis left? Worker exits
- Scheduler checks freed resources vs resources needed for next stage worker
- Resources available? Worker started
- Sometimes workers fail a stage
- Bad data, db server went away, etc
- After max retries, worker gives up on that wiki
- Maybe later problem is fixed
- Catch-all stage at the end runs everything, skipping completed jobs

Who schedules the scheduler

- One cron to rule them all
- Run twice a day in case a bad deploy kills off everything
- Only restart early enough in the run window that we can complete the run during window
- Why a bad deploy kills off everything:
- 911 failures would be a HUGE mess to clean up
- Die after n consecutive failures and wait for a human to fix things in the meantime

Workers

Reusing output

- Two-pass dumps
- 1. Generate metadata for all revisions wanted
- 2. Get content for each revision
- This means that content dump job requires access to ‘stubs’ (metadata output) from earlier job

More reuse during content dumps

- Naive approach: request content from the external storage database cluster for each revision. Expensive!
- Smart approach: check previous content dump, use revision content if present, otherwise query db. Cheap!
- We can do this because REVISION CONTENT NEVER CHANGES (in theory).
- Downsides: more cpu to decompress previous content dump

Status file updates, lock files

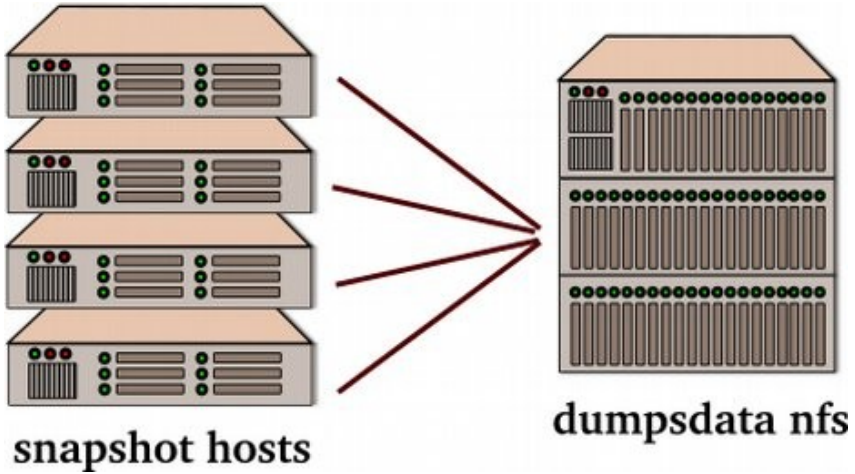
- One worker runs a job which fails; status file shows failure
- Another worker tries that job again and it's successful; same status file must be updated
- Lock files to prevent multiple workers from claiming the same wiki's run at the same time

“Solution”: NFS

- IOPS limits! Bandwith problems! Cache bugs! General weirdness! Here be dragons!
- It still gets the job done

Worker Hosts

- Worker hosts running dump scripts write to common NFS server
- Any worker host can run any job for any dump for any wiki
- Any worker host can verify results of a previous dump job for any run for any wiki
- Bottlenecks: network, nfs server disk iops



Helping NFS limp along

- Do's:
- Do limit bandwith on rsyncs (e.g. to failover hosts), running them sequentially and not in parallel
- Do run disk-intensive jobs, such as monthly statistics reporting, on a secondary host
- Do compress (almost) all output from the workers before writing it

- Don'ts:
- Don't NFS mount the filesystem for other users
- Don't provide web service to the world from the NFS server
- Don't rsync to third party mirror hosts from the NFS server
- (We used to do all of these things)

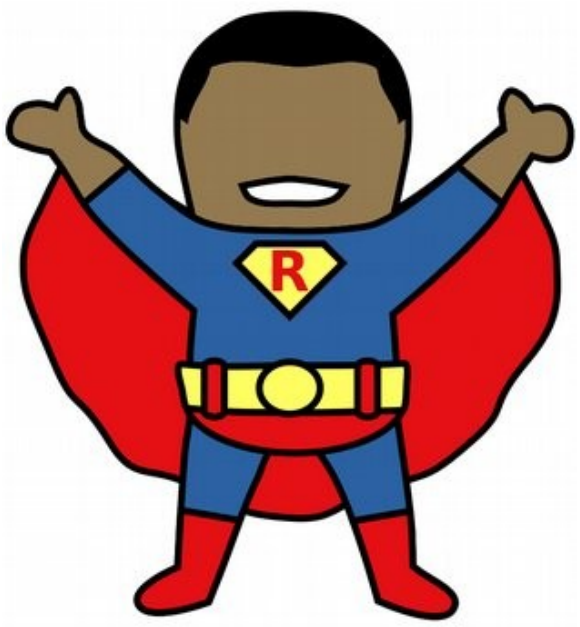
Monitoring

Once upon a time

- Dump output was written on one NFS server, also the public web server
- Index.html files? Write them at dump run time, instantly available
- Progress of a dump job? Write the message into the per-dump html file, instantly available
- Which jobs are complete? Write the info into a text file, instantly available
- But now we don’t do everything from one overloaded host. So...

