SMALL CACHE, BIG EFFECT: PROVABLE LOAD BALANCING FOR RANDOMLY PARTITIONED CLUSTER SERVICES

Bin Fan, Hyeontaek Lim, David G. Andersen, Michael Kaminsky* (CMU, *Intel)

GOAL: SCALE SYSTEM THROUGHPUT LINEARLY AS ADDING SERVERS

Observation:

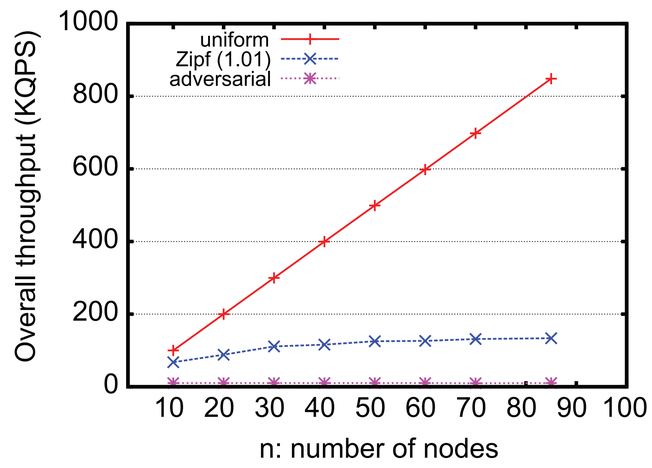
- Load balance is often workload dependent

Example: 85-node FAWN key-value cluster

- 10K reqs/sec per node for key lookups
- Hash-based partition: nodeID = Hash(key)
- Uniformly access keys: tput scales linearly
- Biased access: underutilize system capacity

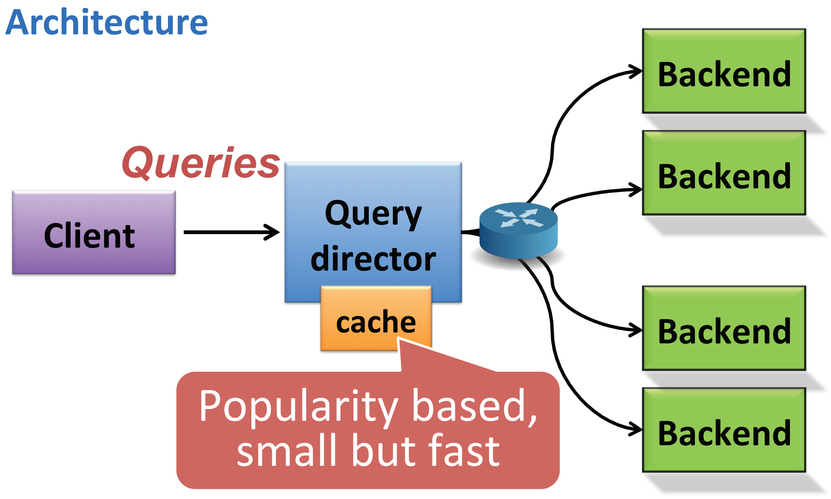
Question:

- Can we provide workload-independent load balance?



SMALL CACHE: EFFECTIVE TO ASSIST LOAD BALANCE

Architecture



Intuitition:

Skewed workload, but cache friendly ⟷ Unfriendly to cache, but uniform workload

Major Result:

- If cache size = k * n * log n,
  tput > (1-ε) * total capacity, regardless of workload and total number of items
  - n: # nodes,
  - k: a small and tunable constant factor

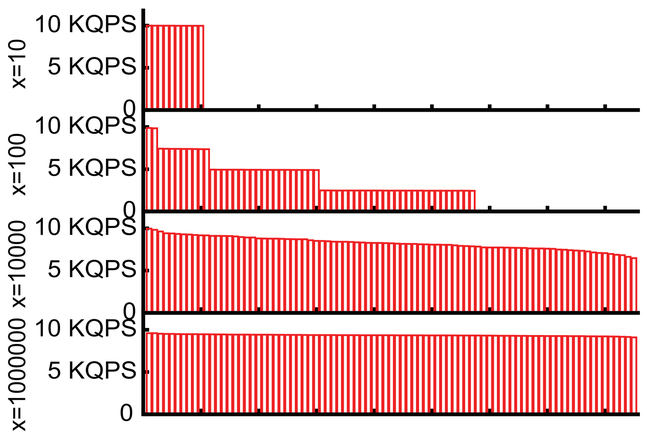
Requirement

- Hash-based service partition
- Service partition opaque to clients
- Cacheable queries

EVALUATION: FRONTEND 900K REQ/S; BACKEND 10K REQ/S

Work distribution w/o cache

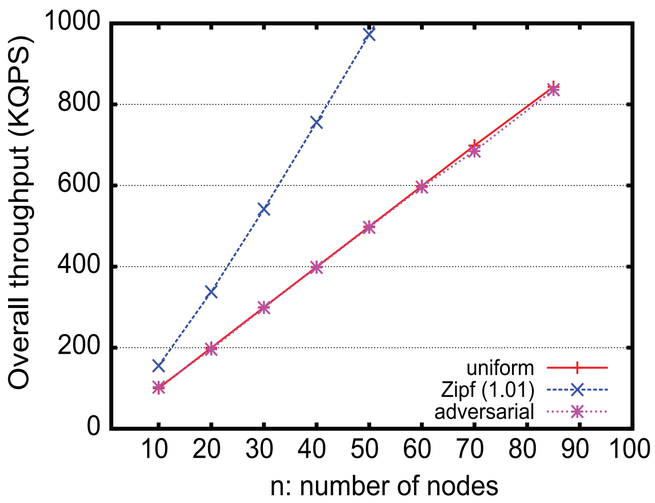
- x: working set size



Larger working set yields better balanced load

Scalability w/ cache

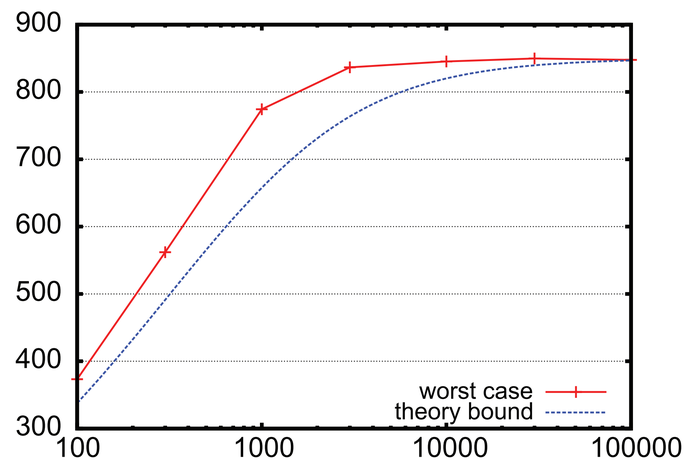
- cachesize = 8 n logn



Worst case tput very close to uniform case

Analytical vs Empirical

- number nodes = 85



Analytical bound is accurate

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley