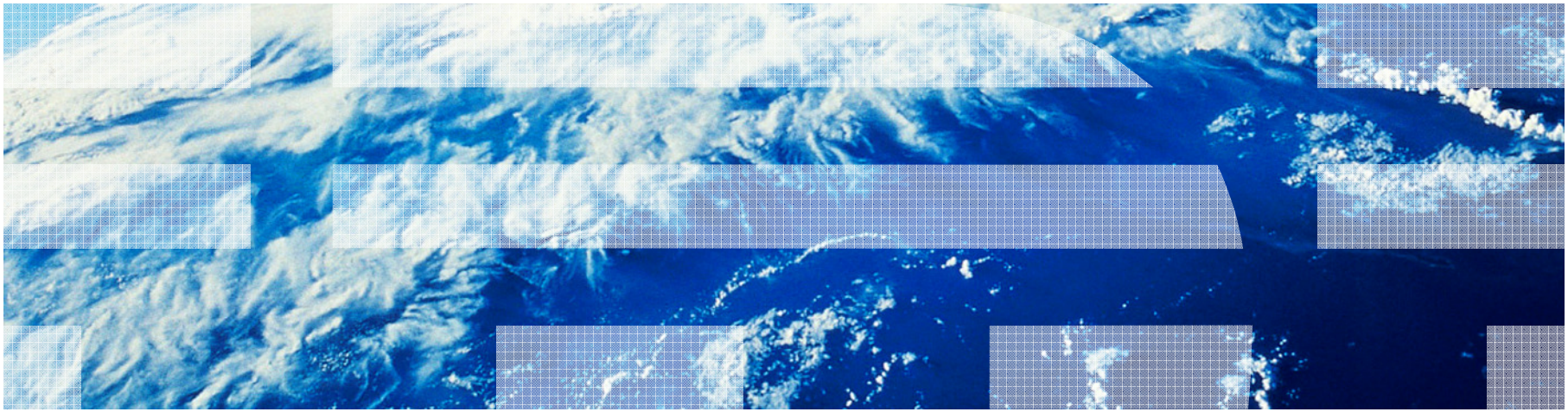
George Goldberg, Danny Harnik, Dmitry Sotnikov