Rsync to the rescue… or not

- We do rolling rsync of dump files as they are completed
- Rsync of html files before the files they link to are copied? Unhappy downloaders!
- Same for text file with info about complete dump jobs… if the job output isn't there yet.



Rsync for the impatient

- Step one: stash all html files with download links, job info files, into a tarball
- Step two: rsync dump job output files
- Step three: rsync html/job info tarball
- Step four: script checks every five minutes for new content, unpacks if needed
- Impatient downloaders may find files with no links, but not links with no files!

Provided for watchers

- ‘latest’ directory: dumps.wikimedia.org/<project>/latest/
- RSS files: <wiki>-latest-<jobname>.<type>.<ext>-rss.xml
- Symlinks to latest versions of dump jobs
- For each run: dumpstatus.json, dumpruninfo.json, dumpspecialfiles.json
- For each run: index.html, status.html, dumpruninfo.txt
- For each run: <wiki>-<date>-md5sums.json and .txt, sha1sums.json and .txt
- Centrally: index.json, backup-index.html, backup-index-test-bydb.html

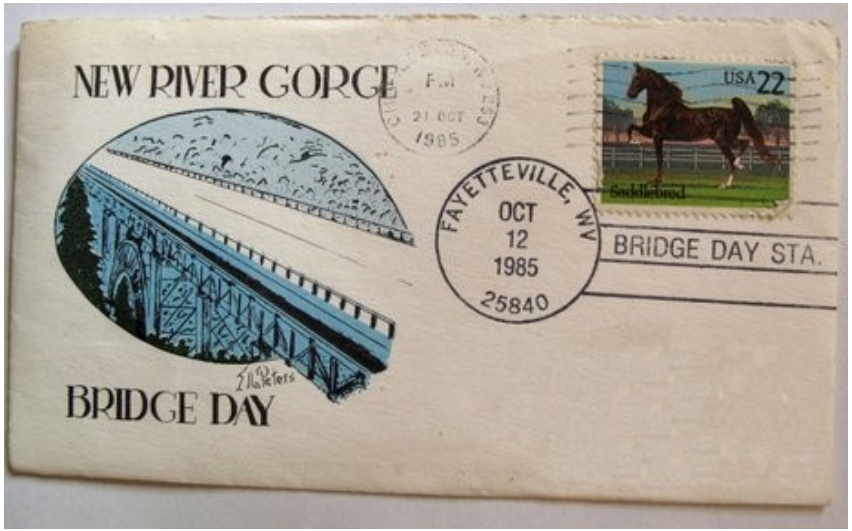
More

Stuff we dump

- Full xml/sql dumps, with all content for all versions of every page (whew!), monthly
- Partial xml/sql dumps, with current page versions only, monthly
- ‘Adds-changes’ dumps with new edits since the previous run, daily
- Wikidata entity dumps in json and rdf format, weekly
- Cirrus search indexes, suitable for loading into ElasticSearch, weekly
- Translation corpora by language pair, weekly
- Category dumps, Global block listings, Short url mappings, weekly
- ...and more!

You have mail

- Monthly FAQ email: uncompressed sizes of dumps for enwiki plus random other wiki
- Bimonthly run time checker for big wiki jobs, once for each run
- Job watcher notifies about stuck jobs (no output for longer than n hours)




ORGANIC PEROXIDE
5.2


ORGANIC PEROXIDE
5.2

Organically Grown

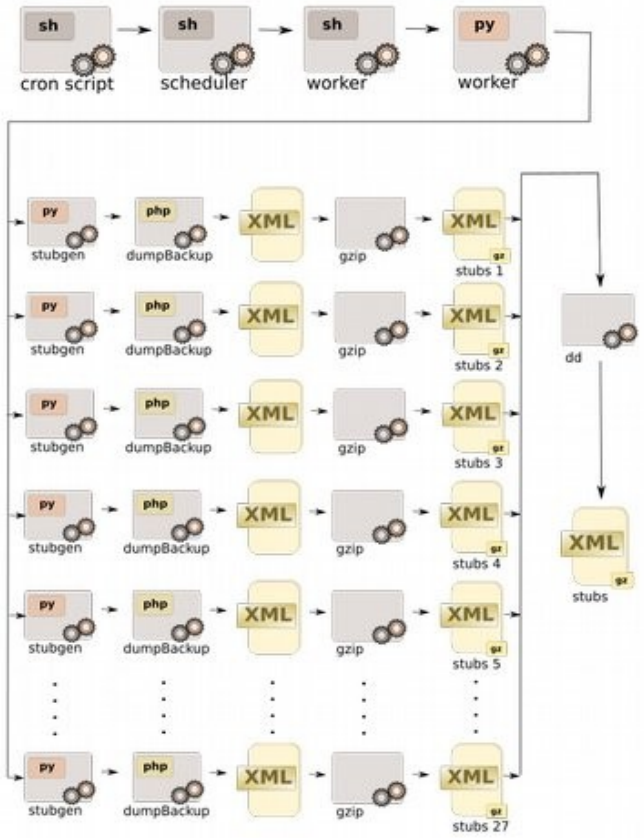
Organic growth pros

- Update code incrementally
- Improvements target area of biggest impact
- Faster to develop/deploy than major refactor
- Great if things already broken, need a quick fix

State of the code

- total scripts in dumps repo: 47 + 4
- lines of python in dumps repo: 17764
- lines of sh in dumps repo: 696
- lines of sh in puppet repo: 2445
- lines of python in puppet repo: 1119
- total lines of c in utils repo: 5756

State of the code: a simple visual aid



- Here's xml stub generation. I left out the python dump library scripts and the C MediaWiki/bz2 utilities.
- Page logs and abstracts also look like this.
- Then there's the revision content dumps, the sql table dumps, the site and namespace information dumps, the json status files, the adds/changes dumps, the page title dumps
- And the page rebalance script, the cleanup script, the monitoring script, the admin script, the rsync script, the job watcher script, the stats collection script, the...


KitchenAid

Nitty gritty details

Crunching, crunching, crunching

- 911 wikis dumped per run
- 4 servers, 32 cores, 64GB RAM do the work
- One job: (decompression|cat) | stuff | compress
- 27 jobs per host, leave a few spare cores for puppet etc

Dumping tables

- mysqldump for fully public tables
- DB server chosen via getReplicaServer.php
- DB creds obtained via getConfiguration.php
- Tables with some private fields are always dumped via MediaWiki

Dumping XML content

- Two passes, one for metadata (cheap), one for content (expensive)
- XML + bz2 output; lots of tools for hadoop etc
- Metadata is enough for setting up a mirror but...
- Not a perfect fork, missing user info etc.

Making jobs small (stubs)

- Small page range is dumped to a flat file w/o header or footer
- On success, file is fed to a gzipper that writes the final file
- On failure of the small page range, retries
- For big wikis, stubs are done in multiple pieces by parallel processes; if one fails, only it retries
- Consistency note: version of MW can change from one small page range to the next! But we write only one version header.

Making jobs small (content)

- Precompute page ranges for content jobs
- Write temp stubs covering these ranges
- Write content with retries for retrieval of any single piece of content
- MediaWiki scripts must catch all the exceptions!

Recombining stubs, content

- Big wikis produce multiple files; (some) downloaders want one
- Gz files can be recombined, skipping headers and footers, by careful use of dd
- BZ2 files must be decompressed (but there is a plan)

Parallel processes per job

- Big wikis don’t complete fast enough without multiple processes running
- Multiple processes for a dump step are run on the same host. For now.
- 27 processes each for wikidatawiki, enwiki
- Full run for wikidata: 19 days. For enwiki: 10!

Python scripts birds-eye view

- monitor.py: checks for stale lock files, writes the main index.html file based on per-run files, uses dumps library of modules
- worker.py: loops through all wikis or runs one, one job or many or all that are configured, uses dumps library
- Adds-changes has its own script, also uses dumps library
- Bash wrappers run monitor.py and worker.py in endless loops with wait intervals
- --dryrun is your friend
- Some scripts call others (stubs, abstracts, page logs) to dump small page ranges; pass –dryrun to these too
- Bash fixup scripts to fill in page content, 7z recompression, hash files for these

Python scripts a bit more

- Each dump job (abstracts, stubs etc) is a class inheriting from the Dump class.
- A DumpItemList is a list of all possible jobs for the wiki based on configuration.
- Items in this list are marked to run depending on whether specific jobs were passed in to worker.py
- The Runner class does runner prep, pre-dump work, loops through pre-job work, running the job, and post-job work for each job, then post-dump work to wrap up.
- Page content dumps are the most complex due to precomputation of page ranges and prefetching previous dump content.
- Page range info for page content dumps and configuration settings for the run are written to disk and re-used on retry.

