CLIP: A Compact, Load-balancing Index Placement Function

Michael McThrow **Carlos Maltzahn** **Neoklis Polyzotis** **Scott Brandt**

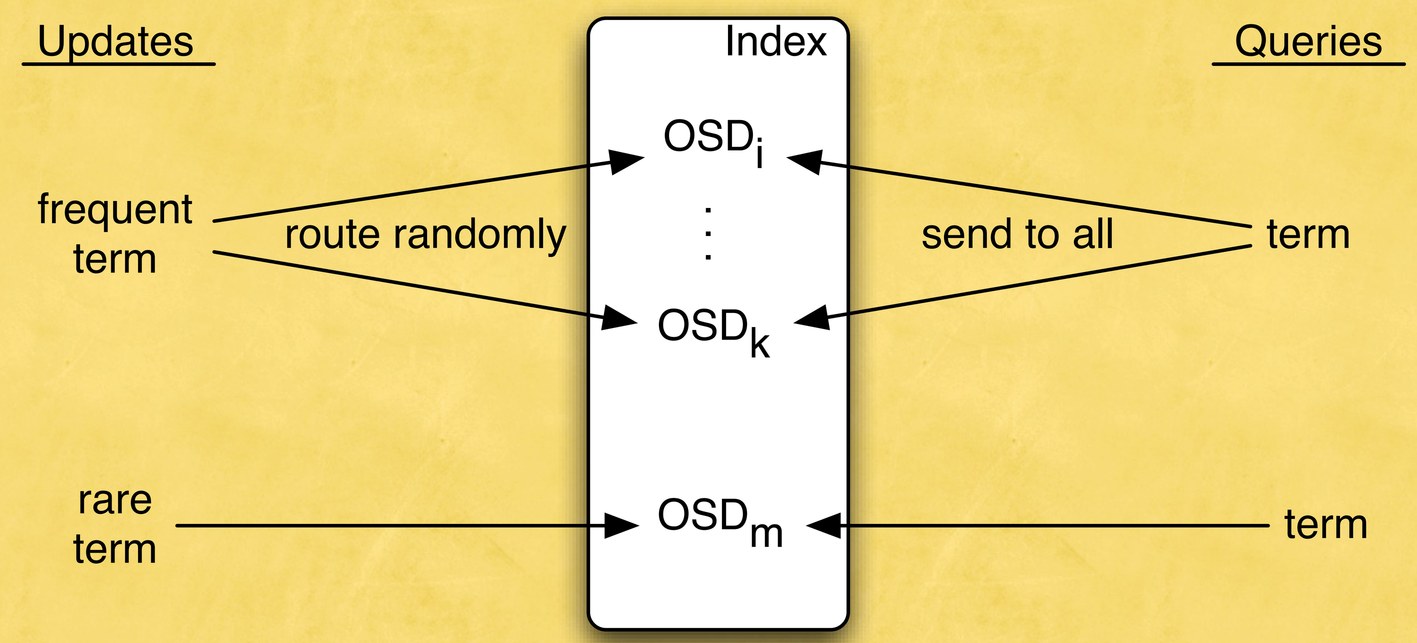
Storage Systems Research Center at the University of California, Santa Cruz
http://ssrc.cse.ucsc.edu

Introduction:

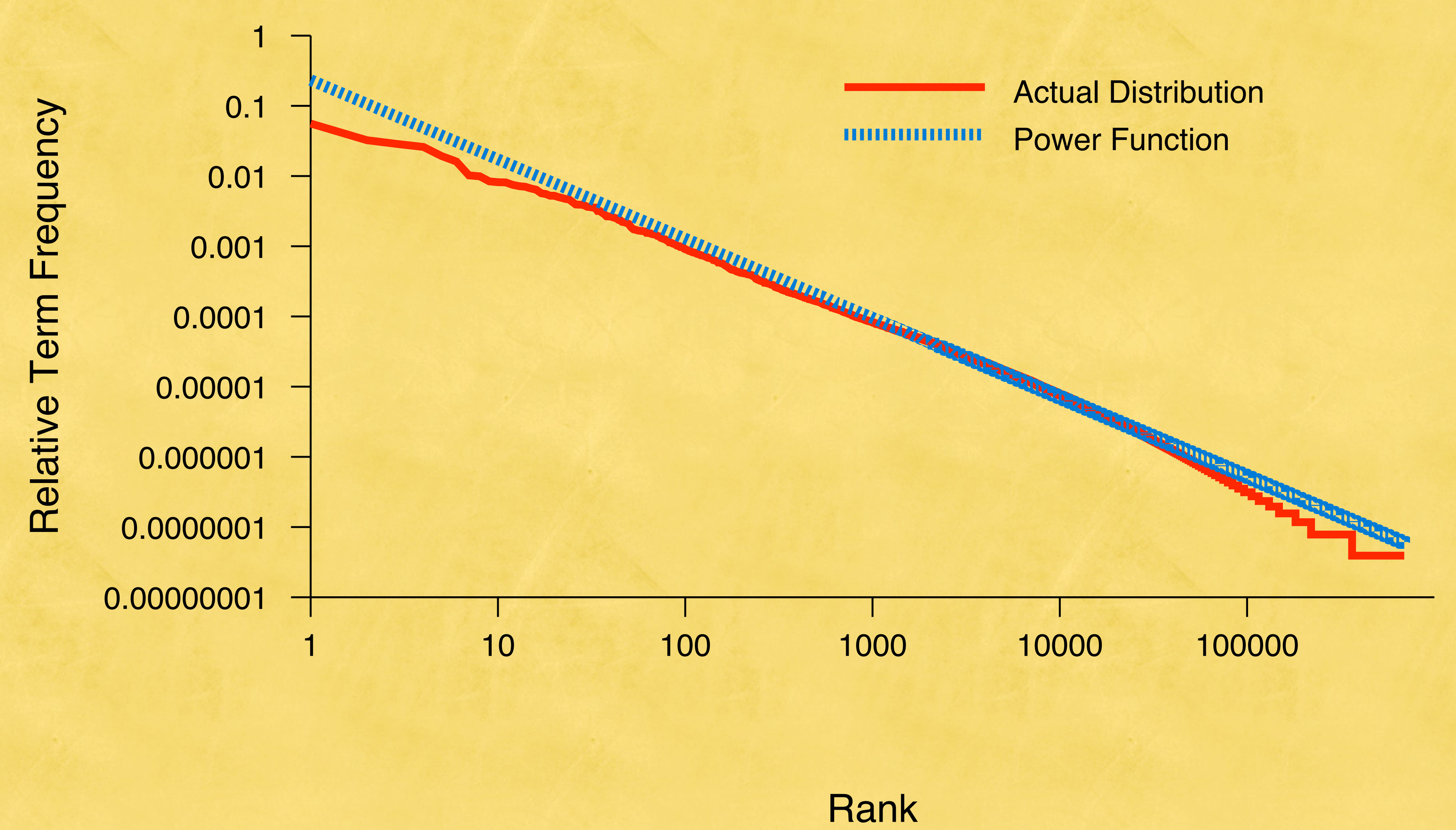
- The petabyte-scale distributed storage system Ceph pseudo-randomly places data on up to 10,000s of object storage devices (OSDs) using a compact function (CRUSH)
- Compactness of CRUSH essential for scalability
- Search in Ceph requires the maintenance of large indices with a very skewed update load profile (Zipf-like distribution).
- How to extend CRUSH so it can handle skewed update profiles while keeping it compact?

Approach:

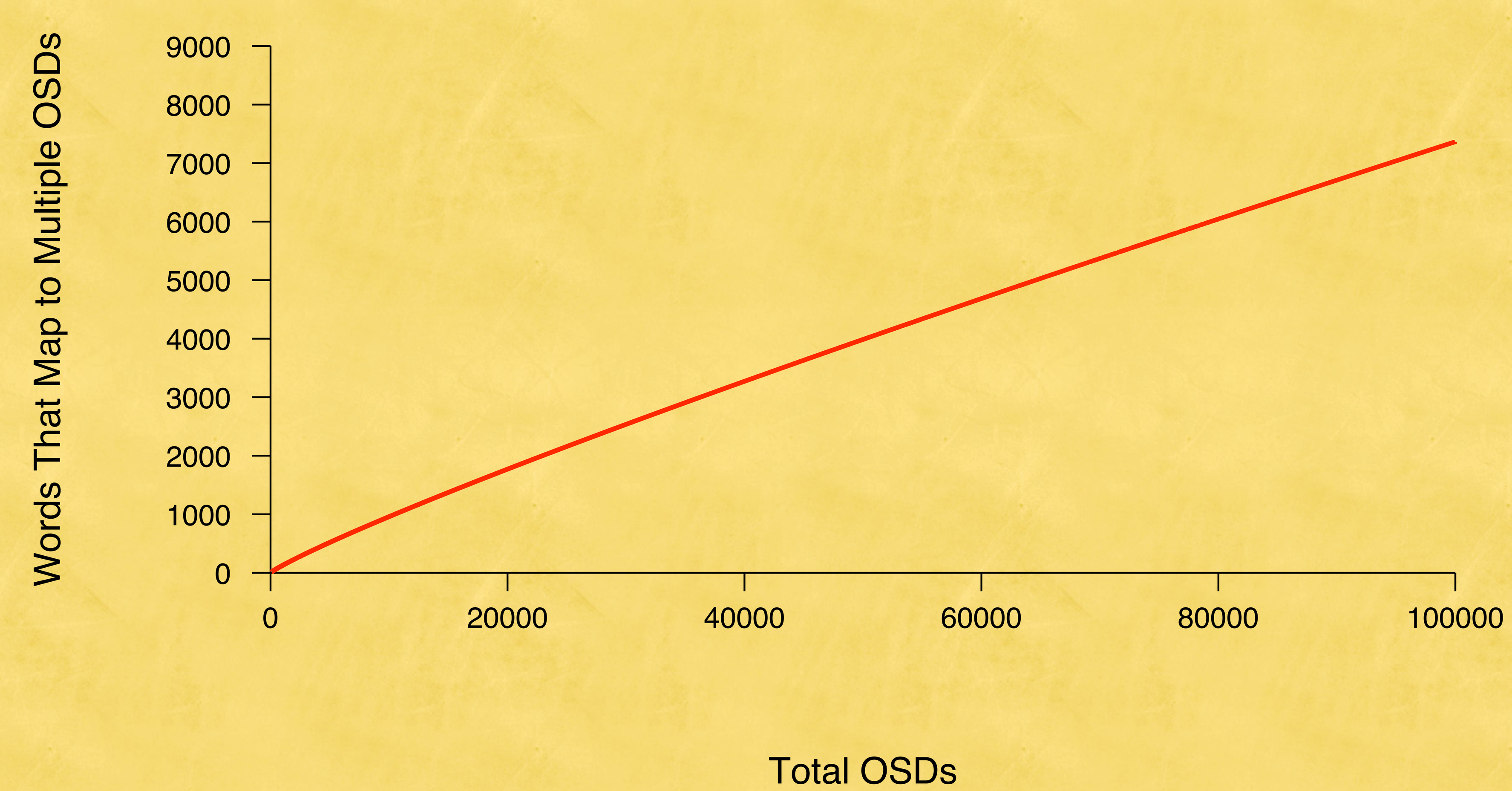
- Split frequently updated parts of index across multiple OSDs, randomly select one of these for each update, and read all of these for queries.
- CRUSH maps a value to a deterministic sequence of OSDs:
  (term, k) → (OSD_1, ..., OSD_k)
- Determine k based on relative term (update) frequency and total number of OSDs
- Need compact representation of term frequency distributions of 100,000s of terms.
- **Idea: only keep track of terms with frequencies that lead to k > 1.**
- How many terms?



Routing of updates and queries to OSDs



Term frequency distribution and fitted power function



Estimating number of terms that map to multiple OSDs

Results:

- Fitted power function to relative term frequency distribution based on Gutenberg Project 2006 DVD:
  relative_freq ≈ $0.2327 \text{ rank}^{-1.1292}$
- The estimated rank_1 at which k ≤ 1 for a given total number of OSDs:
  $1/\text{totalOSDs} \approx 0.2327 \text{ rank}_1^{-1.1292}$
  $\text{rank}_1 \approx (1/(0.2327 \text{ totalOSDs}))^{-1/1.1292}$
- Linear approximation:
  rank_1 ≈ 7-8% of total number of OSDs

Conclusions

- **Even in very large systems only a relatively small number of terms require more than one OSD**
- Storing those terms and their relative frequencies still leads to a compact placement function
- Initial approach: use Bloom filters to categorize terms by their frequency.
  - Unnecessary and too expensive (time and space) due to small number of terms
  - False positives can lead to significant communication overhead
- Future work:
  - Verify that CLIP balances load.
  - Integrate CLIP into Ceph

This work was completed as part of UCSC's SURF-IT summer undergraduate research program, an NSF CISE REU Site. This material is based upon work supported by the National Science Foundation under Grant No. CCF-0552688 and CCF-0621534