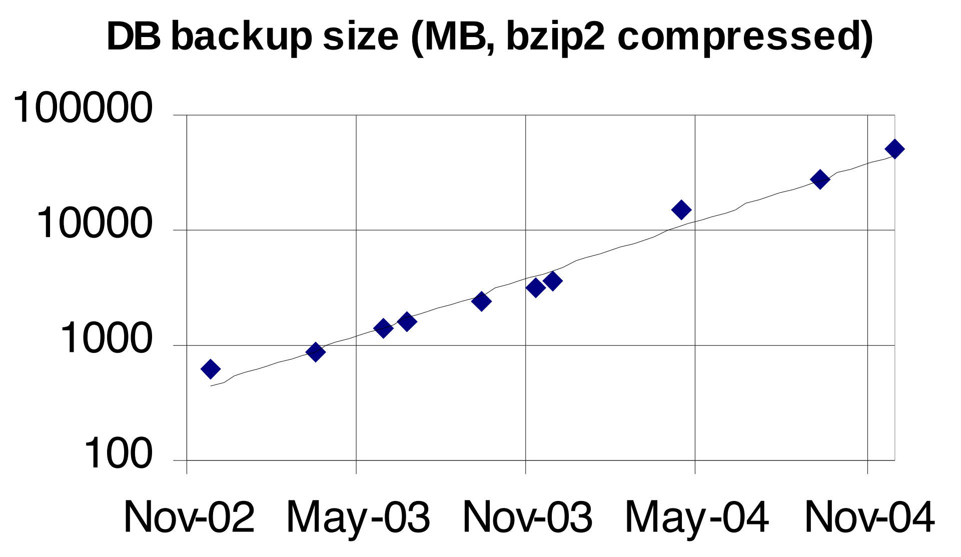
Wikipedia: Edit This Page

Differential Storage

Tim Starling

Wikipedia Growth

- Wikipedia and related projects have been growing at a phenomenal rate
- Database size doubles every 16 weeks



MediaWiki Design

- Based on the principle that hard drive space is cheap
- Minimal development time
- Each revision stored separately
  - Completely uncompressed until January 2004
  - Revisions now compressed with gzip for 50% saving
- Everything stored in MySQL – copy of every revision on every master or slave machine

Hardware Requirements

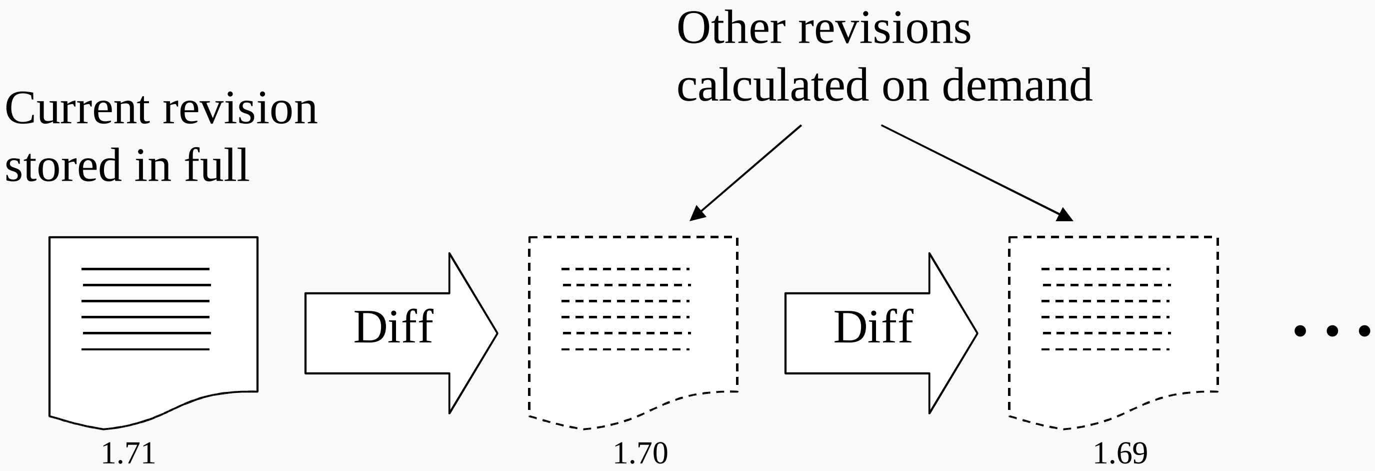
- Master DB server: ariel
- Worth $12,000
- Dual Opteron, 6x73GB 15K SCA SCSI drives: 4 RAID 1+0 (146GB), 2 RAID 1 (72GB)

Effective capacity	200 GB
Database size	171 GB

- No more drive bays available
- Only a week of growth left

Differential Storage

- Why not store diffs, instead of complete revisions?
- Canonical example: RCS



Other revisions calculated on demand

Current revision stored in full

Diff

Diff

. . .

