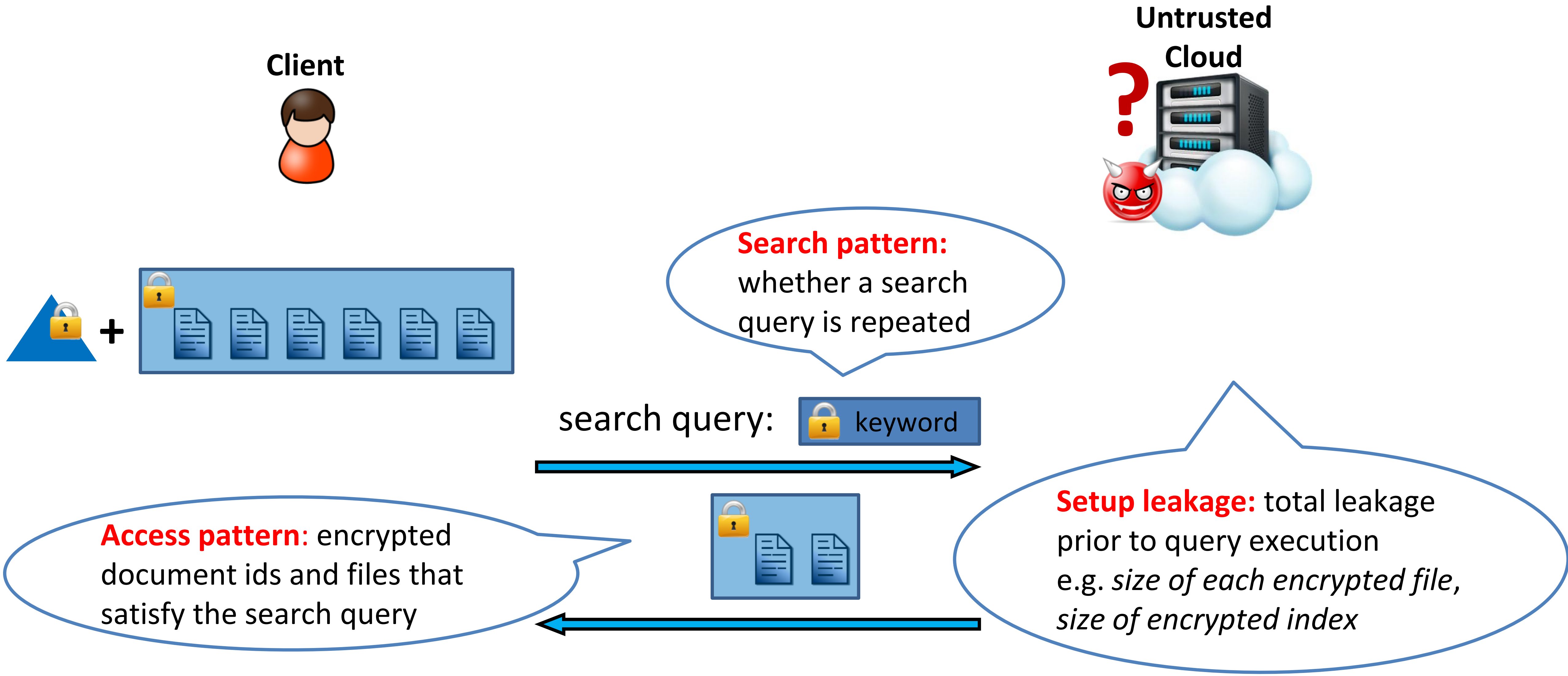
Searchable Encryption with Optimal Locality:
Achieving Sublogarithmic Read Efficiency

Ioannis Demertzis
University of Maryland
yannis@umd.edu

Dimitris Papadopoulos
Hong Kong UST
dipapado@cse.ust.hk

Charalampos Papamanthou
University of Maryland
cpap@umd.edu

What is Searchable Encryption (SE)?



<u>Security (informal):</u> The adversary does not learn anything beyond the above leakages!

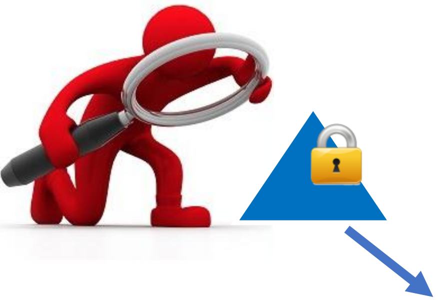
Searchable Encryption – Locality and Read Efficiency

Scalable SE requires low locality and read efficiency

Locality is an important efficiency dimension ([CJS+14],[DP17],…

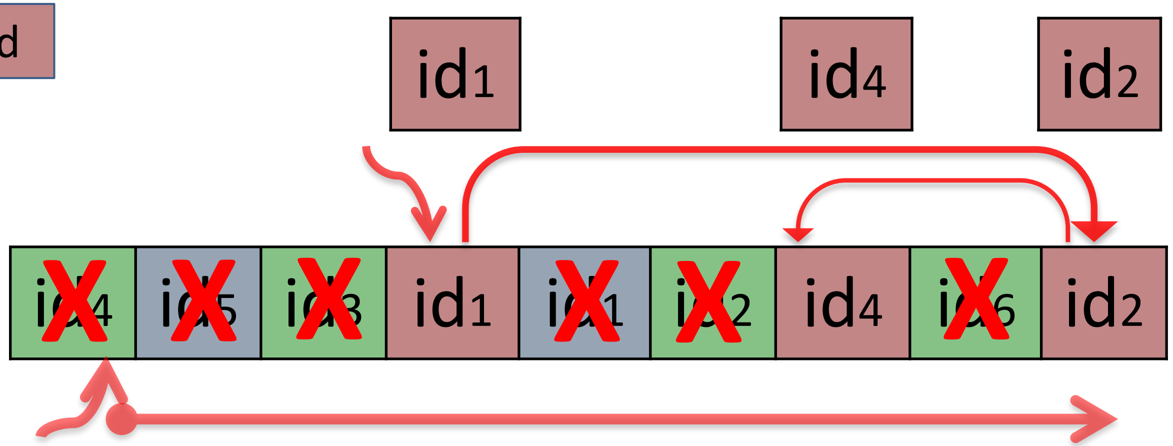
Locality: #non-continuous reads for each query

Read Efficiency: #memory locations per result item



search query: keyword

locality = 3 & read efficiency = 1



X : false positives

locality = 1 & read efficiency = O(N)

**under some assumptions for the SE scheme*

Previous Works & Our Result

"Cash and Tessaro Eurocrypt 2014"

Locality (L): **O(1)** and Read Efficiency (R): **O(1)** requires* Space (S): **ω(N)**

<u>General Schemes</u>

[ANS+16] – NlogN scheme

L: O(1), **R:** O(1), **S:** O(NlogN)

[DP17] – ReadOpt

L: O($N^{1/s}$), **R:** O(1), **S:** O(sN)

[ANS+16] – OneChoiceAlloc

L: O(1), **R:** Õ(logN), **S:** O(N)

<u>Schemes with limitation on the maximum keyword-list size</u>

[ANS+16] – TwoChoiceAlloc *

L: O(1), **R:** Õ(loglogN), **S:** O(N)

* keyword lists in the dataset have size less than $\mathbf{N^{1-1/\log\log N}}$.

[ASS18]**

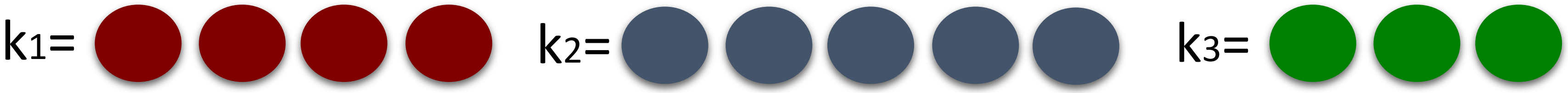
L: O(1), **R:** O(ω(1)*ε^{-1}(n) + logloglogN) for n = $N^{1-\varepsilon(N)}$, **S:** O(N)

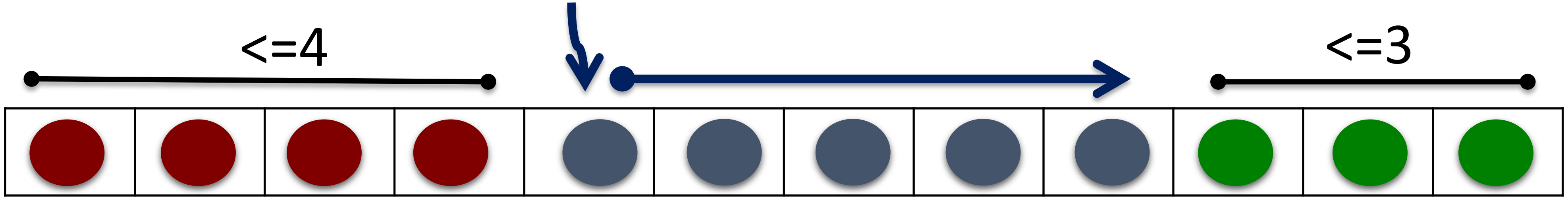
** keyword lists in the dataset have size less than $\mathbf{N/\log^3 N}$