5 June 2014

The Case for Sampling on Very Large File Systems

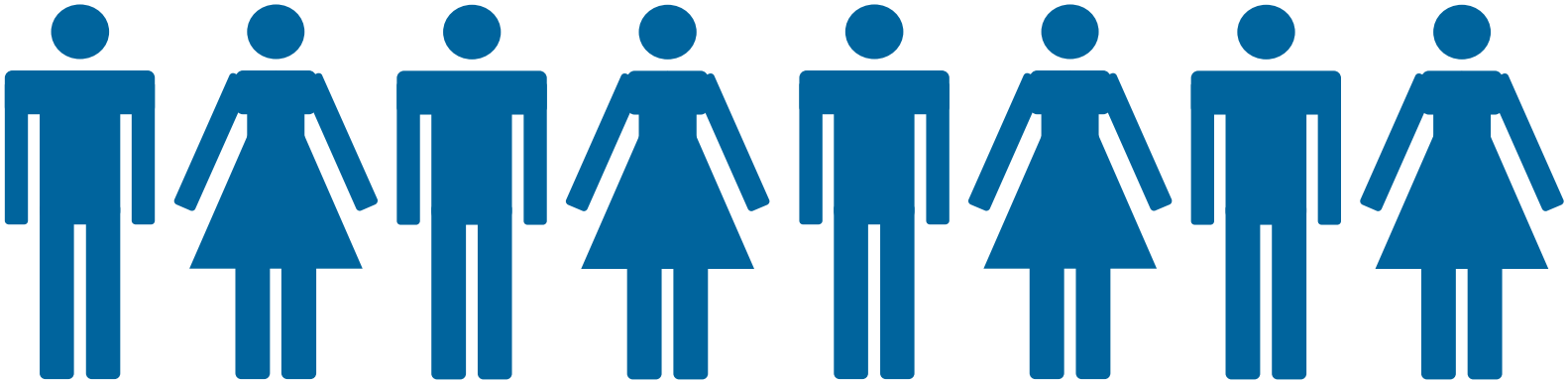


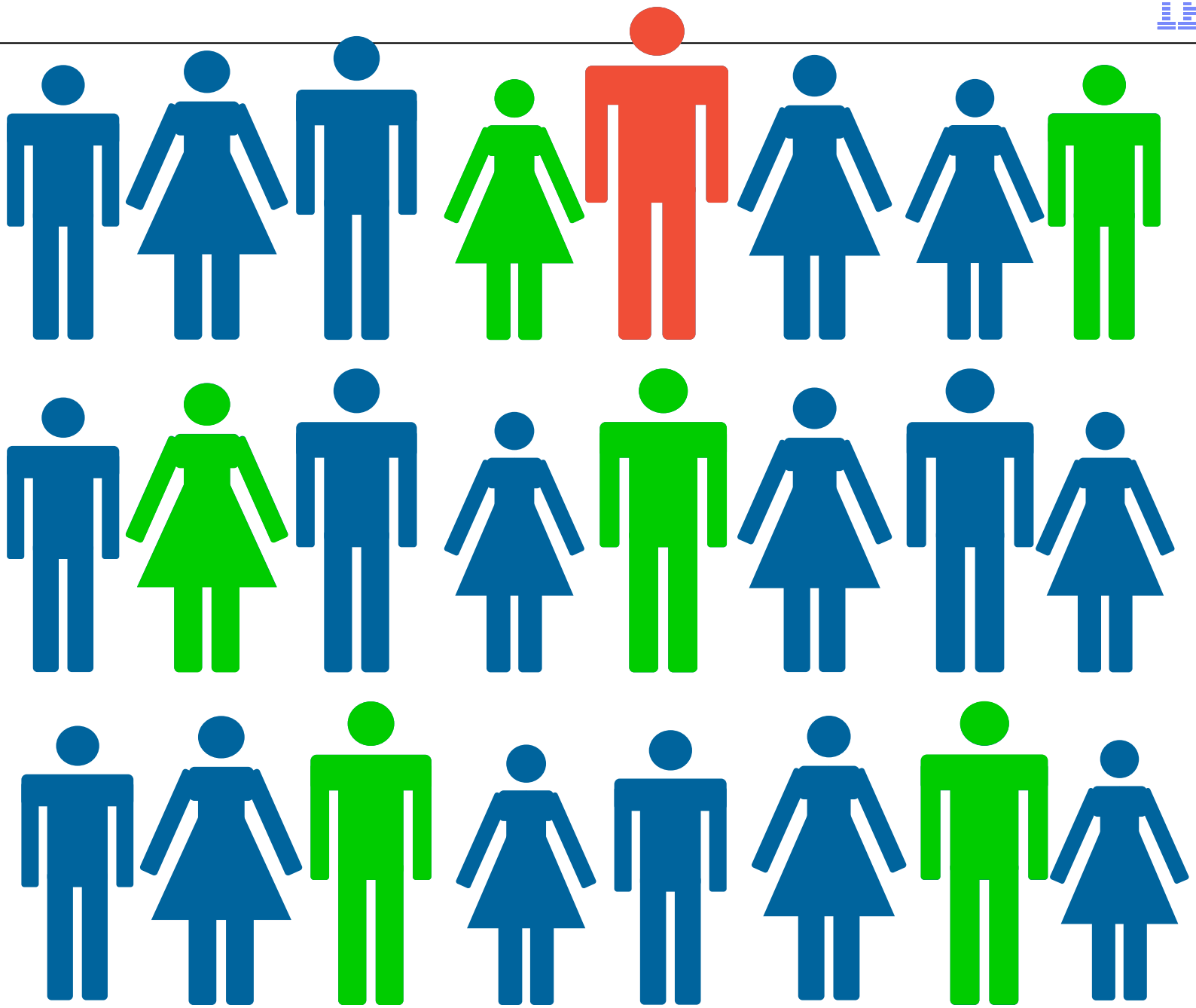
Sampling

- Sampling is broadly used in statistics and analytics, first and foremost for big amounts of data
- But sampling is hardly used for file systems

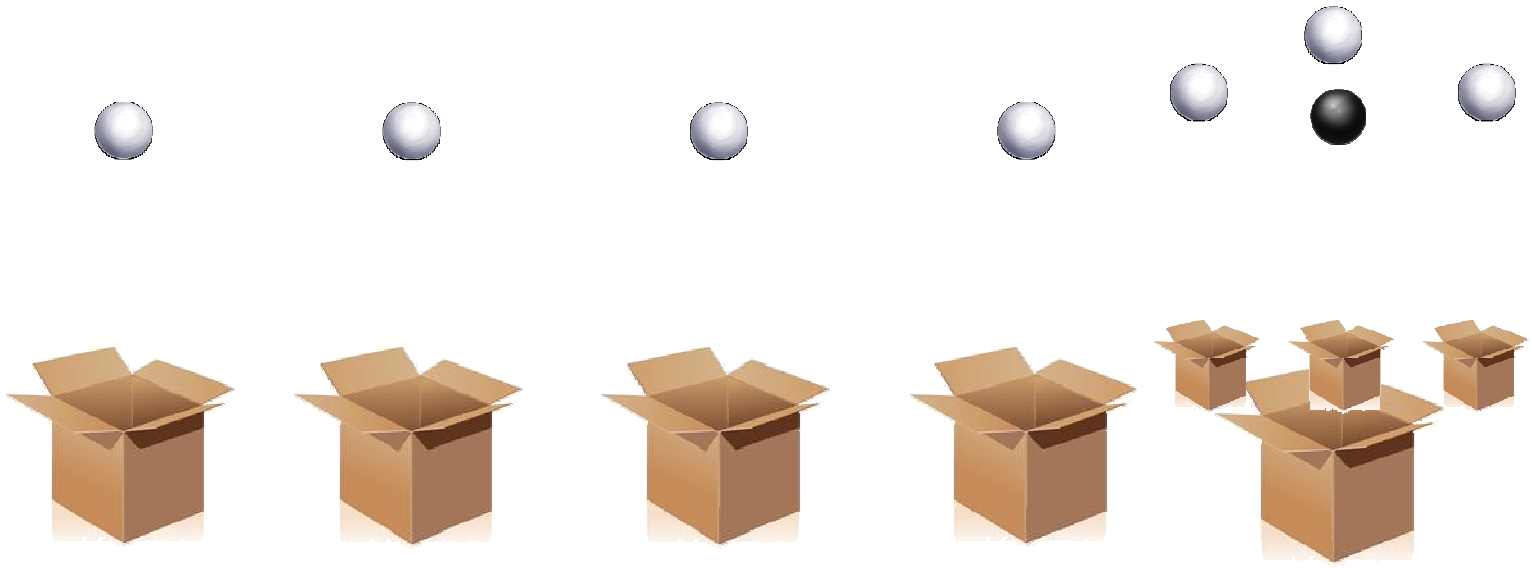
Why isn’t sampling used for file systems?

Sampling Example:





Another Example: How many black balls are in a box (on average)?



A full scan (in expectation) is required to find the black balls.
But a full scan can calculate the exact number of black balls.

No apparent benefit for sampling in file systems.