1.71

1.70

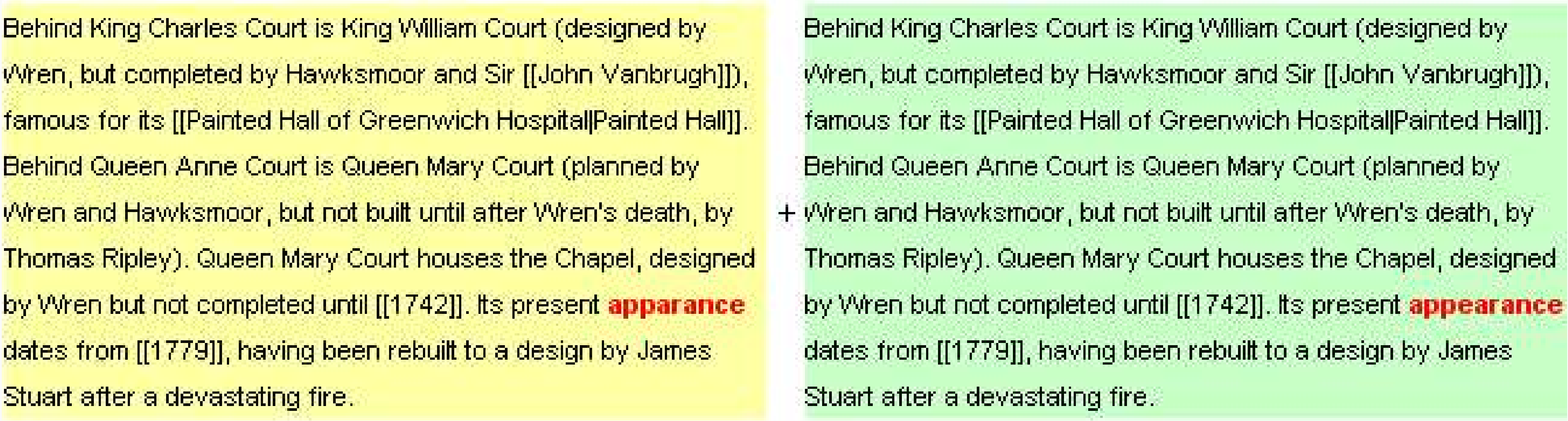
1.69

Differential Storage

- RCS:
  - is designed to store code
  - has a simple ASCII data format
- We want the best possible compression ratio
- No need for readability
- Can we do better than RCS?

Wiki Compared to Code

- Wikipedia articles have long lines, many minor changes are made



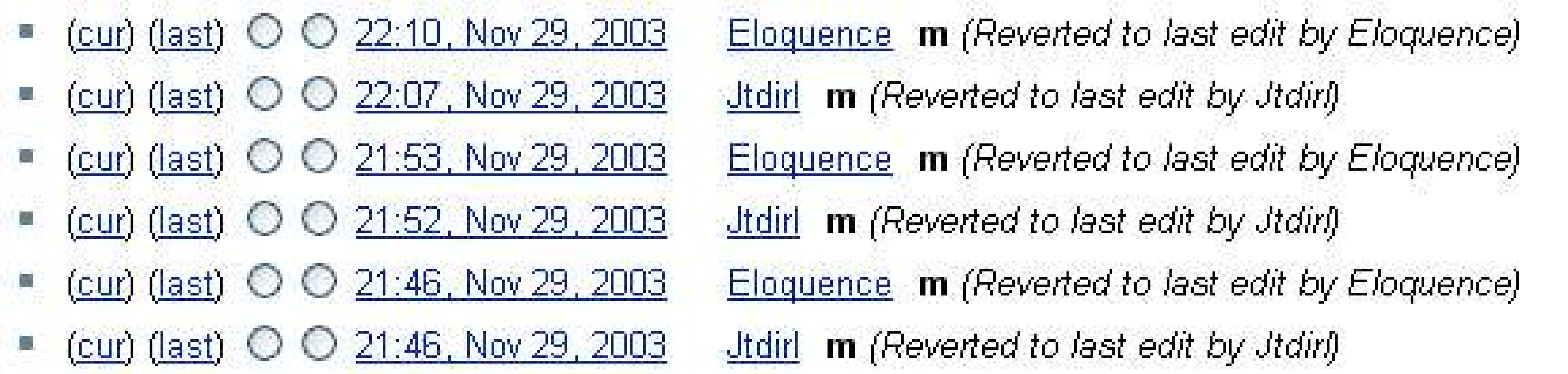
Behind King Charles Court is King William Court (designed by Wren, but completed by Hawksmoor and Sir [[John Vanbrugh]]), famous for its [[Painted Hall of Greenwich Hospital|Painted Hall]]. Behind Queen Anne Court is Queen Mary Court (planned by Wren and Hawksmoor, but not built until after Wren's death, by Thomas Ripley). Queen Mary Court houses the Chapel, designed by Wren but not completed until [[1742]]. Its present **apparance** dates from [[1779]], having been rebuilt to a design by James Stuart after a devastating fire.