Our Approach

L: O(1), **R:** O($\log^{\gamma}$N), **S:** O(N), **for** γ>2/3

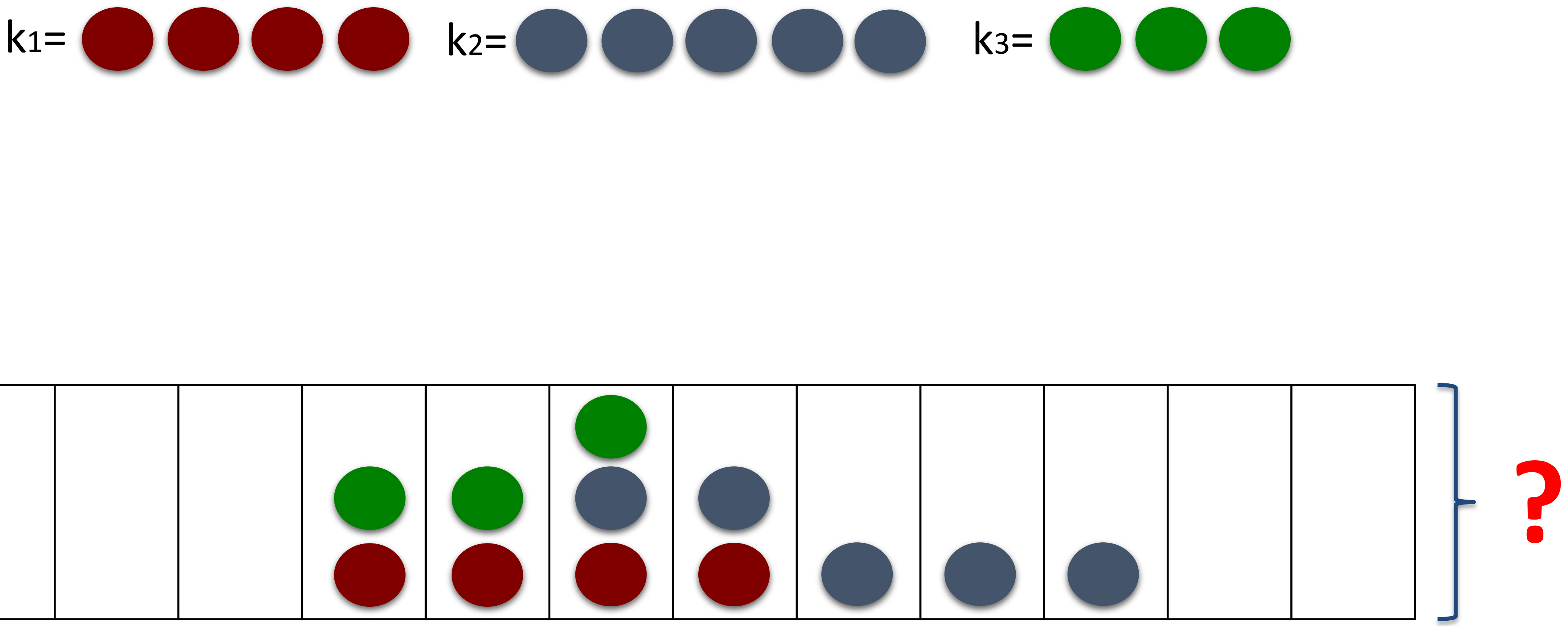
Searchable Encryption – Naïve Approach 1





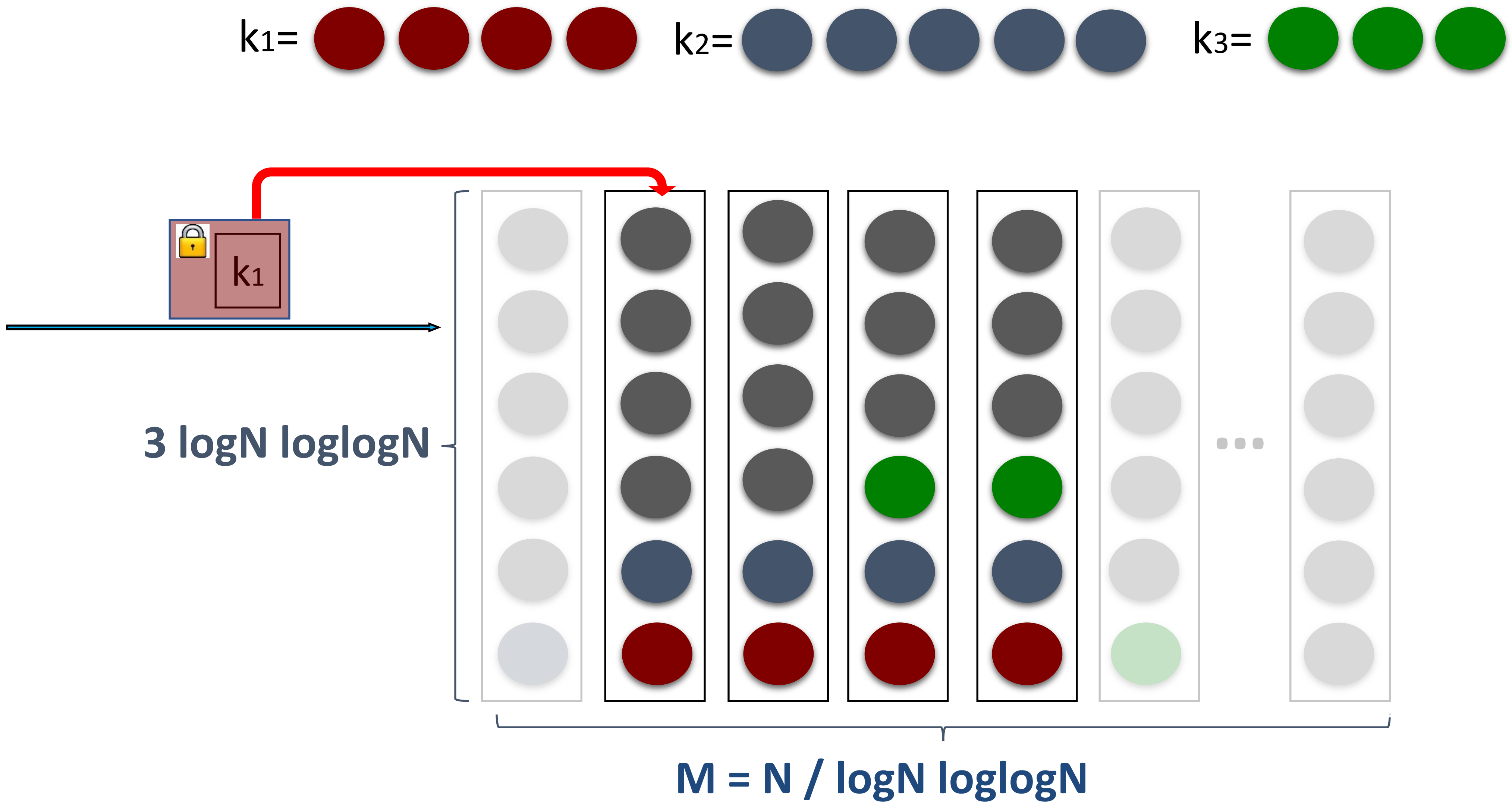
locality = 1 & read efficiency = 1 & optimal space

Searchable Encryption – Naïve Approach 2



[ANS+16]– OneChoiceAllocation

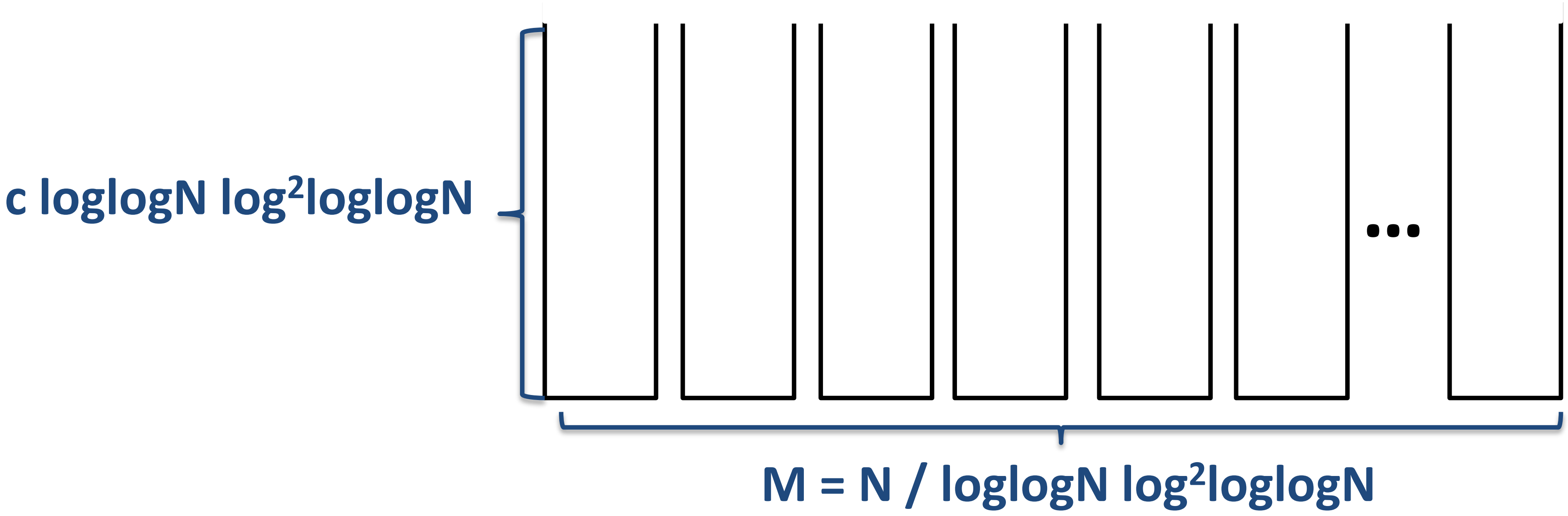
O(N) space, O(1) locality and Õ(logN) read efficiency



