A New Upper Bound on Cache Hit Probability for Non-anticipative Caching Policies

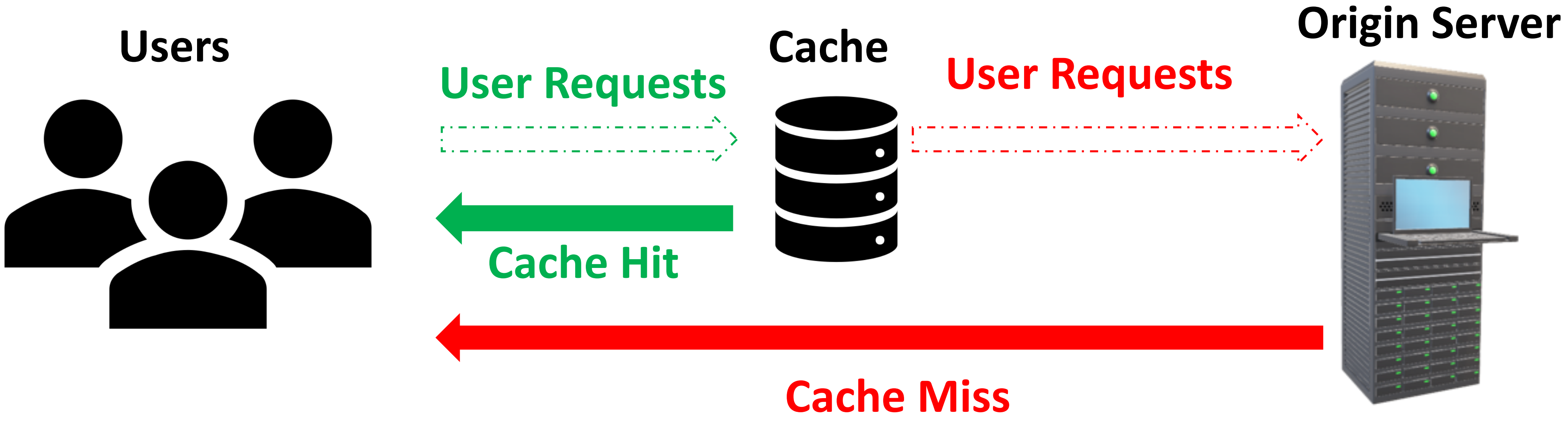
Nitish K. Panigrahy, Philippe Nain, Giovanni Neglia, Don Towsley

Performance 2020

Talk Outline

- ❑ Motivation
- ❑ Background
- ❑ Hazard Rate Based Upper Bound
- ❑ Extension to Variable-size Objects
- ❑ Trace Driven Simulation
- ❑ Future work

Caching Objective: Maximize Hit Probability



Maximize **Object Hit Prob:** $\dfrac{\#\ of\ reqs\ served\ from\ cache}{total\ \#\ of\ reqs}$

Caching Dimensions

- ❑ When to store in cache
  - **Prefetching**: Store before needed
  - Non-prefetching: Store on request
- ❑ To store or not
  - **Admission**: May not store the requested object
  - Eviction: Must store the requested object
- ❑ Knowledge of future
  - Anticipative: Entire request trace is known
  - **Non-anticipative**: Only request history is known

Many Caching Policies….

GDSF

LFU-DA

LRU-K

LFU

HYPERBOLIC

LRFU

CAR

MQ

PLRU

LRU

ARC

FIFO

LIRS

LHD

LFF

GDLFU

GDS

ADAPTSIZE

Upper bound on obj hit prob

How to evaluate the performance of these policies?