+ Behind King Charles Court is King William Court (designed by Wren, but completed by Hawksmoor and Sir [[John Vanbrugh]]), famous for its [[Painted Hall of Greenwich Hospital|Painted Hall]]. Behind Queen Anne Court is Queen Mary Court (planned by Wren and Hawksmoor, but not built until after Wren's death, by Thomas Ripley). Queen Mary Court houses the Chapel, designed by Wren but not completed until [[1742]]. Its present **appearance** dates from [[1779]], having been rebuilt to a design by James Stuart after a devastating fire.

⇒Better if we don’t have to duplicate the whole line

Wiki Compared to Code

- Some articles have lengthy “edit wars”, where the article alternates between two significantly different versions.



- (cur) (last) ○ ○ 22:10, Nov 29, 2003 Eloquence **m** *(Reverted to last edit by Eloquence)*
- (cur) (last) ○ ○ 22:07, Nov 29, 2003 Jtdirl **m** *(Reverted to last edit by Jtdirl)*
- (cur) (last) ○ ○ 21:53, Nov 29, 2003 Eloquence **m** *(Reverted to last edit by Eloquence)*
- (cur) (last) ○ ○ 21:52, Nov 29, 2003 Jtdirl **m** *(Reverted to last edit by Jtdirl)*
- (cur) (last) ○ ○ 21:46, Nov 29, 2003 Eloquence **m** *(Reverted to last edit by Eloquence)*
- (cur) (last) ○ ○ 21:46, Nov 29, 2003 Jtdirl **m** *(Reverted to last edit by Jtdirl)*

- Can we store this efficiently?

Efficient Differential Storage

- What if someone moves a paragraph from one location to another? An ordinary diff won’t store that efficiently.

```
12,13d11
< [[Image:AndalusQuran.JPG|thumb|right|280px|[[12th
century]] [[Andalusia]]n Qur'an]]
<
17a16,17
> [[Image:AndalusQuran.JPG|thumb|right|280px|[[12th
century]] [[Andalusia]]n Qur'an]]
>
```

The LZ Connection

- What we need is an algorithm which will recognise arbitrary sequences of bytes in one revision which are repeated in another revision, and then encode them such that we only store the sequence once.
- This just happens to be what compression algorithms such as LZ77 do.

New Storage Scheme

- Concatenate a number of consecutive revisions
- Compress the resulting “chunk”
- A good compression algorithm will take advantage of the similarity between revisions, and achieve very high compression ratios

Proof of Principle

- We compressed history of three articles:
  - [[Atheism]], an article with lots of edit wars
  - [[Wikipedia:Cleanup]], a discussion page which is incrementally expanded
  - [[Physics]], a typical article with a long revision history
- Because all these articles have a very long revision history, we would expect better than average compression ratios

Proof of Principle

Size of the compressed text compared to the original text:

	gzip	bzip2	diff
Atheism	2.5%	2.3%	15.5%
Cleanup	2.5%	2.5%	1.1%
Physics	2.2%	2.4%	6.9%

- As expected, diffs performed poorly in the edit war case, but very well for incremental addition of text
- Compression methods always performed well

Gzip, Bzip2 and Diff

- Other tests showed bzip2 to give better compression than gzip, but at a much slower speed
- Ratio for diff could have been improved by choosing the most similar revision to take a diff against
- Diff much faster than gzip or bzip2
- Diff-based compression is harder to implement

Implementation

- We implemented a gzip method in MediaWiki 1.4
- Compression is taking place as I speak
- Expected effects:
  - Better utilisation of kernel cache
  - Higher I/O bandwidth for uncached revisions
  - Smaller DB size
- Average compressed size: ~15% of original
- Higher than the tests because the tests used articles with many revisions

Future Directions

- More detailed evaluation of diff-based methods
- Other ways to solve the space problem:
  - Application-level splitting across distinct MySQL instances
  - Distributed filesystems, e.g. GFS