[ANS+16]– TwoChoiceAllocation

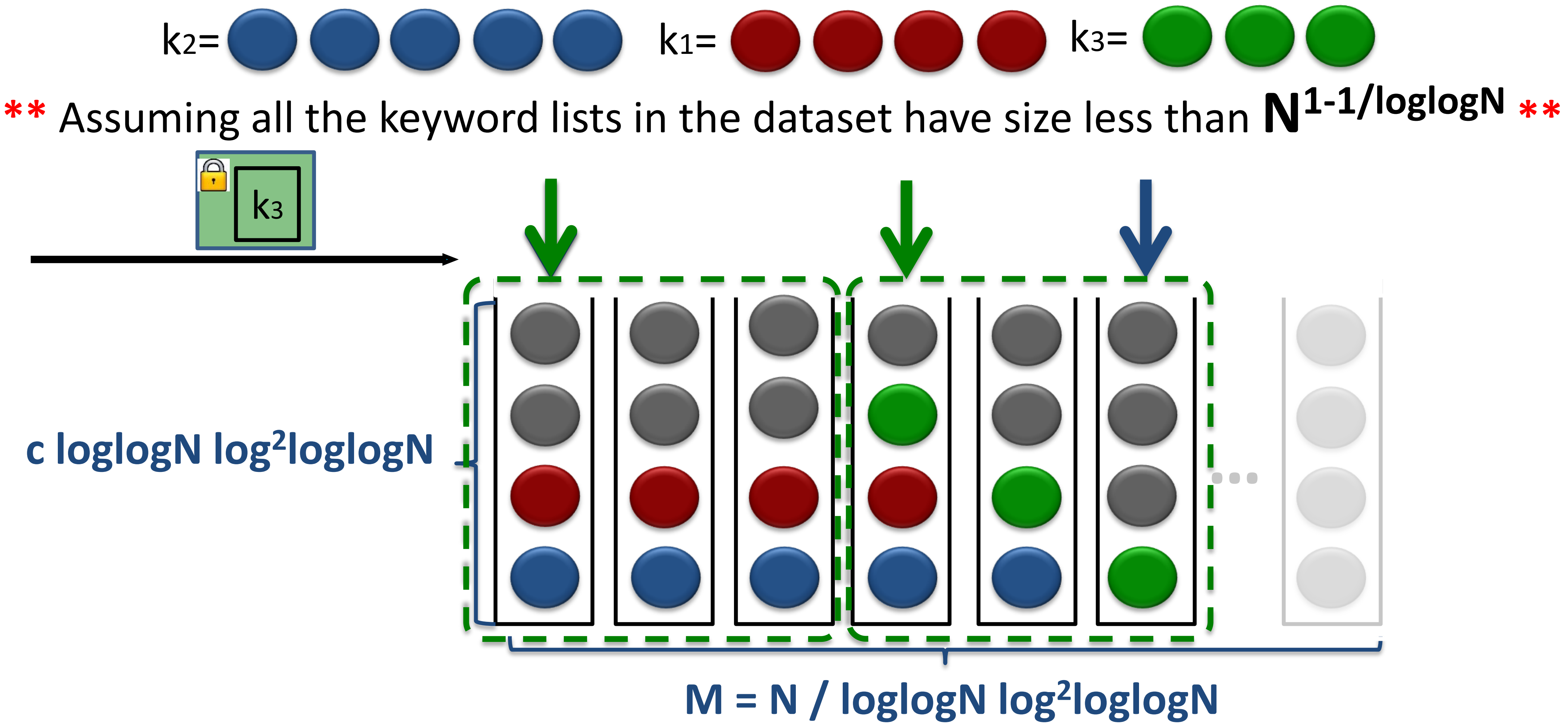
O(N) space, O(1) locality and Õ(loglogN) read efficiency