Python scripts: a tiny bit more

- Many jobs use output from other jobs as input
- Some jobs clean up existing files before retries
- Output files need symlinks to ‘latest’ directory, rss feeds, listing in index.html
- Files may be generated with or without pageranges in the name, and with or without ‘part numbers’ (for parallel produced files) in the name.
- TL;DR: there are many file listing methods, listing possible or existing files. Classes FileLister and OutputFileLister. Some jobs subclass the latter.

mwbzutils: C utils

- A package of small utilities for manipulating xml and bz2-compressed files, with the capability to:
- Check that the last bz2 block of a file is intact
- Dump the last bz2 block
- Dump a bz2 file from some offset
- Split up an xml file into smaller ones with specified page ranges (used for temp stubs generation)
- Find a specific page in a bz2 file if present, displaying the bz2 block offset
- Read xml pages from stdin, write a recompressed file with multiple bz2 “streams”, with a fixed number of pages per stream, and an index of page to bz2 block

MediaWiki scripts drive-by

- Maintenance scripts (maintenance, maintenance/includes):
- dumpBackup.php + BackupDumper.php (stubs for two-pass dumps, content for third parties for single-pass dumps
- dumpTextPass.php + TextPassDumper.php (content for two-pass dumps)
- writers, compressors, filters, core dumps scripts

Writers

- includes/export
- DumpOutput.php (base class)
- DumpStringOutput.php (append contents to $output)
- DumpFileOutput.php (write to a file)
- DumpPipeOutput.php (yes, writes to a pipe)
- DumpMultiWriter.php (writes to multiple outputs)

Filters

- includes/export/
- DumpFilter.php: base class
- DumpLatestFilter.php: write only the most current revision (via page_latest)
- DumpNamespaceFilter.php: write only the specified namespaces
- DumpNotalkFilter.php – write everything but talk pages
- ExportProgressFilter.php - ability to write occasional progress lines to console

Compressors

- includes/export/
- DumpFileOutput.php - base class, the rest of these do what you would expect
- Dump7ZipOutput.php
- DumpBZip2Output.php
- DumpDBZip2Output.php
- DumpGZipOutput.php
- DumpLBZip2Output.php: uses only one core (-n 1)

Base dump scripts + misc

- includes/export/
- WikiExporter.php - base export code, dump by page range, single pages, lists of pages, revision range, etc.
- XmlDumpWriter.php - write xml for stubs/content/logs/abstracts
- BaseDump.php - prefetch content
- SevenZipStream.php - open 7z files as input (used for prefetch)

Extensions

- Flow: separate maintenance scripts, separate db
- ActiveAbstracts: plugin to dumpBackup.php for generating text snippets from articles
- Wikibase: weekly entity dumps, wb tables
- MediaInfo: weekly commons structured data dumps, core tables

Abstract dumps sample ps

```
  0   0:12 /usr/sbin/smartd -n
  0   8:48 /usr/sbin/cron -f
  0   0:00  \_ /usr/sbin/CRON -f
400   0:00  |   \_ /bin/sh -c /usr/local/bin/fulldumps.sh 20 25 regular partial 28 > /dev/null
400   0:00  |       \_ /bin/bash /usr/local/bin/fulldumps.sh 20 25 regular partial 28
400   0:32  |           \_ /usr/bin/python3 /srv/deployment/dumps/dumps/xmldumps-backup/dumpschedule
400   0:00  |               \_ /bin/sh -c /bin/bash ./worker --skipdone --exclusive --log --configfi
400   0:00  |               |   \_ /bin/bash ./worker --skipdone --exclusive --log --configfile /etc
400   3:12  |               |       \_ python3 ./worker.py --configfile /etc/dumps/confs/wikidump.co
400   0:07  |               |           \_ /usr/bin/python3 xmlabstracts.py --config /etc/dumps/conf
400   0:00  |               |               \_ /bin/sh -c gzip >> /mnt/dumpsdata/xmldatadumps/public
400   0:12  |               |               |   \_ gzip
400   0:16  |               |               \_ /usr/bin/php7.2 /srv/mediawiki/multiversion/MWScript.
400   0:00  |               \_ /bin/sh -c /bin/bash ./worker --skipdone --exclusive --log --configfi
400   0:00  |               |   \_ /bin/bash ./worker --skipdone --exclusive --log --configfile /etc
400   2:33  |               |       \_ python3 ./worker.py --configfile /etc/dumps/confs/wikidump.co
400   0:03  |               |           \_ /usr/bin/python3 xmlabstracts.py --config /etc/dumps/conf
400   0:00  |               |               \_ /bin/sh -c gzip >> /mnt/dumpsdata/xmldatadumps/public
400   0:04  |               |               |   \_ gzip
400   0:03  |               |               \_ /usr/bin/php7.2 /srv/mediawiki/multiversion/MWScript.
400   0:00  |               \_ /bin/sh -c /bin/bash ./worker --skipdone --exclusive --log --configfi
400   0:00  |               |   \_ /bin/bash ./worker --skipdone --exclusive --log --configfile /etc
400   3:27  |               |       \_ python3 ./worker.py --configfile /etc/dumps/confs/wikidump.co
400   0:03  |               |           \_ /usr/bin/python3 xmlabstracts.py --config /etc/dumps/conf
400   0:00  |               |               \_ /bin/sh -c gzip >> /mnt/dumpsdata/xmldatadumps/public
400   0:03  |               |               |   \_ gzip
400   0:22  |               |               \_ /usr/bin/php7.2 /srv/mediawiki/multiversion/MWScript.
400   0:00  |               \_ /bin/sh -c /bin/bash ./worker --skipdone --exclusive --log --configfi
400   0:00  |               |   \_ /bin/bash ./worker --skipdone --exclusive --log --configfile /etc
```

Multistream dumps sample ps

```
111 28246:31 /usr/bin/prometheus-node-exporter --collector.buddyinfo --collector.conntrack --collect
  0   0:00 /usr/sbin/mcelog --daemon
  0   5:49 /usr/sbin/cron -f
  0   0:00  \_ /usr/sbin/CRON -f
400   0:00      \_ /bin/sh -c /usr/local/bin/fulldumps.sh 20 25 wikidatawiki partial 28 > /dev/null
400   0:00          \_ /bin/bash /usr/local/bin/fulldumps.sh 20 25 wikidatawiki partial 28
400   0:06              \_ /usr/bin/python3 /srv/deployment/dumps/dumps/xmldumps-backup/dumpschedule
400   0:00                  \_ /bin/sh -c /bin/bash ./worker --skipdone --exclusive --log --configfi
400   0:00                      \_ /bin/bash ./worker --skipdone --exclusive --log --configfile /etc
400 1242:09                           \_ python3 ./worker.py --configfile /etc/dumps/confs/wikidump.c
400   0:00                              \_ /bin/sh -c /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/pub
400  90:52                              |   \_ /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/public/wik
400 605:38                              |   \_ /usr/local/bin/recompressxml --pagesperstream 100 --b
400   0:00                              \_ /bin/sh -c /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/pub
400  92:16                              |   \_ /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/public/wik
400 602:45                              |   \_ /usr/local/bin/recompressxml --pagesperstream 100 --b
400   0:00                              \_ /bin/sh -c /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/pub
400  93:41                              |   \_ /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/public/wik
400 604:32                              |   \_ /usr/local/bin/recompressxml --pagesperstream 100 --b
400   0:00                              \_ /bin/sh -c /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/pub
400  88:01                              |   \_ /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/public/wik
400 608:55                              |   \_ /usr/local/bin/recompressxml --pagesperstream 100 --b
400   0:00                              \_ /bin/sh -c /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/pub
400  85:59                                  \_ /bin/bzip2 -dc /mnt/dumpsdata/xmldatadumps/public/wik
400 609:00                                  \_ /usr/local/bin/recompressxml --pagesperstream 100 --b
```

A few more considerations

- Slow queries on the vslow db servers where dumps run, can significantly slow down the run
- MediaWiki exceptions must be caught in XmlDumpWriter or we have broken dumps
- Dumps should NEVER write to the database
- (* cough * wb_terms drop * cough *)
- We dump billions of revisions: a small bit of additional processing means a lot more time for the run

Testing

- TL;DR: Needs improvement.
- Some python and php unit tests, need more
- Dump testing in deployment-prep of master for smoke test (deployment-snapshot01)
- Use a custom config on the snapshot testbed and run against the production dbs as dumpsgen user, writing to a temp area
- Run some special very gross dump output comparison scripts on laptop