State of the Art (Upper Bounds)

Equal size objects

- Independent Reference Model
  - Statically cache most popular objects
- Arbitrary requests: **Belady's** MIN
  - Evict the object whose next request is farthest in future
  - **Anticipative and non-prefetching**

Variable size objects

- Finding OPT is NP-hard
  - Approximate solutions exist
- Upper bound on object hit probability
  - **FOO and PFOO** methods [Berger et al, Sigmetrics'18]
  - **Anticipative and non-prefetching**

Questions:

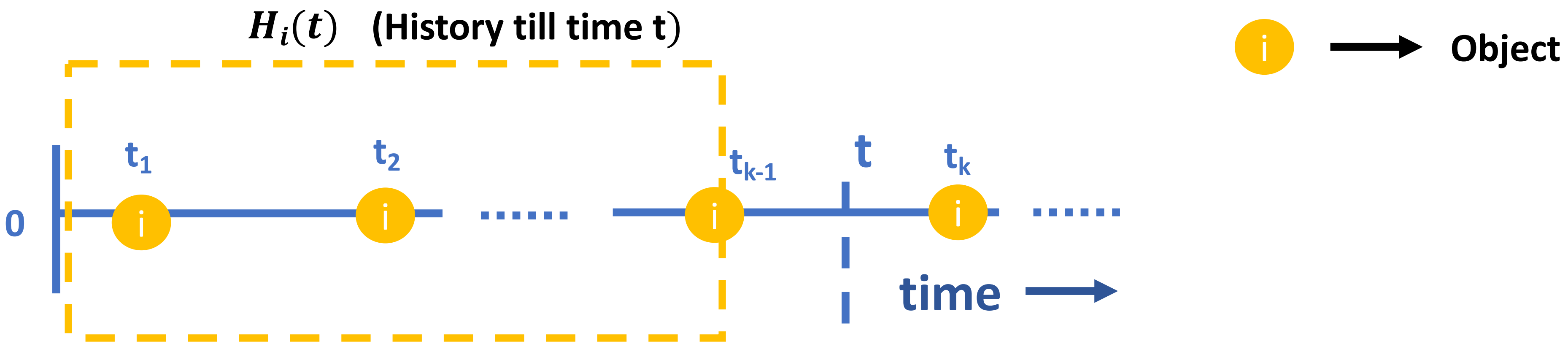
1. Online upper bound with limited statistical knowledge of object request pattern?
2. More general assumptions than IRM?

Non-anticipative and prefetching

Solution:

1. Our Approach: Hazard rate based ordering

Background: Hazard Rate Function



$$\boxed{\lambda_i^*(t)} = \frac{f_i(t|H_i(t))}{1 - F_i(t|H_i(t))}.$$

Conditional density function

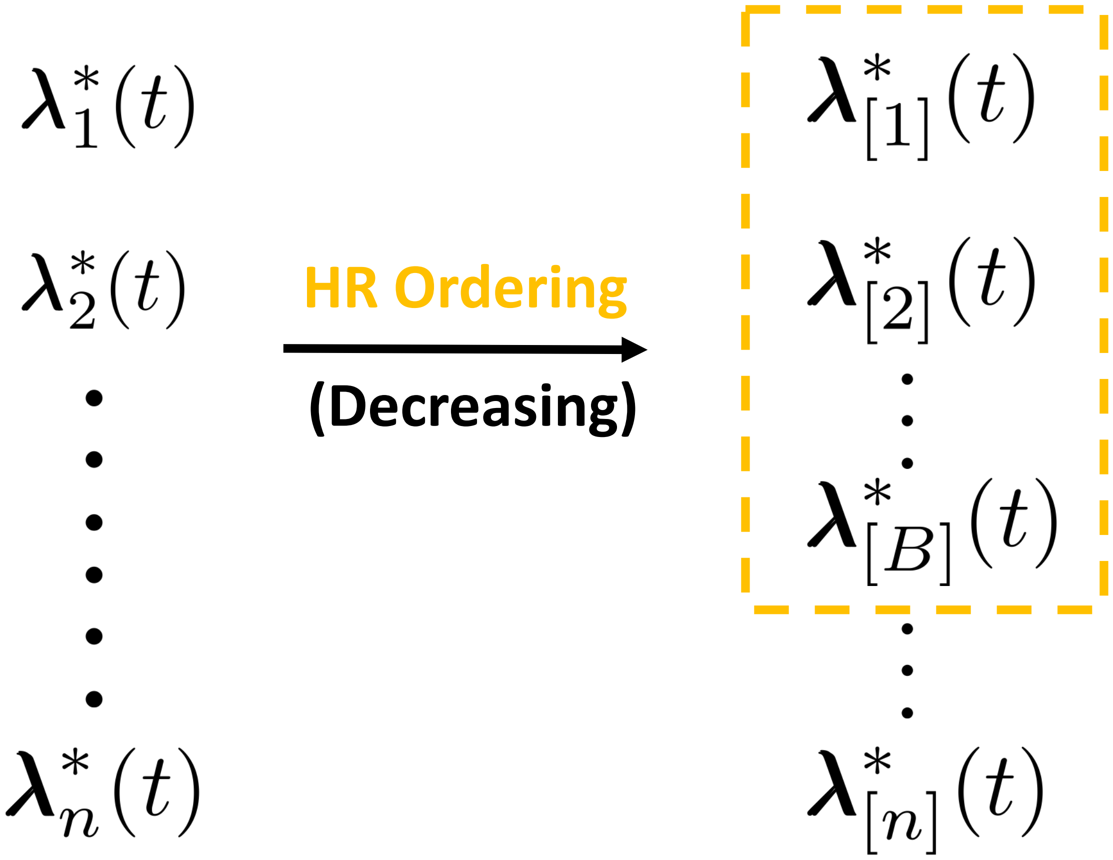
Conditional ccdf

Hazard Rate Function

System Model

- ❑ Single Cache; Size: B
- ❑ n objects: $\{1, \ldots, n\}$
- ❑ Equal-size objects
- ❑ Minimal assumptions on object request processes
    - Can be **dependent** processes
    - HR function should be well defined at all points of time

Our Approach: Hazard Rate Based Ordering



$\boldsymbol{\lambda}_1^*(t)$

$\boldsymbol{\lambda}_2^*(t)$

$\vdots$

$\boldsymbol{\lambda}_n^*(t)$

HR Ordering

(Decreasing)

$\boldsymbol{\lambda}_{[1]}^*(t)$

$\boldsymbol{\lambda}_{[2]}^*(t)$

$\vdots$

$\boldsymbol{\lambda}_{[B]}^*(t)$

$\vdots$

$\boldsymbol{\lambda}_{[n]}^*(t)$

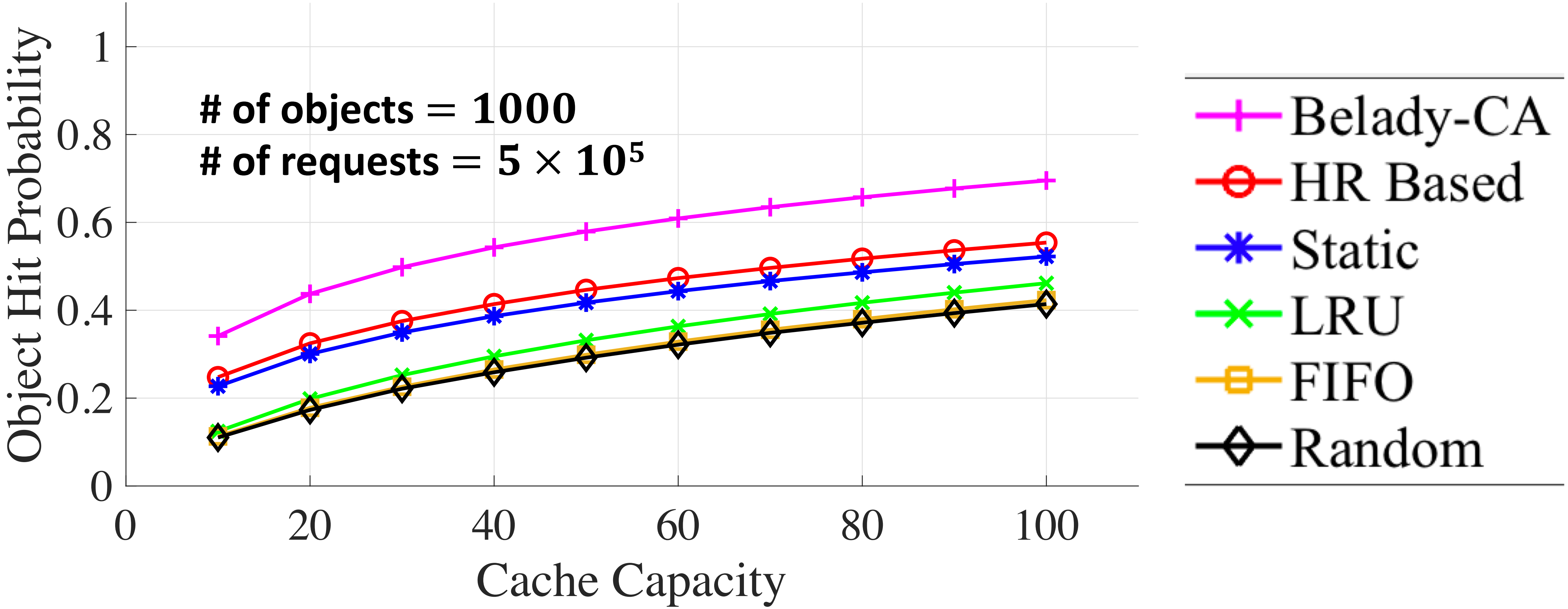
Hit if requested object belongs to one of top B HRs