To Summarize:

- Full scan is required
  - Full directory tree traversal
- Naïve random sampling fails
  - due to the high variability of file sizes
  - E.g. 99% of the files account for less than 10% of the capacity

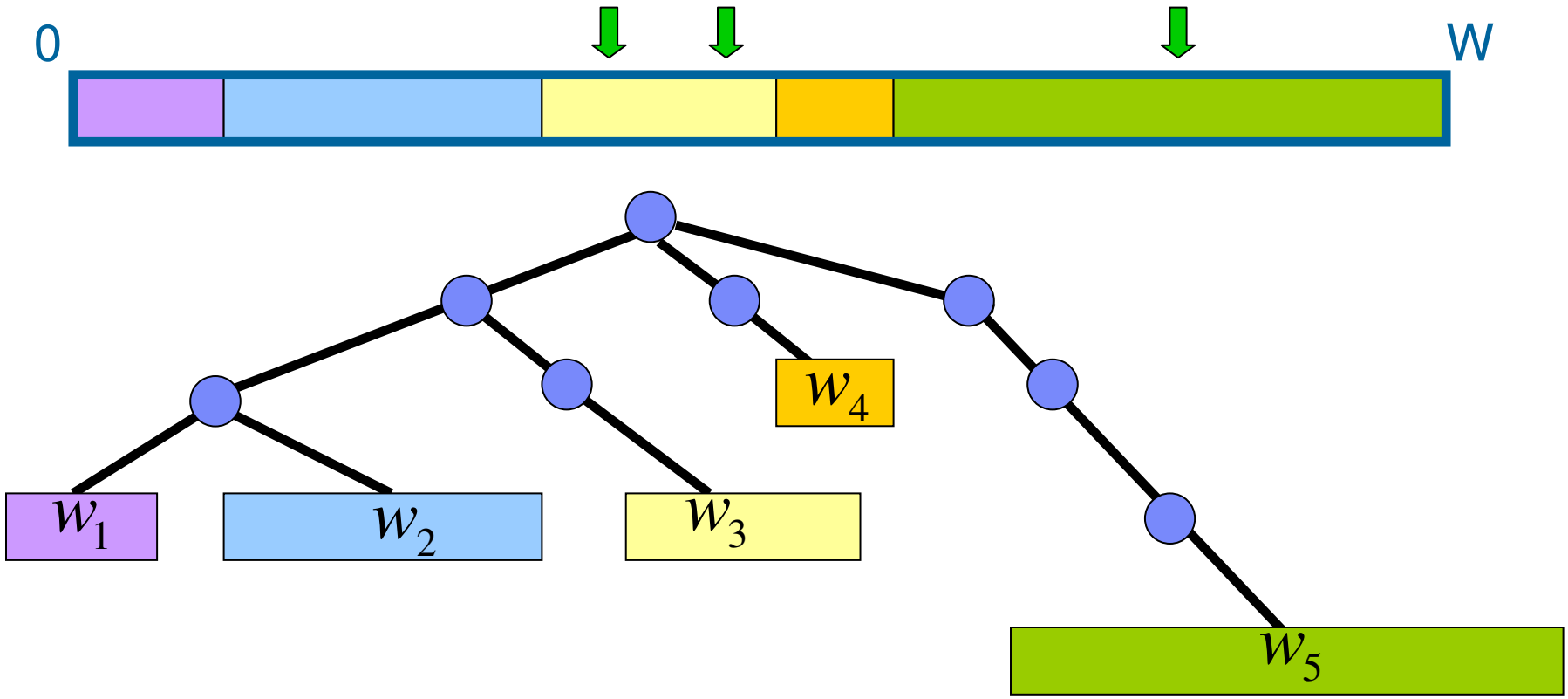
Why isn't sampling used for file systems?

1. No apparent benefit for sampling in file systems
2. Technical problems of sampling:
   - One pass
   - Randomness complexity
   - Large scale
   - Distributed environment

Remainder of this talk:

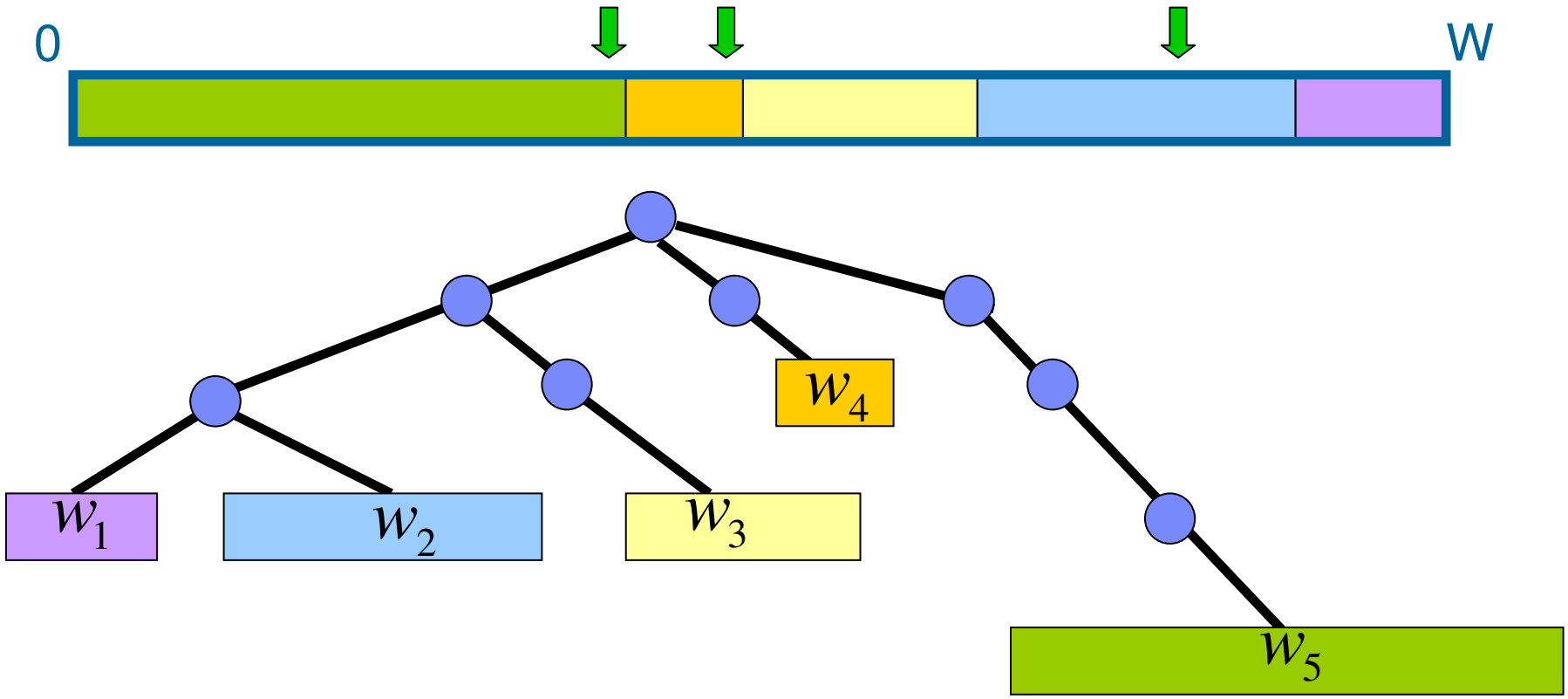
1. Solving the technical problems of sampling
2. Demonstrating beneficial real life use cases