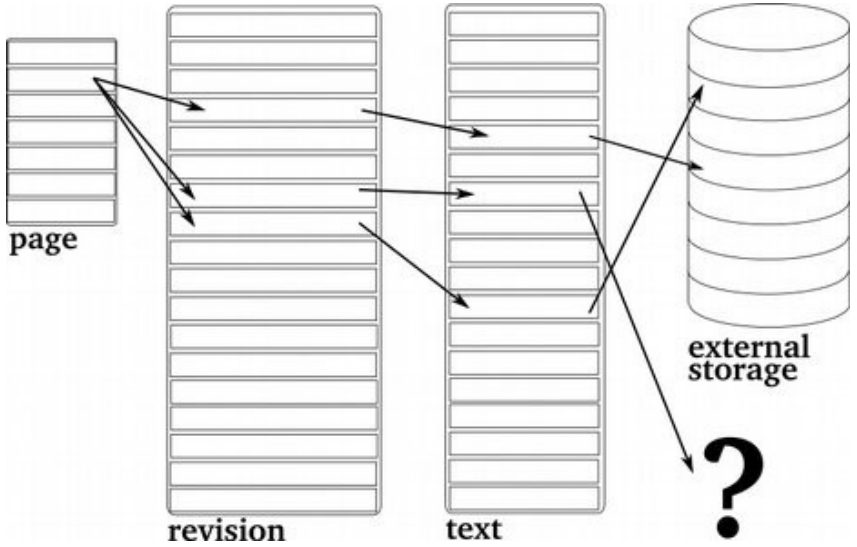
Dirty Data

Neither snow nor rain nor heat nor gloom of night...

- 19 years of data
- 19 years of MediaWiki commits… and bugs!
- Dumps must be resilient against all errors for isolated revisions:
- Addresses pointing to nowhere, empty user fields, revision texts bigger than the current limit, broken serialization…

Text addresses pointing to nowhere

- Each page record points to several revision records
- Each revision record points to one text record
- Each text record points to an entry in the external storage cluster…
- Or not?!



Names and Namespaces

```
wikiadmin@10.64.48.35(sawikisource)> select page_id, page_namespace, page_title, page_latest, page_is_redirect from page where page_id = 8821;
+---------+----------------+-----------------------------------------+-------------+------------------+
| page_id | page_namespace | page_title                              | page_latest | page_is_redirect |
+---------+----------------+-----------------------------------------+-------------+------------------+
|    8821 |            104 | Kumarasambhavam_-_Mallinatha_-_1888.djvu/5 |      157361 |                0 |
+---------+----------------+-----------------------------------------+-------------+------------------+
```

```
wikiadmin@10.64.48.35(sawikisource)> select page_id, page_namespace, page_title, page_latest, page_is_redirect from page where page_id = 8829;
+---------+----------------+-------------------------------------------------------------------------+-------------+------------------+
| page_id | page_namespace | page_title                                                              | page_latest | page_is_redirect |
+---------+----------------+-------------------------------------------------------------------------+-------------+------------------+
|    8829 |              0 | पृष्ठम्:Kumarasambhavam_-_Mallinatha_-_1888.djvu/5                       |       28418 |                1 |
+---------+----------------+-------------------------------------------------------------------------+-------------+------------------+
1 row in set (0.00 sec)
```

Prefix for namespace 104: पृष्ठम्
Full name of first page: पृष्ठम्:Kumarasambhavam_-_Mallinatha_-_1888.djvu/5
Full name of second page: yup, same as the first page

Huge revisions are huge

- Limit in configuration files (KB):
- $wgMaxArticleSize = 2048;
- Max revision size in the database?
- Over 10 MB!
- Don’t try this at home:
- https://en.wikipedia.org/w/index.php?
- title=User:ColenFace&
- oldid=39456244

User names vs. IPs vs. nothing


```
wikiadmin@10.64.32.136(zhwikisource)> select rev_user, rev_user_text from revision where rev_id = 209483;
+----------+---------------+
| rev_user | rev_user_text |
+----------+---------------+
|     2632 | Wmrwiki       |
+----------+---------------+
1 row in set (0.00 sec)

wikiadmin@10.64.32.136(zhwikisource)> select rev_user, rev_user_text from revision where rev_id = 209484;
+----------+---------------+
| rev_user | rev_user_text |
+----------+---------------+
|        0 |               |
+----------+---------------+
1 row in set (0.00 sec)
```

Lessons learned after 10 years

- Everything is an edge case
- No really, everything
- You haven’t found all of the edge cases yet
- You will never find all of the edge cases
- This is your dev(ops) life

Consistency

The hobgoblin of little minds?

- MediaWiki version can change in the middle of stubs/content run
- A page can be moved in the middle of being dumped
- Revisions may be hidden after stubs dump and before content dump
- Namespace configuration can change in the middle of stubs/content run
- Tables aren’t locked while being mysqldump-ed.
- Dumps of one part of the database are done at different times than other parts, so they may reflect different states
- And much more...

Imports

Why import?