** Assuming all the keyword lists in the dataset have size less than $\mathbf{N^{1-1/loglogN}}$ **



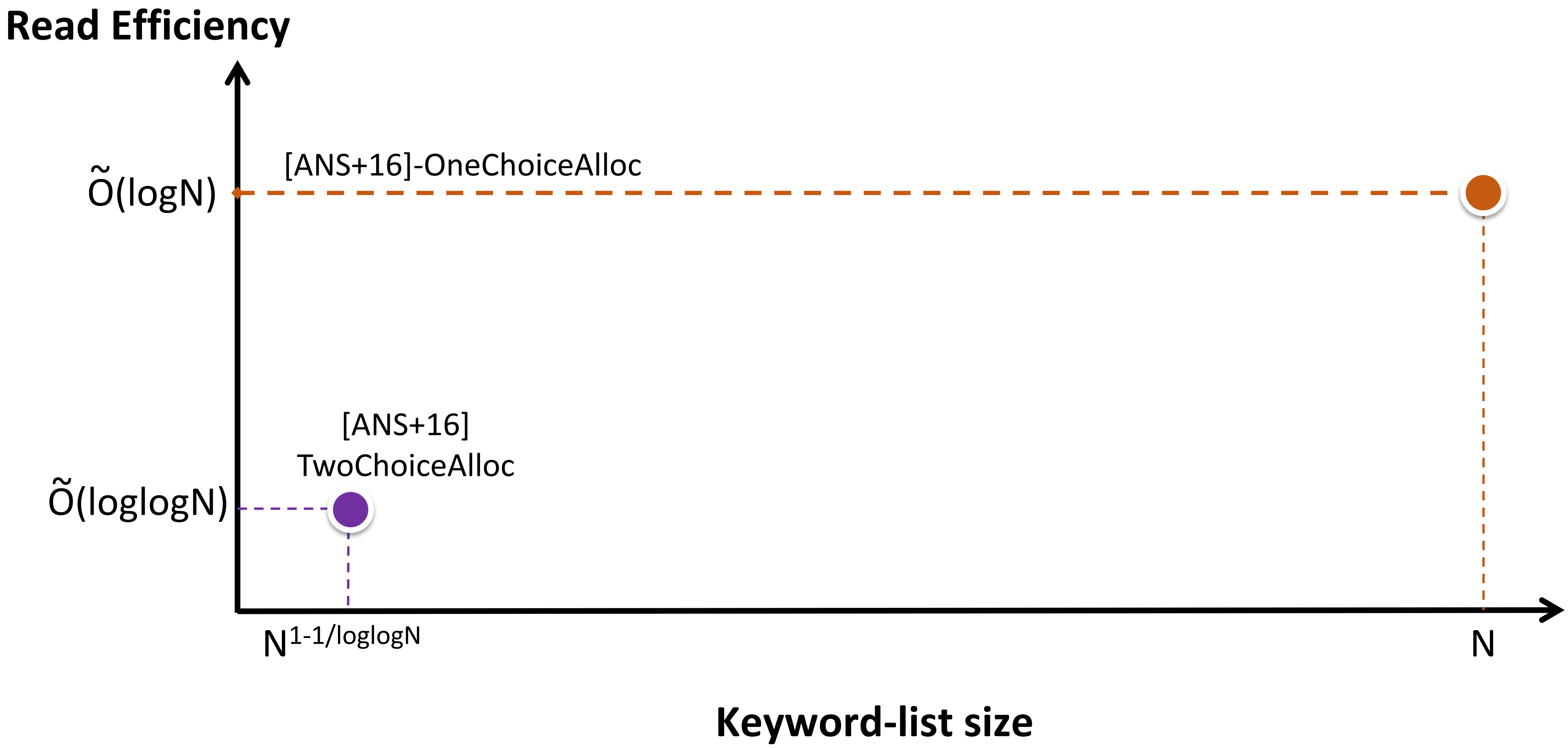
[ANS+16]– TwoChoiceAllocation

O(N) space, O(1) locality and Õ(loglogN) read efficiency



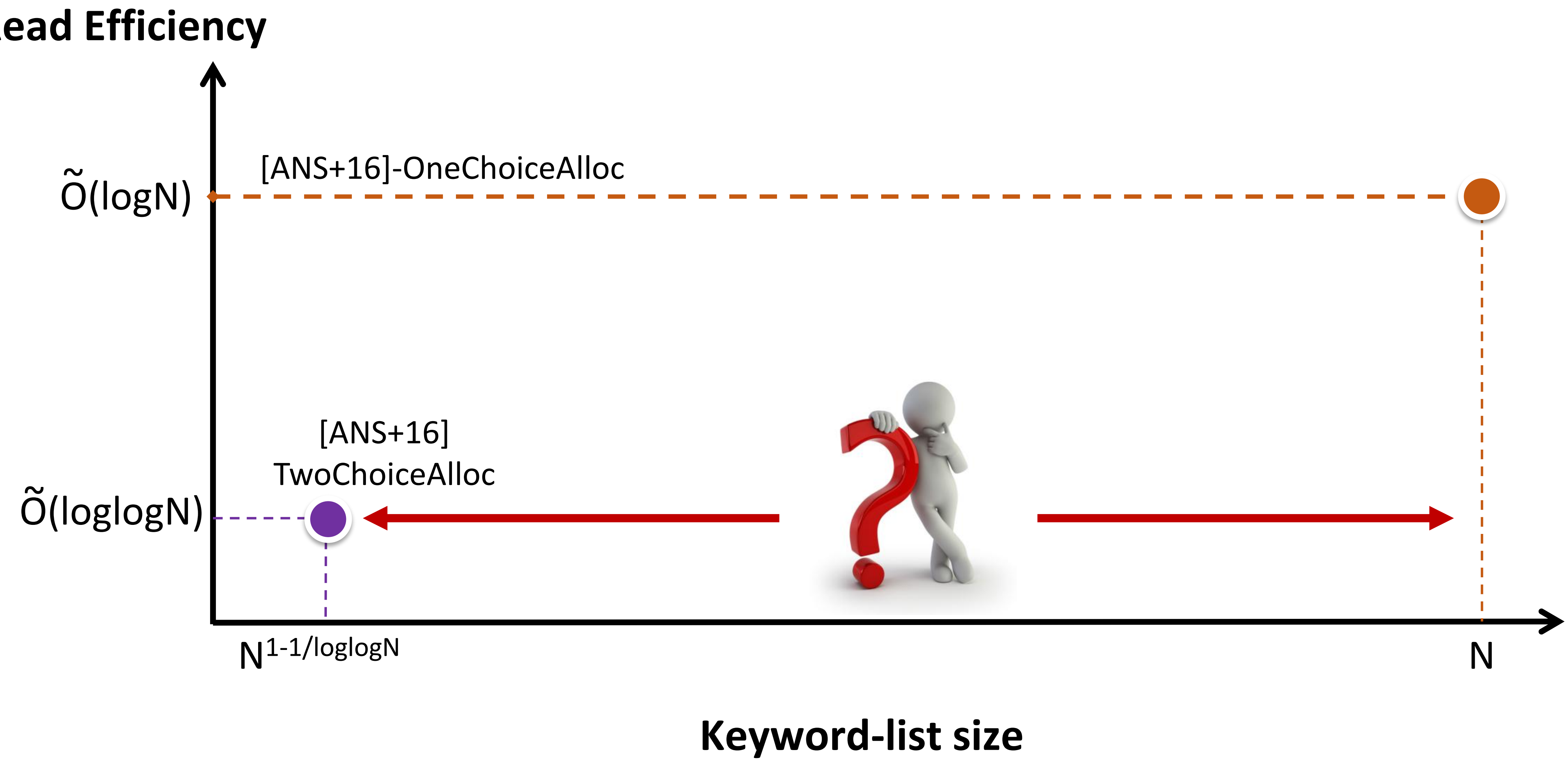
Our Approach

O(N) space, O(1) locality and O(log$^{\gamma}$N), for γ>2/3



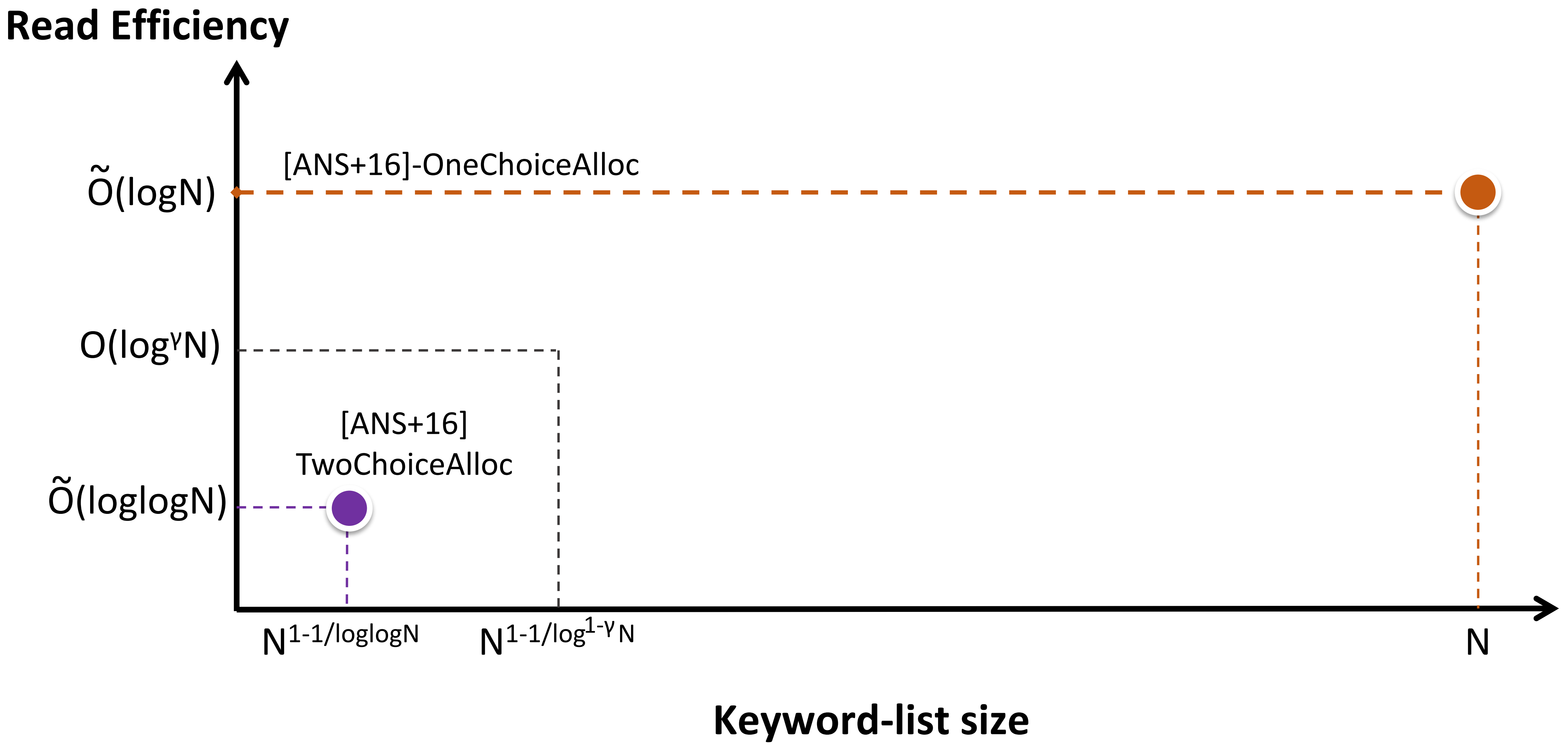
Our Approach

O(N) space, O(1) locality and $O(\log^{\gamma} N)$, for $\gamma > 2/3$



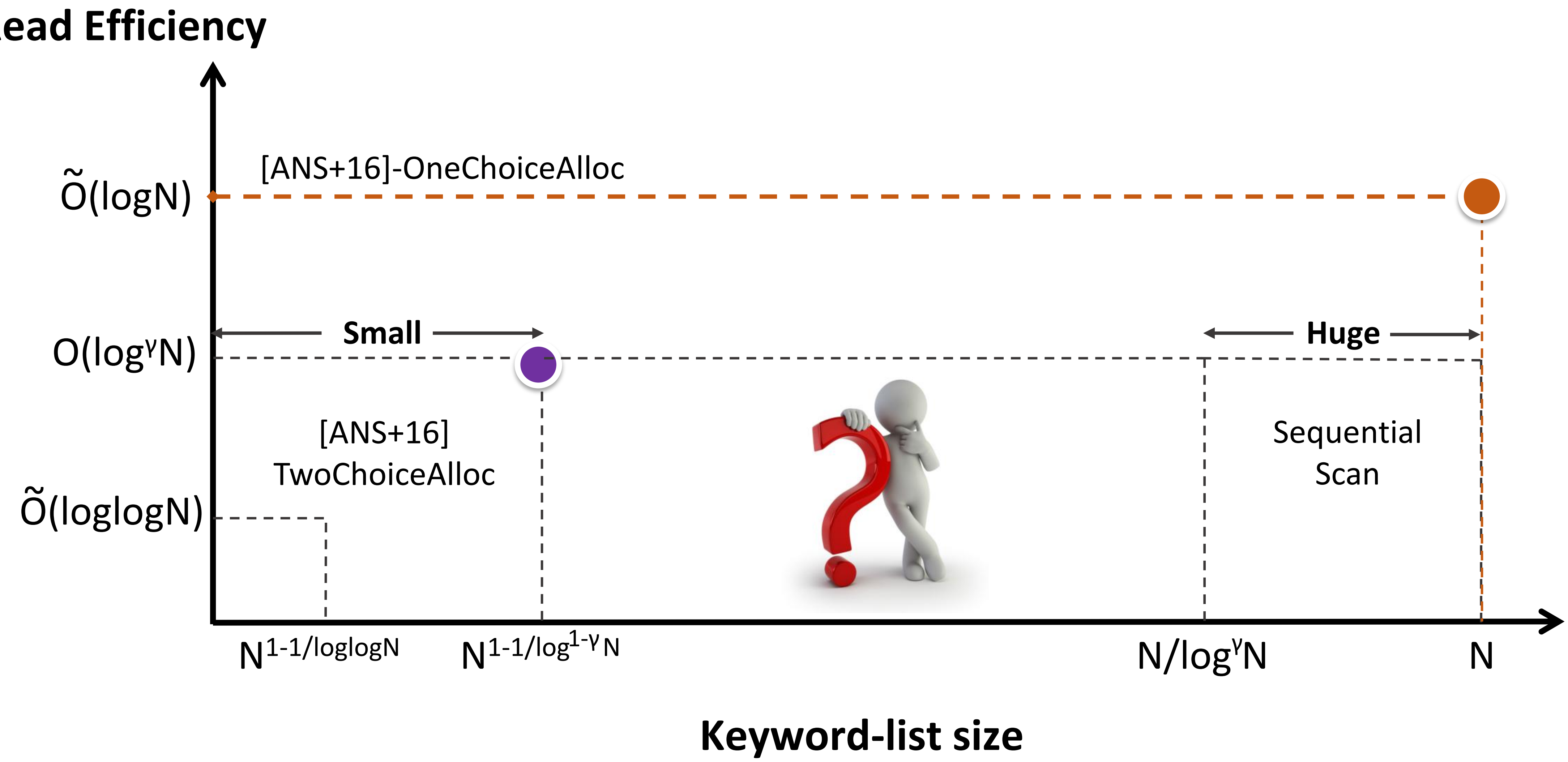
Our Approach

O(N) space, O(1) locality and O(log$^{\gamma}$N), for γ>2/3



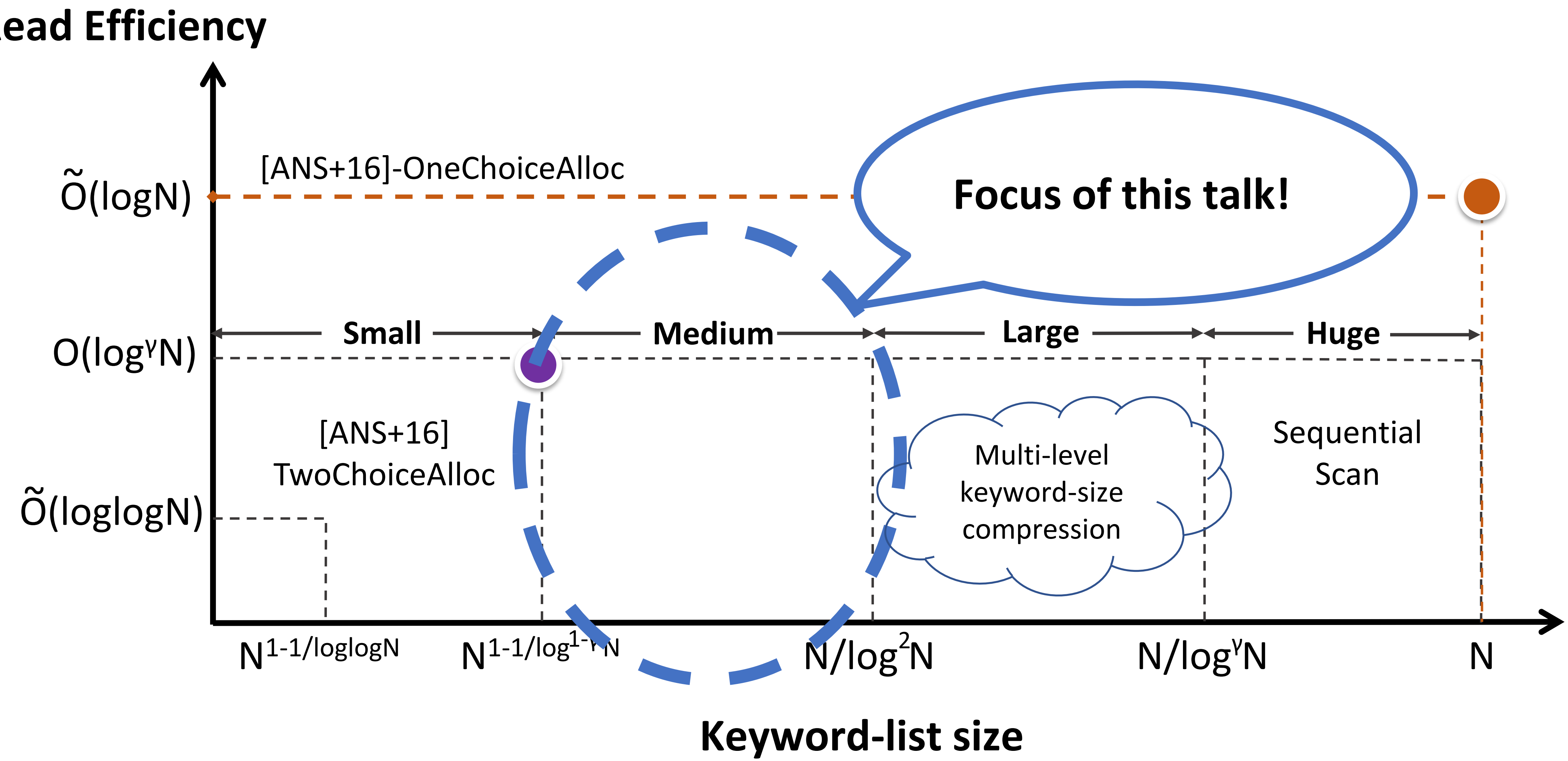