Our Core Technique



Sample:

Our Core Technique



Sample:

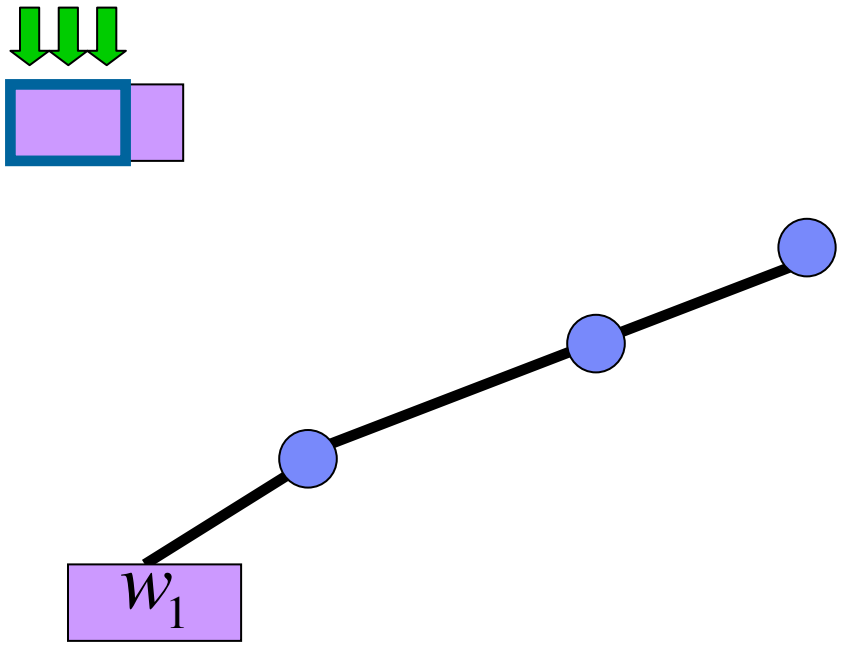
The Desired Distribution of File Appearances in the Sample

$$k_i \sim B(M, \frac{w_i}{W})$$

Is *W* always known in advance?

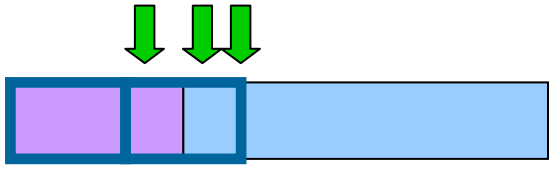
Computing *W* may require the a full scan by itself

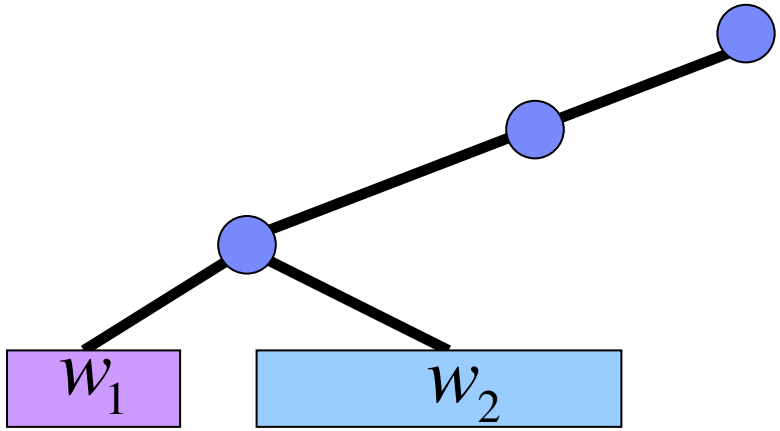
Our One Pass Solution



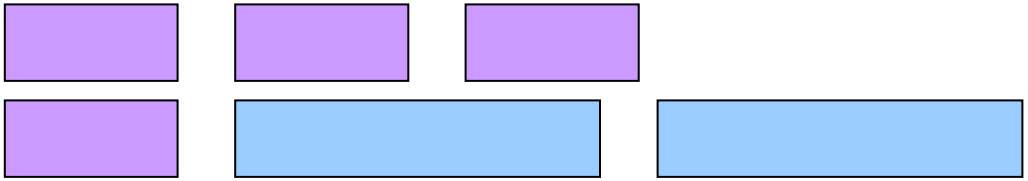
Sample:

Our One Pass Solution

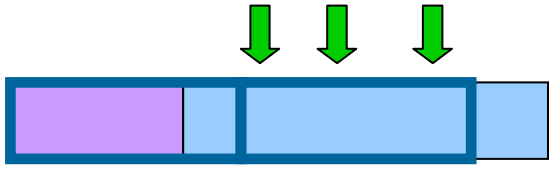


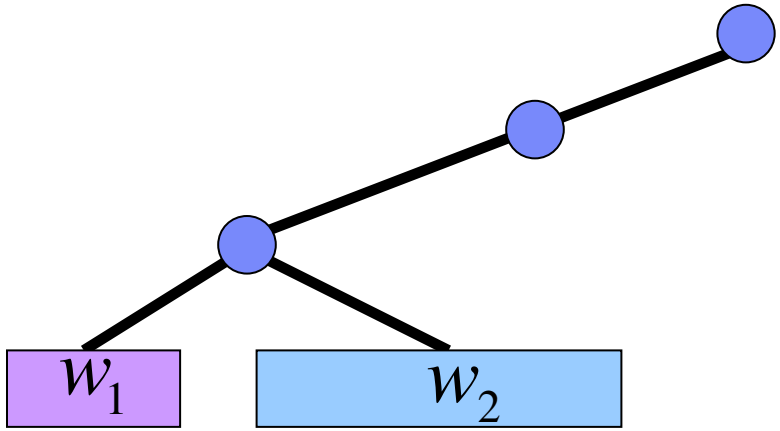


Sample:

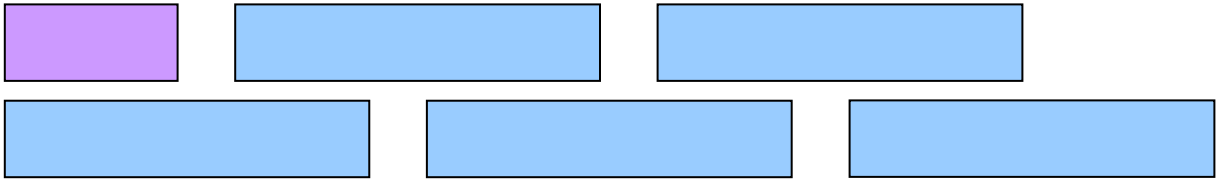


Our One Pass Solution

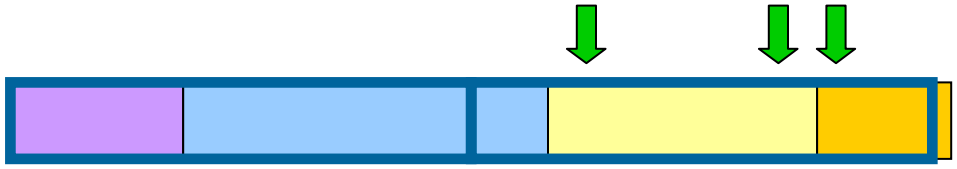


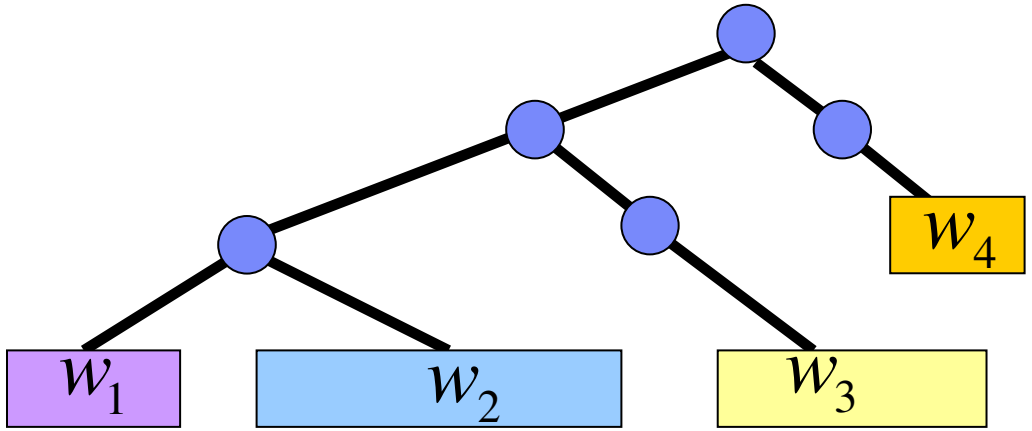


Sample:

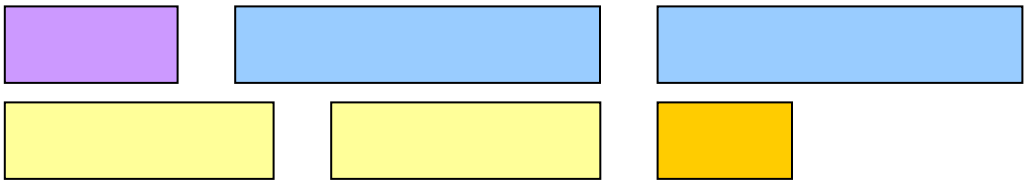


Our One Pass Solution

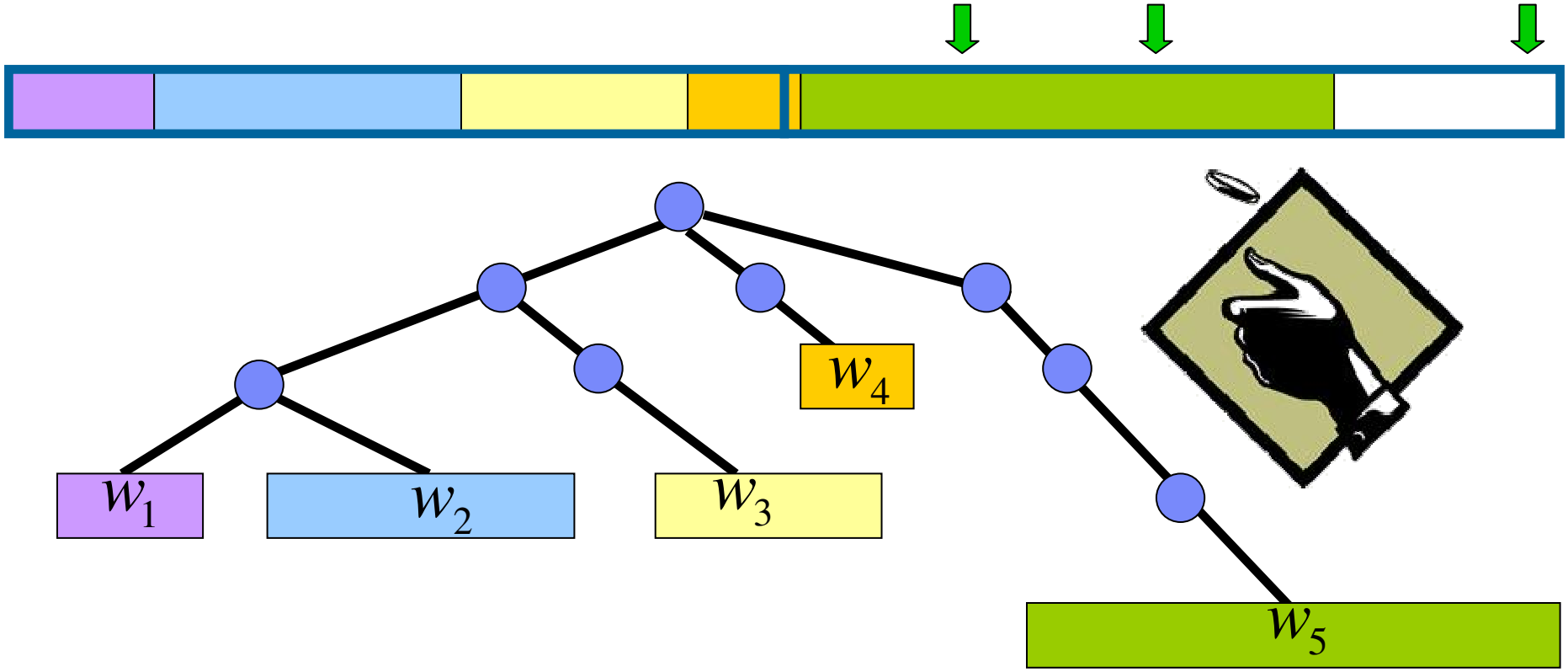




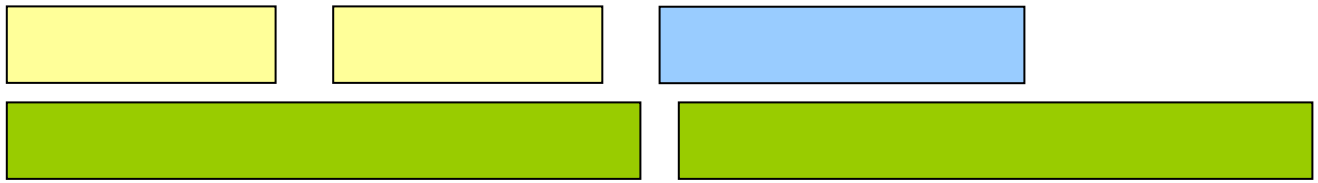
Sample:



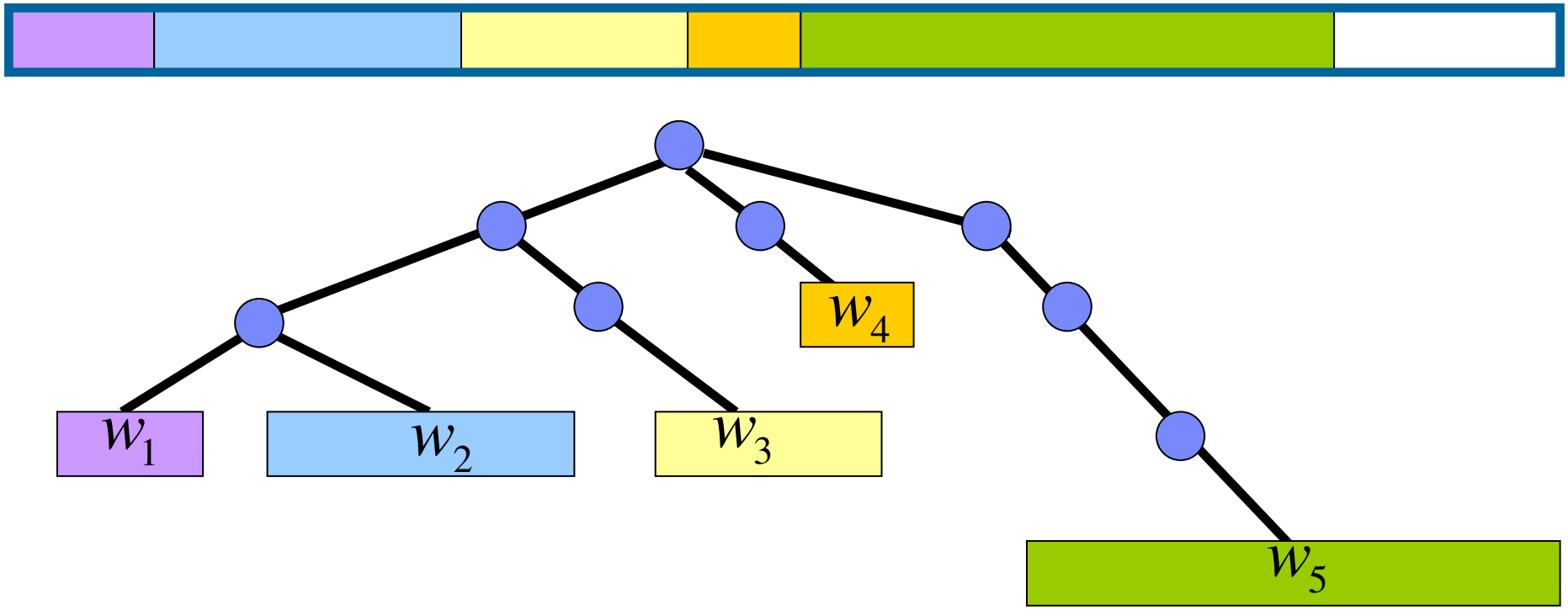
Our One Pass Solution



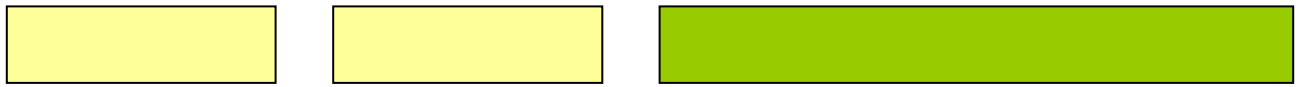
Sample:



Our One Pass Solution



Sample:



Additional Challenges - Distribution

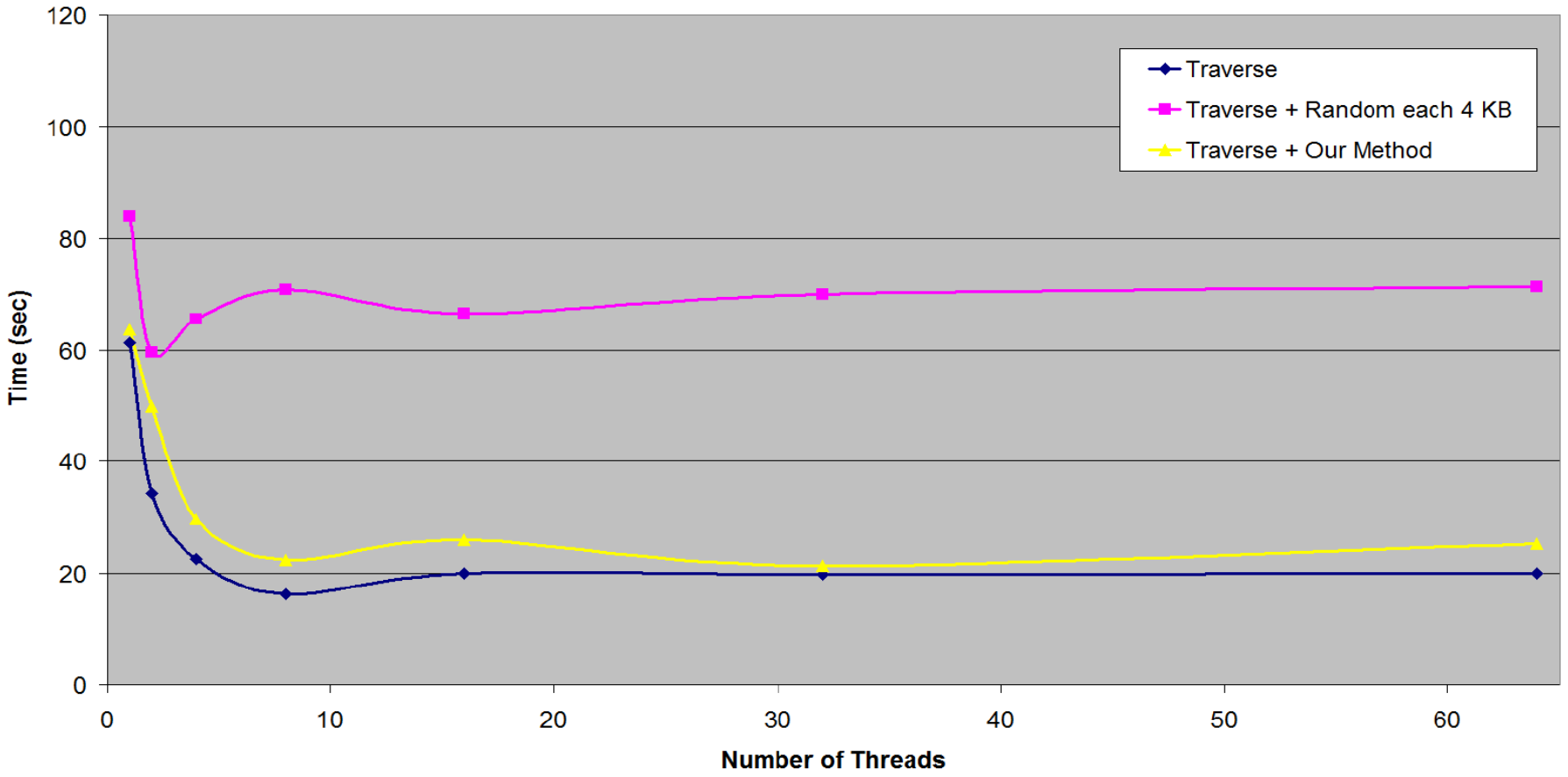
Traversal would greatly benefit from the multithreading and distribution.

So the sampling mechanism should also support distribution.

We have such solutions:

- Multithreaded (in a single node)
- Distributed (across nodes)

Evaluation of Our Method: Time Complexity



Use Cases

- Two scenarios where sampling is beneficial in file systems:
  - The query is expensive
  - The query is unknown in advance

- Real life examples:
  - The query is expensive
    - Testing and auditing mechanism e.g. Speech to text verification
    - Compression estimation
  - The query is unknown in advance
    - Offline analysis of file system distribution