- Local testing of a new feature
- Having your own copy for research and analysis
- Sharing with a community that has slow/no Internet
- Added value (features or data not available on the original site, such as annotations or a different skin)
- Right to fork!

Official tools

- MediaWiki maintenance script: Import.php
- Slow.
- No, really. As molasses.
- The only tool expected to work (everything else may be outdated, missing fields, etc)
- Mwdumper (Java)
- Handles multiple versions
- Can write sql output instead of importing directly
- Often out of sync with current MediaWiki version
- Still slow, but better than Import.php

Needed for import

- Page, revision, text (and now slot, content, actor, comment) tables
- Related (page properties, page restrictions, restricted titles, categories, all link tables, users, user groups…)
- Wikidata items for wikis that include them
- Wikidata items for modules that render maps via WDQS
- Media locally uploaded and from Commons
- Some items not dumped, must be generated (user info)

Unofficial tools

- Various, generate revision and text tables from xml content dumps
- None deal with Wikidata inclusion
- None deal with maps rendering
- None handle slots and content tables
- None write out actor, comment tables

Alternate approaches

- Parse the dumps directly, for research or analysis [tools here please; mwparserfromhell?]
- Kiwix, XOWA, for offline readers
- HTML dumps (when they come!)
- …?

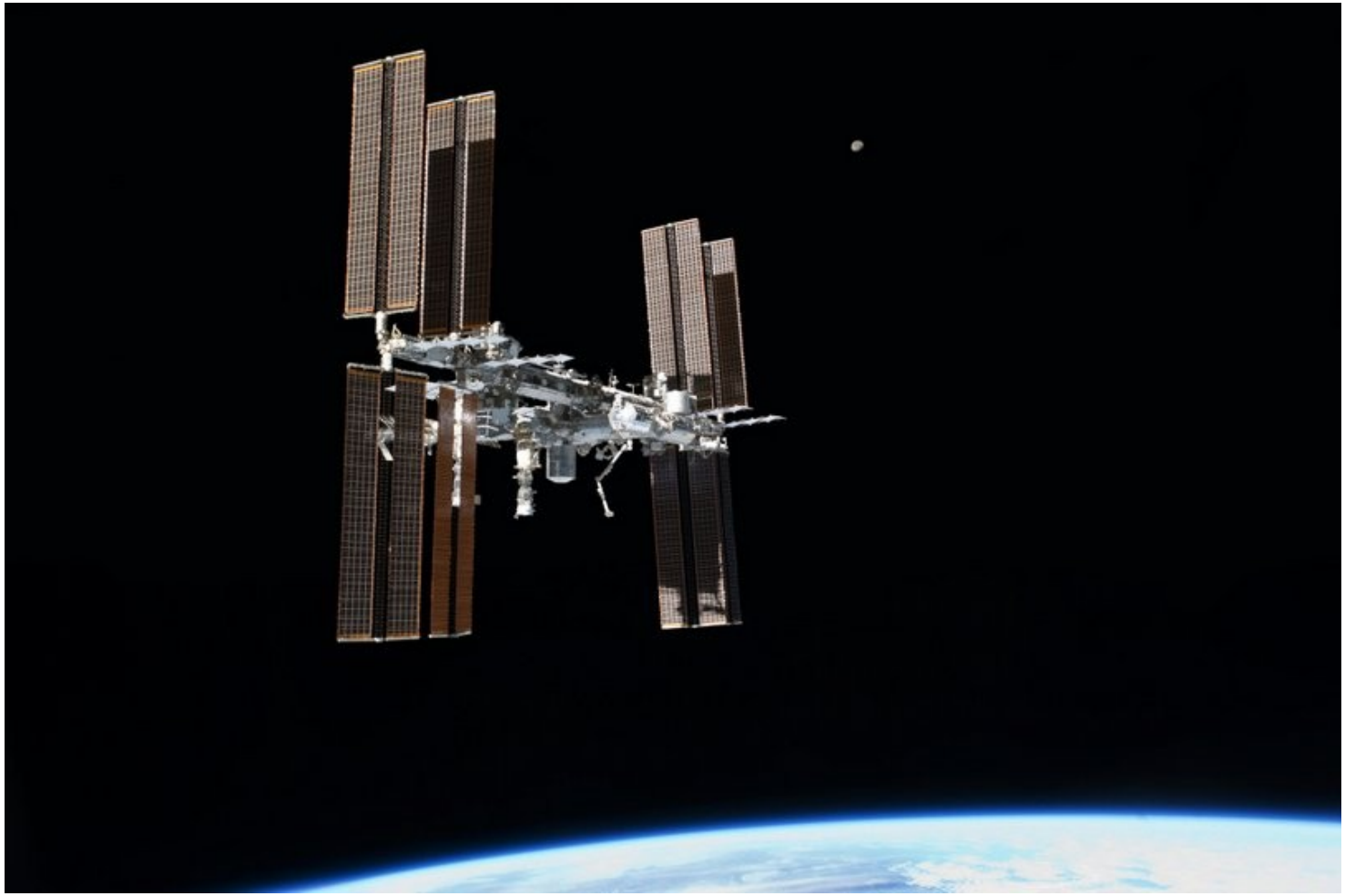
Future

Dump MOAR, finish on time

- English language Wikipedia is not small
- Wikidata is larger
- Commons is huge and Structured Data will make it a monster
- New tables as they are added

What about...

- More datasets?
- Media?
- HTML?
- Other formats?
- Importable sql, or downloadable mysql dbs?
- All the new stuff we haven't heard of yet?



Moar Future

Still moar to do

- make jobs 15 minutes long so reruns are cheap cheap cheap
- no-decompression bz2 file recombines (yes it can be done!)
- Easily expandable worker pool, no shuffling jobs and wikis around
- some fscking pylint and refactoring
- MORE MIRRORS


"LA PENSIEROSA"
Roberto Terracini 1938

Final Thoughts

Ten years of ‘learning experiences’:

- How the Mariadb optimizer lies and/or makes bad choices