HR upper bound similar to instantaneous LFU

Results: Equal-size Objects

HR Based bound tighter than **Belady's**

Similar results for other irt distributions



Generalized Pareto inter-request times

Extension to Variable Sized Objects

❑ Upper bound on number of byte hits

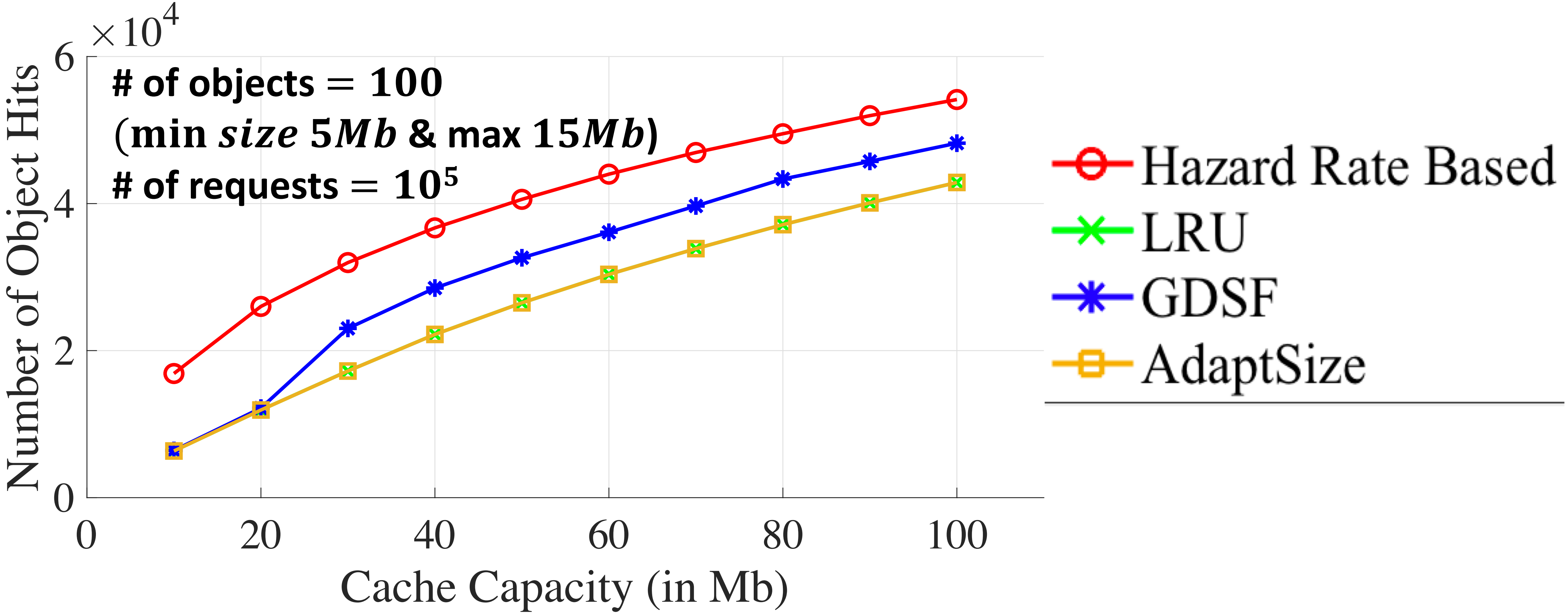
- Normalize by object sizes
- Maps to **fractional knapsack problem**

❑ Upper bound on number of object hits

- Maps to $\mathbf{0-1}$ **knapsack problem**
- Order by **HR/size**

Results: Variable-size contents

HR Based bound indeed an upper bound!!