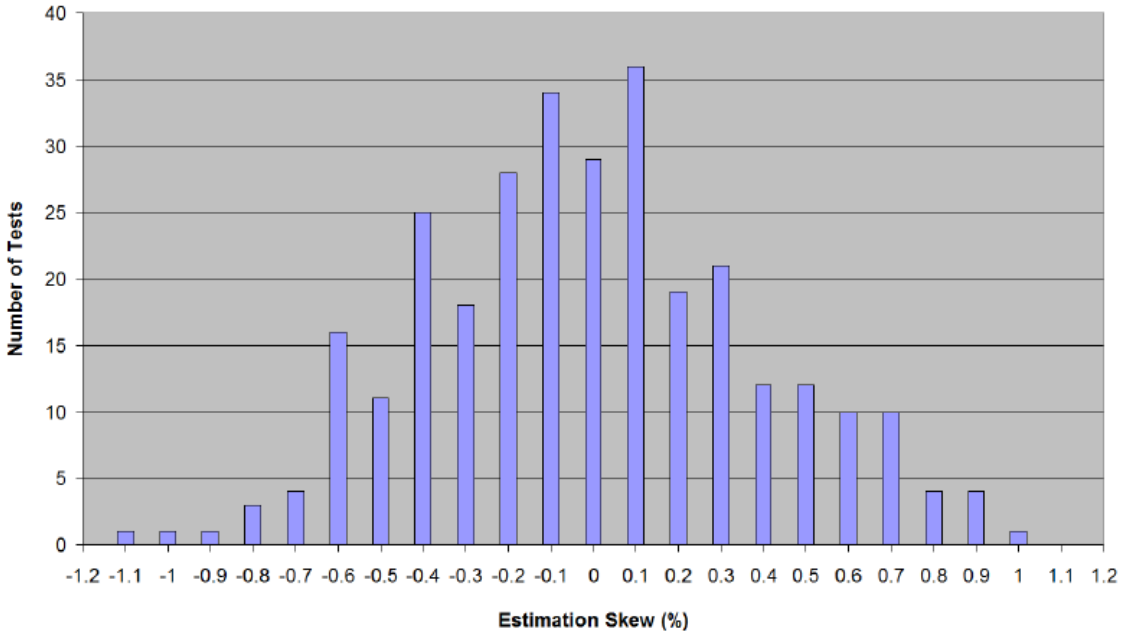
Evaluated Data Sets

Name	#file	Size
Impression FS	4.9 M	1.86 TB
Project Repository	17.6 M	7.8 TB
Compression Collection	21.5 K	430 GB
Bloated Repository	220M	93 TB

Use Case 1 – Expensive Query

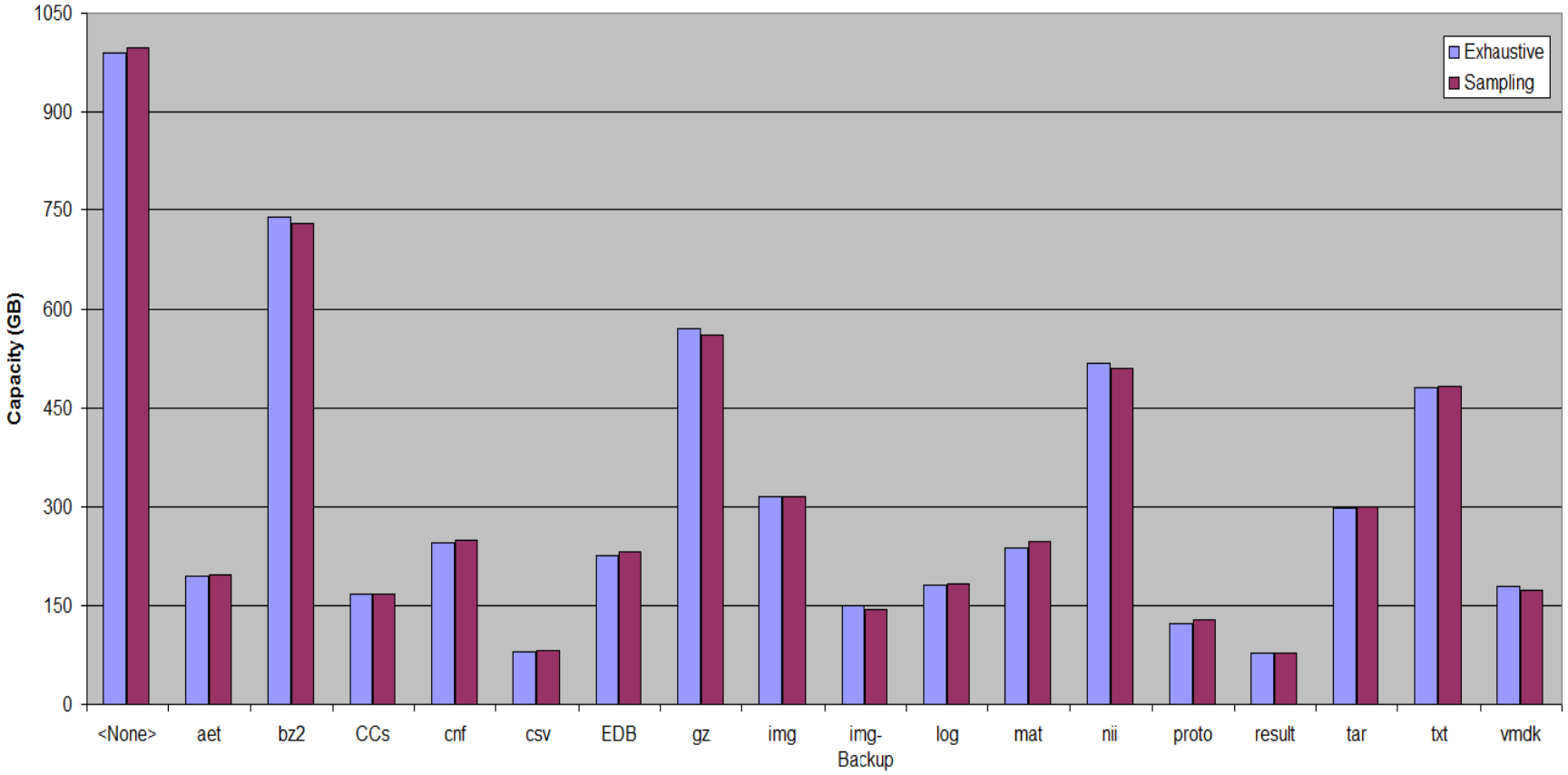
Compression Estimation

Data Set	Total Capacity (GB)	Capacity Read for Estimation	Exhaustive Time	Sampling + Estimation Time
Impressions	1905 GB	320 MB	**> 13 hours**	**26 min**
Compression	428 GB	320 MB	**> 100 min**	**42 sec**



Use Case 2 – Offline File System Analysis

Capacity Distribution per File Extension Type



To Summarize:

Why isn't sampling used for file systems?

1. No apparent benefit for sampling in file systems
2. Technical problems of sampling:

Remainder of this talk:

1. Solving the technical problems of sampling
2. Demonstrating beneficial real life use cases

Thanks

Questions