- How weird bugs in MediaWiki core can go undiscovered for over a decade (the '===' bug)
- How Wikidata will eat us all for lunch
- How scale and edge cases are everything
- How to survive a in framework written for fast stateless web requests, with slow stateful dump jobs


ROUTE 66
ROUTE 66

The End

Credits: commons.wikimedia.org

- File:2013.08.03.195132_Catis_Dominoeffekt_Neustadt.jpg
- File:Amboy_(California,_USA),_Hist._Route_66_--_2012_--_1.jpg
- File:Ana_Dulce_F%C3%A9lix_Daegu_2011.jpg
- File:Bowery_men_waiting_for_bread_in_bread_line,_New_York_City,_Bain_Collection.jpg
- File:Burpee%27s_farm_annual_-_garden,_farm,_and_flower_seeds,_thoroughbred_stock_(1884)_(20484358136).jpg
- File:Children_watch_with_binoculars,_birdwatching.jpg
- File:ConCon_bsod.png

Credits, continued

- File:Construction_site_workers_loading_water,_sand,_ballast_and_cememt_into_a_concrete_mixer_in_Embu,_Kenya_6.jpg
- File:Custom-watch-clock-face-dial-wiki-attempt1.png
- File:Dirty_dishes.jpg
- File:Disassembled-rubix-1.jpg
- File:Dumps_logo_black_and_white.svg
- File:Gojira_1954_Japanese_poster.jpg
- File:Grand_Bazaar_Qarqan_Xinjiang_China_%E6%96%B0%E7%96%86_%E4%B8%94%E6%9C%AB_%E5%A4%A7%E5%B7%B4%E6%89%8E_-_panoramio.jpg

Return of the credits

- File:Herd_Of_Goats.jpg
- File:Humble_sink.jpg
- File:JackhammerDelhi.JPG
- File:KitchenAid_Sausage_Attachment.jpg
- File:La_Pensierosa.png
- File:Leonard_Cohen_Albert_Hall_Backstage_Pass_25_May_1976.png
- File:Martin_Jetpack_Unveiling,_Liftoff!_(2714934801).jpg
- File:New_River_Gorge_Bridge_Day_envelope_1985.jpg

Revenge of the credits

- File:Nurul_Izzah_reporters.jpg
- File:Organic_Peroxide.png
- File:Phodopus_sungorus_-_Hamsterkraftwerk.jpg
- File:Progress_MS-01_docked_to_ISS_(ISS046-E-043290).jpg
- File:Project-triangle.svg
- File:Pull_hair.jpg
- File:Reduce_Reuse_Recycle.jpg
- File:SFFf-1989091.162777_(4544868563).jpg
- File:Sisifus_the_faculties.jpg

Credits: the final chapter

- File:STS-135_final_flyaround_of_ISS_1.jpg
- File:Superman_Clipart.svg
- File:The_nickel_and_dime_store,_WPA_poster,_ca._1941.jpg
- File:The_Scream_Pastel.jpg
- File:Try_Again.jpg
- File:Ttukbaegi-spaghetti.jpg
- File:Wikimedia_Foundation_Servers-8055_01.jpg
- File:Wooden_hourglass_2.jpg

Questions?