Our Approach

O(N) space, O(1) locality and O($\log^{\gamma}$N), for γ>2/3



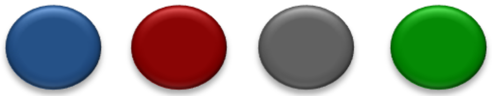
Our Approach

O(N) space, O(1) locality and O(log^γN), for γ>2/3

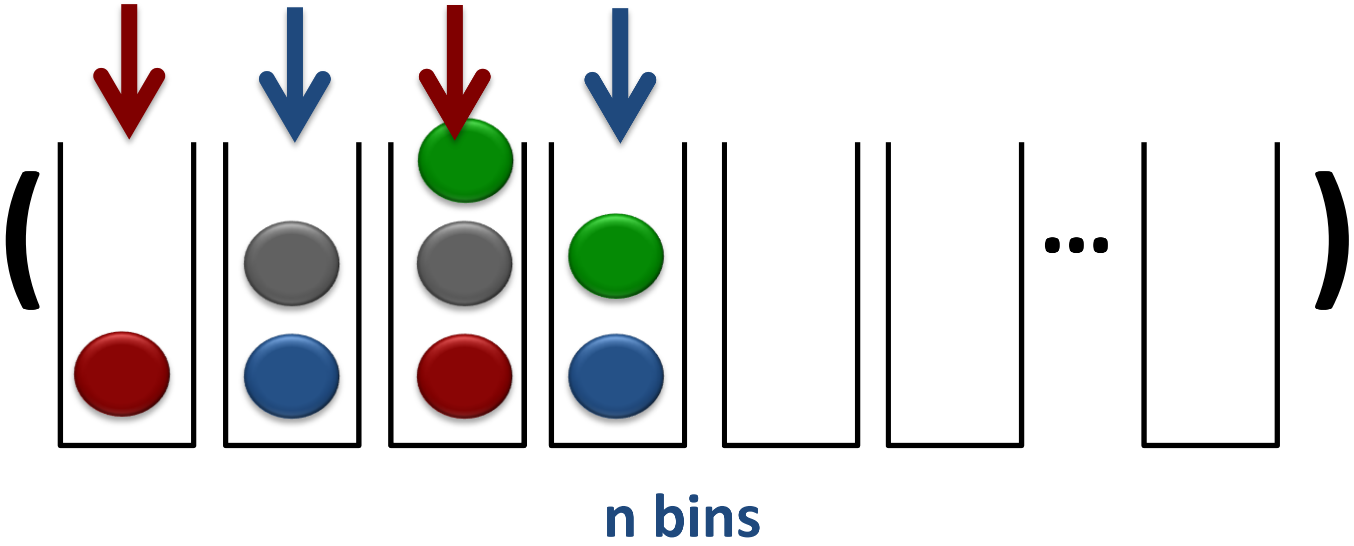


Starting Point: Offline Two Choice Allocation (OTA) – [SEK03]

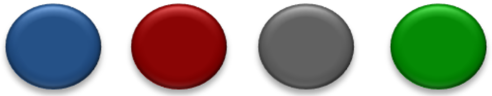
OfflineTwoChoiceAlloc for **m** balls and **n** bins:



MaxFlow



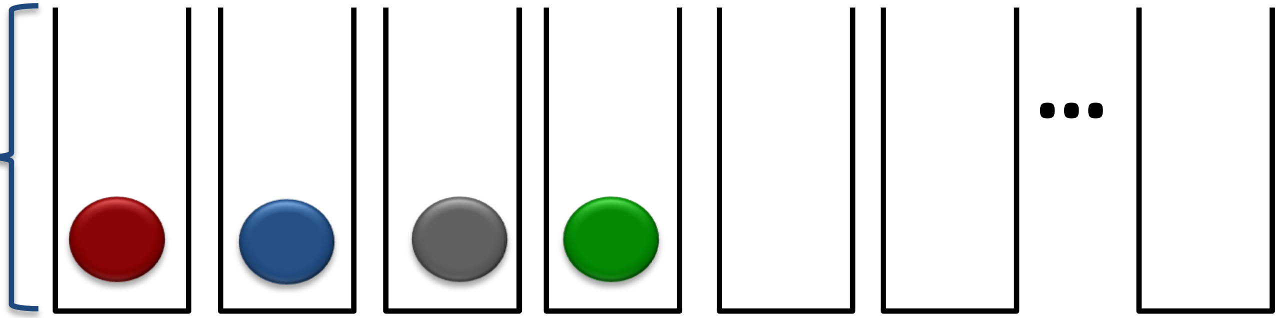
Starting Point: Offline Two Choice Allocation (OTA) – [SEK03]

OfflineTwoChoiceAlloc for **m** balls and **n** bins: 

Key IDEA: One OTA per size and then Merge!!