# of objects = 100

(min *size* 5*Mb* & max 15*Mb*)

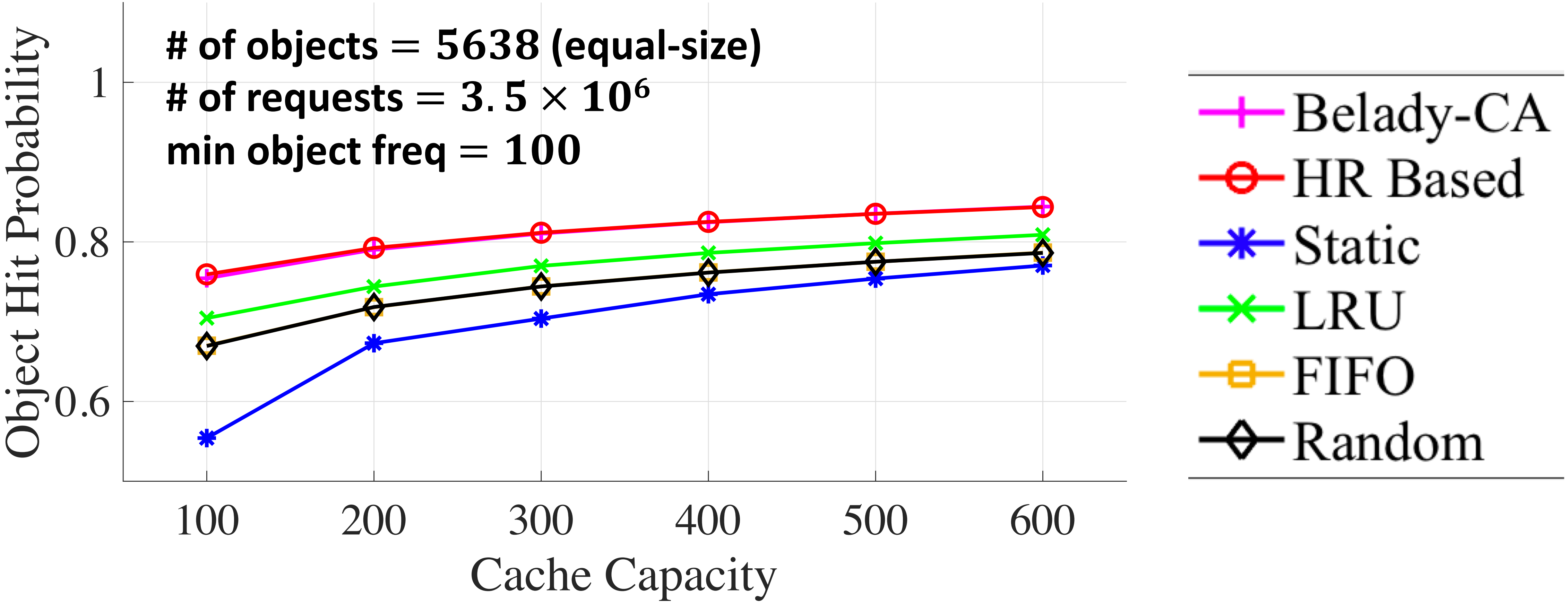
# of requests = 10^5

Generalized Pareto inter-request times

Preliminary Results: Real-world Data Trace

Parametric Fitting to Generalized Pareto Distribution

HR Based bound still an upper bound!!


# of objects = 5638 (equal-size)
# of requests = 3.5×10^6
min object freq = 100

IBM Web Access Trace

Future Directions

❑ Prefetching cost for realizable Hazard Rate Based Policy

❑ Non-parametric HR estimators

- Gaussian kernel density estimators

❑ Online fitting of distributions and HR estimation

❑ Closed-form upper bound?

Questions