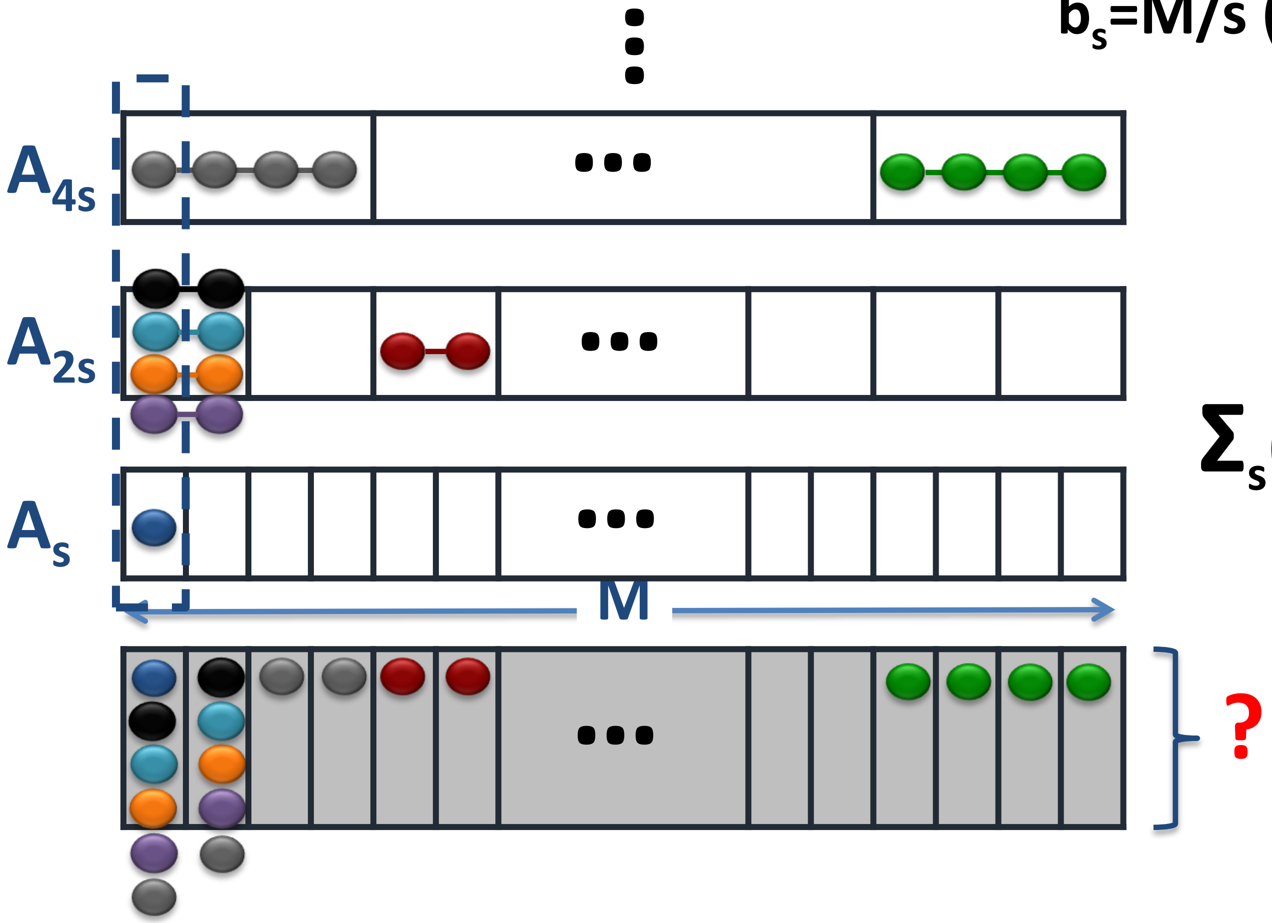
with probability at least 1 – O(1/n)

Max load <= $\lceil m/n \rceil + 1$



n bins

Our Approach: OTA per size + Merge



k_s: #keyword lists with size s
b_s=M/s (#superbuckets)

Overflow Probability = $O(1/b_s)$

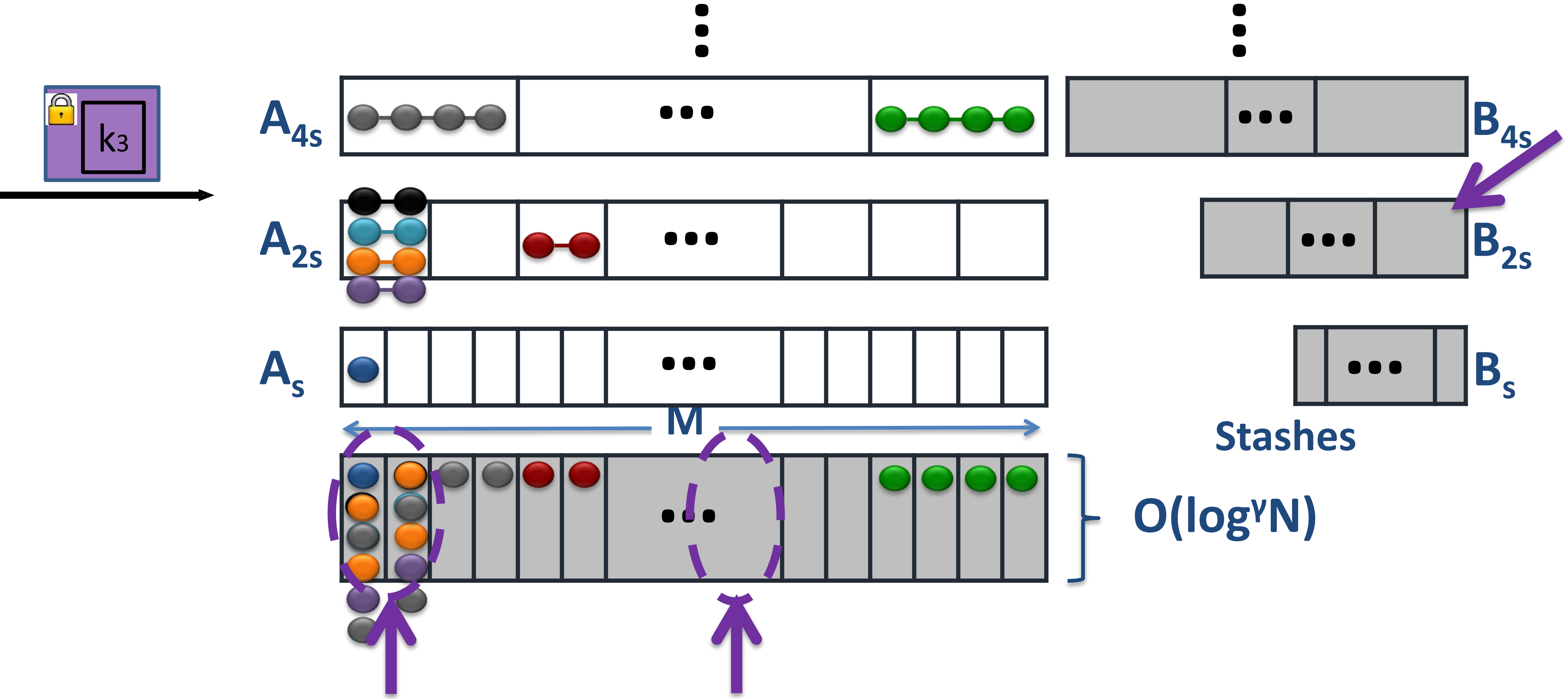
See Lemma 4 in our paper

$$\Sigma_s\left(\lceil k_s/b_s \rceil + 1\right) = O(N/M + \log^{\gamma} N)$$

$$\overset{M = N/\log^{\gamma} N}{=} O(\log^{\gamma} N)$$

Our Approach: New analysis for OTA

****Novel analysis for OTA**** The probability that more than **$O(\log^2 N)$** lists of size **s** overflow **is negligible!** – see Lemma 5 in our paper

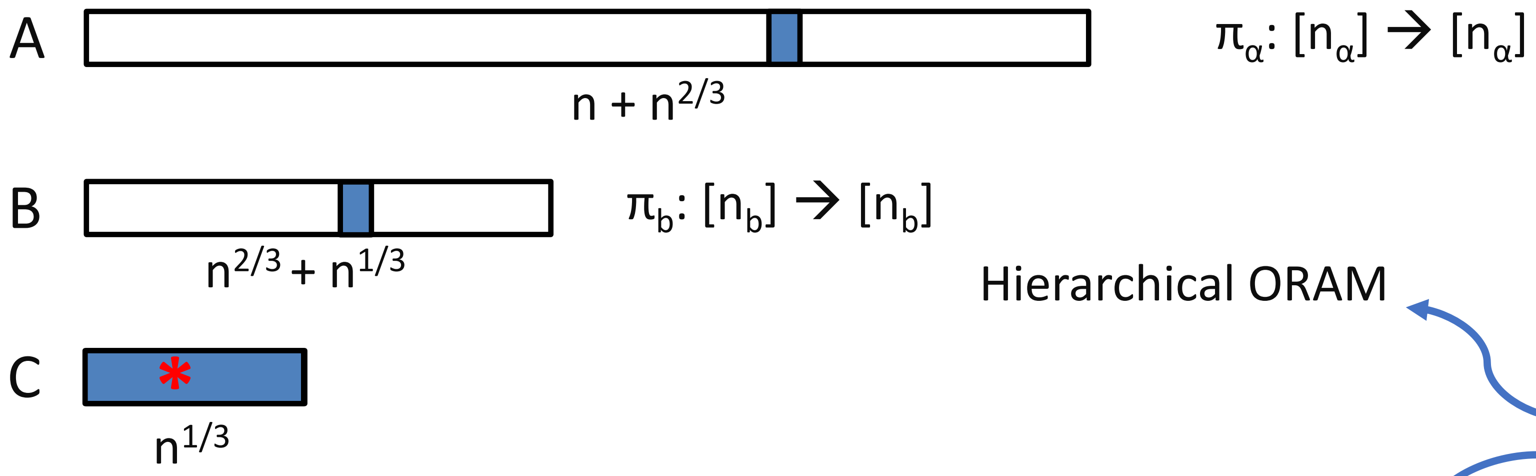


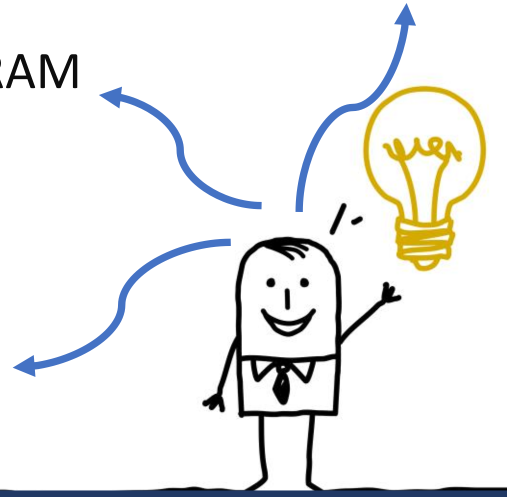
Our Approach: New locality-aware ORAM

$O(n^{1/3}\log^2 n)$ Bandwidth and $O(1)$ Locality

We need an ORAM with the following properties:

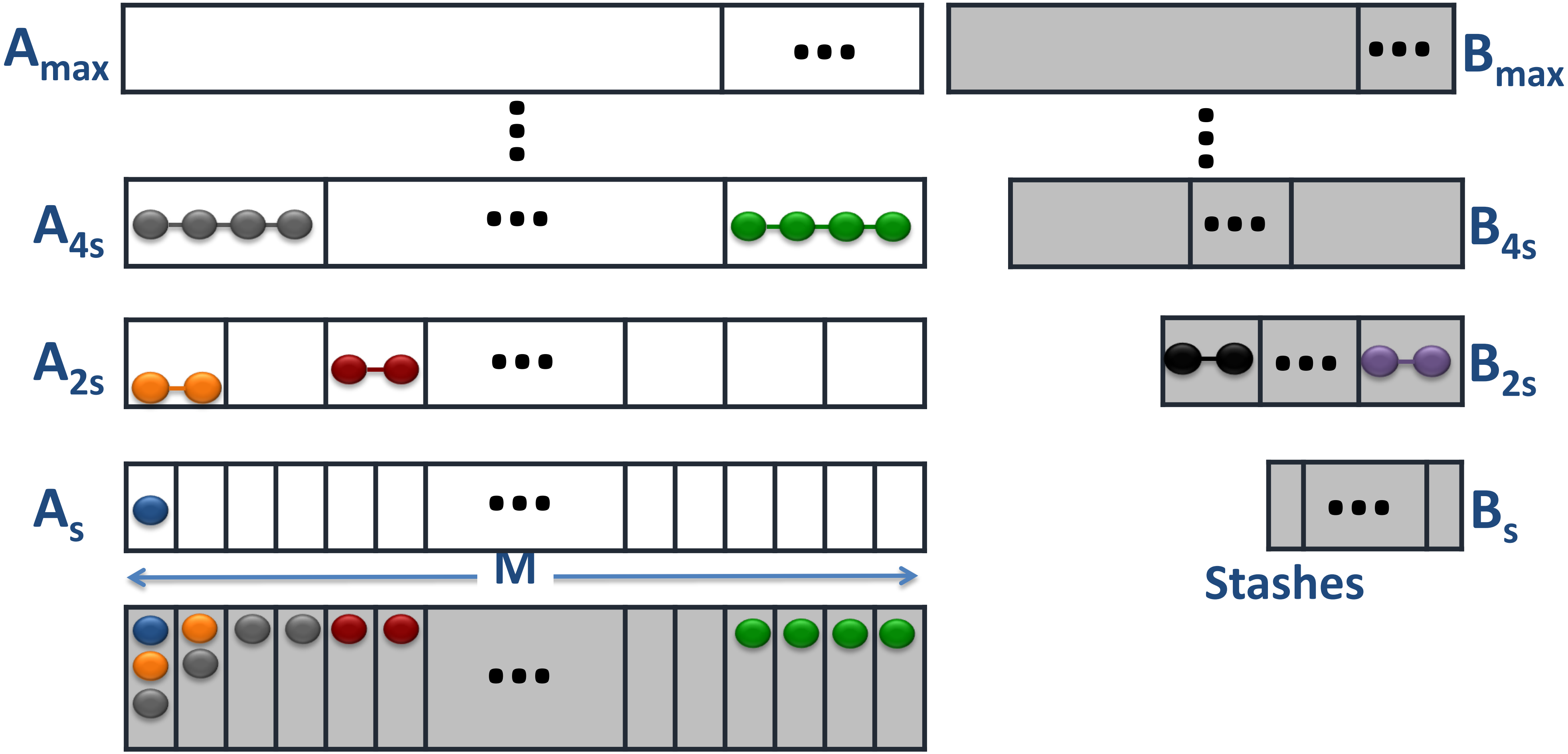
1. **O(1) locality**, existing ORAMs with polylogn bandwidth have logn locality
2. **Zero failure probability**, since it will be applied on only $\log^2 n$ elements
3. **o(√n) bandwidth**, in order to achieve sublogarithmic read efficiency → $o(\sqrt{\log^2 n}) = o(\log n)$



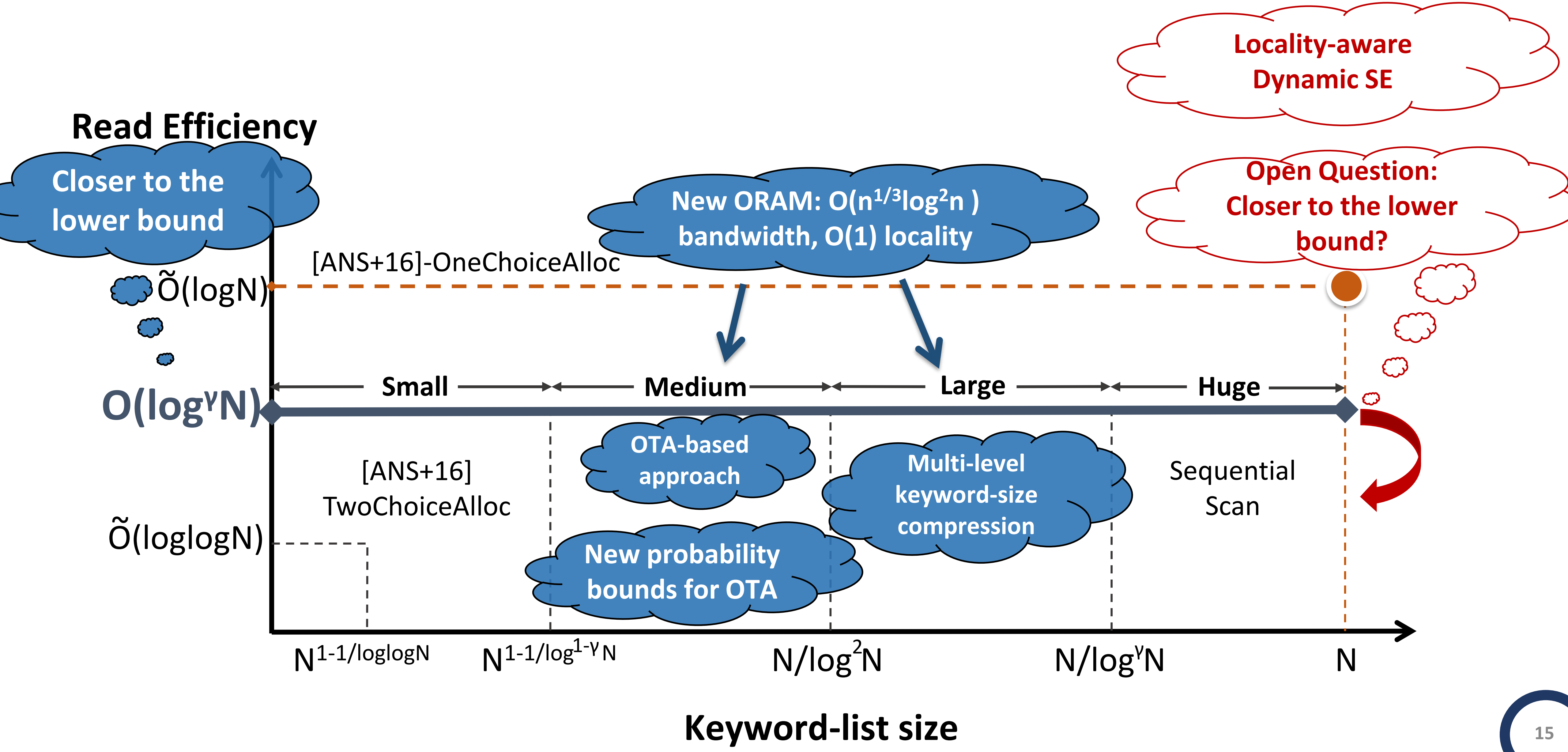


Our Approach: OTA Stashes

Important: max ≤ $N/\log^2 N$ for maintaining $O(N)$ index size

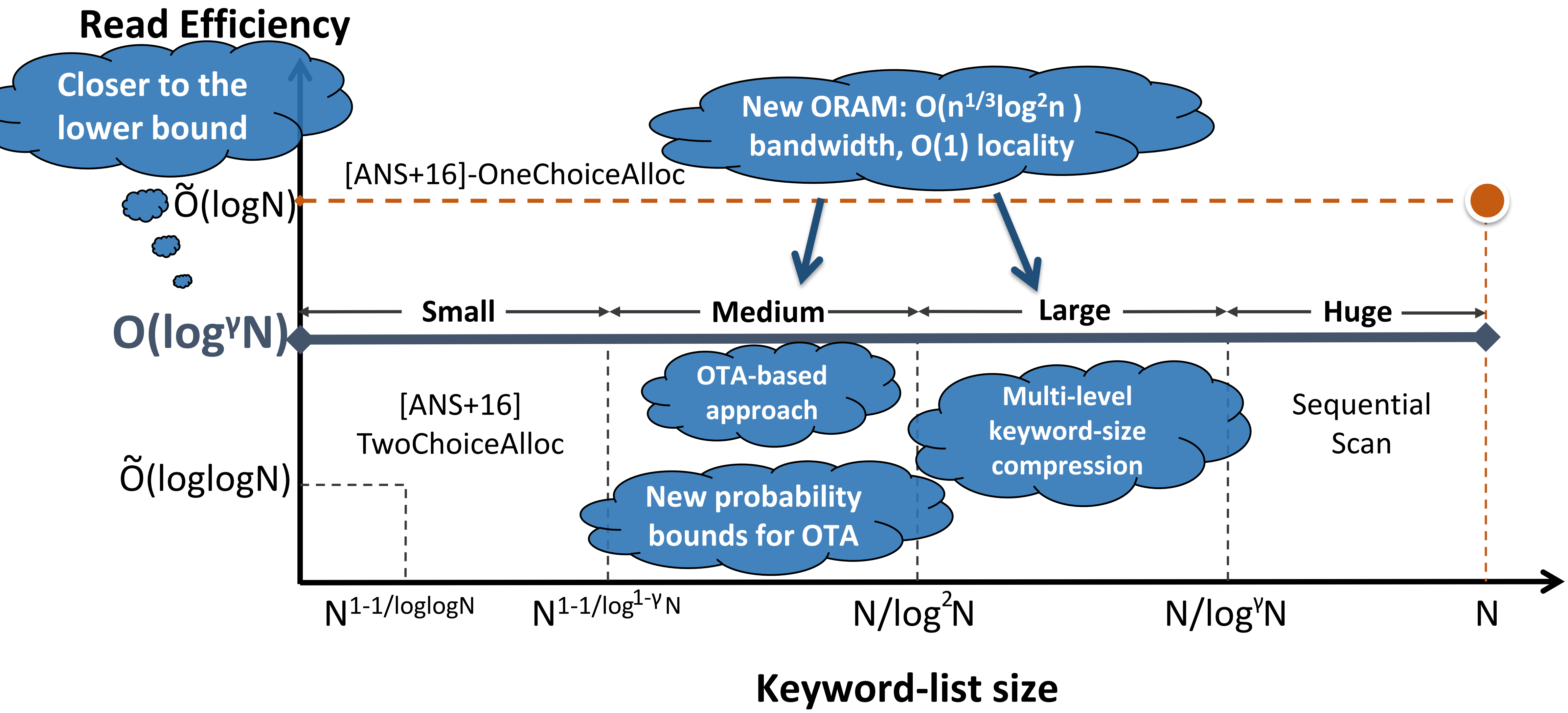


Conclusion – Future Work?



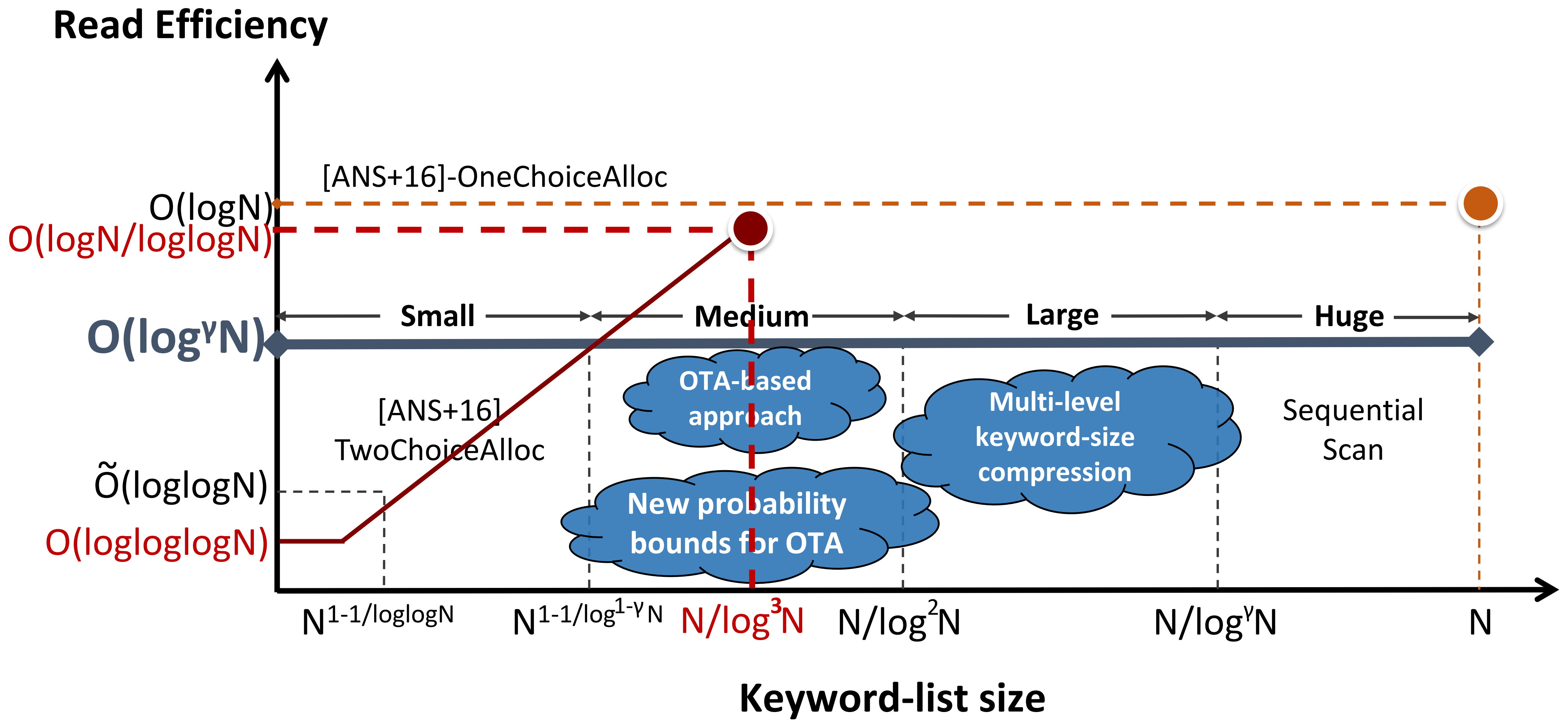
Thank You!

https://eprint.iacr.org/2017/749

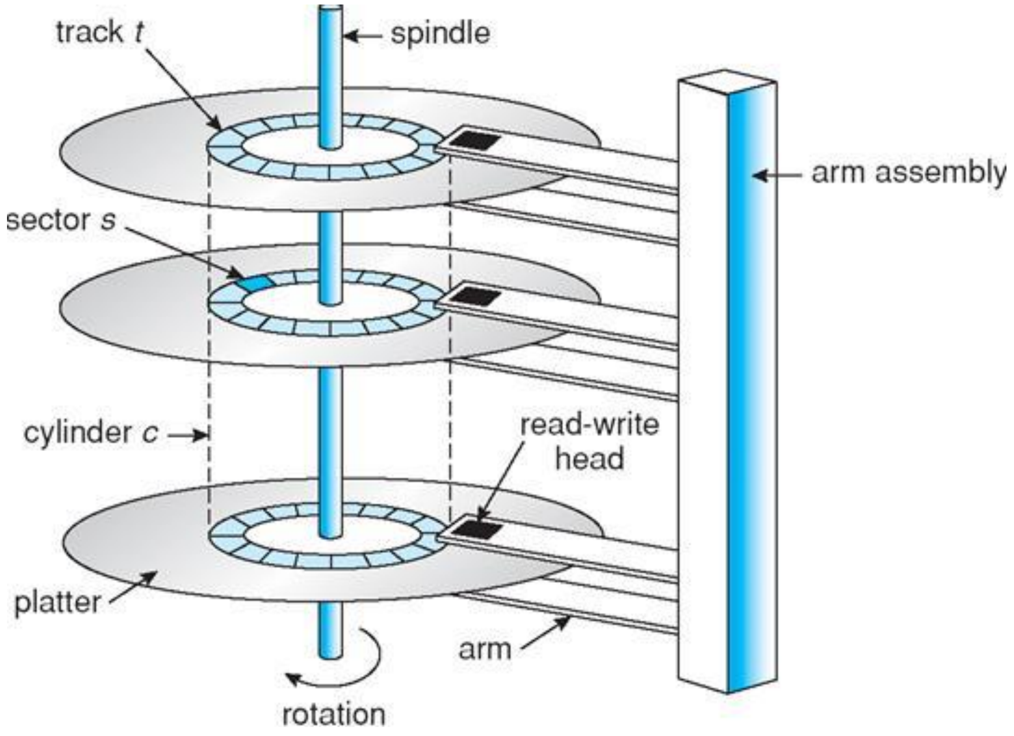


[ASS18] in CRYPTO

$O(N)$ space, $O(1)$ locality and $\omega(1)\cdot\epsilon(n)^{-1}+O(\log\log\log N)$ read efficiency where $n = N^{1-\epsilon(n)}$



Studying locality for HDD



$$\text{Access Cost} = \underbrace{(\text{seek time}) + (\text{rotational delay})}_{\text{Random I/O Cost}} + \underbrace{(\text{transfer time})}_{\text{Sequential I/O Cost}}$$

Random I/O Cost: **~4-12 ms**

Sequential I/O Cost: **~10 µs for 1 byte**

Studying locality for SDD

Samsung 960 Pro M.2 NVMe SSD

	Read	Write	Locality
Sequential Transfer Page size = 2MB	2222.93 MB/sec	1786.72 MB/sec	High
Random Transfer Page size = 2MB	1339.76 MB/sec	1237.57 MB/sec	
Random Transfer Page size = 2KB	34.30 MB/sec	150.83 MB/sec	Low

More detailed analysis → http://www.storagereview.com/samsung_960_pro_m2_nvme_ssd_review

Studying locality for RAM