Placing Dynamic Content in Local Caches

Moez Draief

Huawei Maths and Algo Lab Paris
and
Imperial College London

joint with M. Leconte, G. Pascos, S. Chouvardas (Huawei)

5G wireless networks

- Congestion at the edge:
  By 2019, increase of 57% of mobile data to 24,3 exabytes per month [Cisco Feb. 2015]
- Congestion at the core due cloud services and Machine-to-Machine communication (IoT)

- To ease congestion 5G wireless architectures with stringent latency constraints advocate placing content near the user, e.g., at a base station (femtocells)
- Cost of deployment caches is negligible compared to that of a base station.

But the storage capacity is much smaller than the content catalogue size.

CDNs with geographical locality

- Content Delivery Networks (CDNs) use proactive data replication to respond to demand for popular content.
- This paradigm yields benefits for performance of networks, e.g. reducing latency and saving bandwidth.
- In a wireless network, content can be stored closer to the user,

CDNs with geographical locality

- Content Delivery Networks (CDNs) use proactive data replication to respond to demand for popular content.
- This paradigm yields benefits for performance of networks, e.g. reducing latency and saving bandwidth.
- In a wireless network, content can be stored closer to the user,

Poor hit rates: small local caches only see small samples.

CDNs with geographical locality

- Content Delivery Networks (CDNs) use proactive data replication to respond to demand for popular content.
- This paradigm yields benefits for performance of networks, e.g. reducing latency and saving bandwidth.
- In a wireless network, content can be stored closer to the user,

Poor hit rates: small local caches only see small samples.

How can one achieve good hit rates for small local caches?

CDNs with temporal locality

- Fresh content such as news, music or TV series is ephemeral
- Tracking an ever changing popularity profile of content is challenging

CDNs with temporal locality

- Fresh content such as news, music or TV series is ephemeral
- Tracking an ever changing popularity profile of content is challenging

How to do caching with small population under the assumption of time-varying and unknown content popularity?

Related work

- In the case of one-base station, [Gunduz-Blasco'14] propose a (knapsack) Bandit approach.
- [Bastug et al '14] propose to use side information to learn popularity.
- [Massoulié et al '15] propose (market) mechanisms to optimize other aspects of the problem such as bandwidth load.

Our contribution

Caching and learning time-varying popularities of (first) **chunks of files**.

- Threshold policy to store content
- Study local versus global popularity estimation
- Score-gated Least-Recently-Used (LRU) to prefetch content

Outline

Time varying popularity

Caching with multiple locations

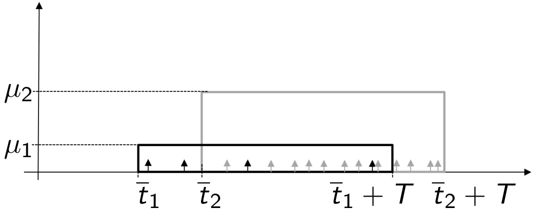
Outline

Time varying popularity

Caching with multiple locations

Poisson-shot noise model (SNM)

[Traverso et al '13] show that SNM fits well real mobile content requests.



- Shot arrival times form a Poisson process with rate λ.
- Pulses are rectangular of fixed duration T.
- Shot volume of content m is μ_m drawn from a power-law distribution with parameter α
- Requests for content m are generated using PP with parameter μ_m

Assumptions

- Only store first chunk of the file.
- Controller knows the exact arrival times $\bar{t}_m$.
- There is no cost for replacing content.

Goal: maximize hit rate over the time horizon, i.e. fraction of demands found in cache.

Hit rate optimization

- At time t the alive content catalogue is given by the set

$$\mathcal{M}(t) = \{m : \bar{t}_m \leq t \leq \bar{t}_m + T\}.$$

- for $m \in \mathcal{M}(t)$, its age is $\tau_m(t) = t - t_m$ and $N_m(t)$ the number of requests for m up to time t
- Let $y_m(t)$ the indicator whether m is in cache at time t, where $\sum_m y_m(t) \leq C$

The challenge lies in learning the μ_ms to optimize the average hit rate.

$$H(y) = \sum_m y_m \mu_m$$

Goal: find

$$y^*\big((N_m),(\tau_m)\big) = \underset{\substack{\forall m,\, y_m \in \{0,1\} \\ \sum_{m\in\mathcal{M}} y_m = C}}{\arg\max} \sum_{m\in\mathcal{M}} y_m \mathbb{E}[\mu_m | N_m, \tau_m]\,.$$

Estimating parameters

Given $N_m(t)$ and τ_m both observed, (numerically) compute

$$\mathbb{E}[\mu_m | N_m, \tau_m] = \frac{\int_{\mu_m} \mu_m \mathbb{P}(N_m | \mu_m, \tau_m) f(\mu_m) d\mu_m}{\int_{\mu_m} \mathbb{P}\left(N_m | \mu_m, \tau_m\right) f(\mu_m) d\mu_m} \qquad (1)$$

where f is the power-law density, and

$$\begin{aligned} \mathbb{P}(N_m | \mu_m, \tau_m) &= \mathbb{P}\left(Pois(\mu_m \tau_m) = N_m\right) \\ &= (\mu_m \tau_m)^{N_m} \frac{e^{-\mu_m \tau_m}}{N_m!}. \end{aligned}$$

Age-Based Threshold (ABT) Policy.

Parameter Selection.
Choose θ to be the $\gamma_c = C/\lambda T$[1]-th upper-percentile of $F_{\widehat{\mu}_m}$ empirical distribution of the μ_ms

$$\theta(\gamma_c) = F^{-1}_{\widehat{\mu}_m}(1 - \gamma_c).$$

Age-Based Threshold. Choose the threshold $\tilde{N}(\tau)$

$$\widetilde{N}(\tau) = \min\{k \in \mathbb{N} : \mathbb{E}[\mu_m | N_m = k, \tau_m = \tau] \geq \theta(\gamma_c)\}$$

Caching Vector. For each content $m \in \mathcal{M}$ observe N_m, τ_m and choose:

$$y_m = \begin{cases} 1 & \text{if } N_m \geq \widetilde{N}(\tau_m), \\ 0 & \text{otherwise.} \end{cases}$$

Ensuring Cache Size Constraint. If $\sum_m y_m > C$, then choose arbitrarily $\sum_m y_m - C$ contents and set $y_m = 0$.

[1] Fraction of catalogue to be stored

Computing the Thresholds $\widetilde{N}(\tau)$

- Offline: For given parameters C, λ, T and those of the power-law distribution compute the thresholds for different values of τ.

- Water-filling heuristic: Split time into small intervals (fraction of T) and increase threshold at each interval inspecting the marginal hit rate improvement, restart the threshold every T-interval.

Optimality in Many Contents Regime

Theorem Let optimal policy $\pi^*(\lambda, T)$. For $\lambda, C \to \infty$, $\lim_{\lambda\to\infty} \frac{C}{\lambda T} = \gamma_c$. Then a.s.

$$\lim_{\lambda\to\infty} \pi^*(\lambda, T) = \text{ABT},$$

in the sense that they asymptotically have the same threshold function, and thus they cache the same contents.
Moreover, ABT is almost surely asymptotically optimal.

Outline

Time varying popularity

Caching with multiple locations

Aggregation: uncorrelated traffic

Proposition Consider SNM with requests $(N^l_m(\tau))_{m,l,\tau}$ are observed by the global system, and thinned version $(N^l_m(\tau))_{m,\tau}$ observed by local cache $l \in \mathcal{L}$.

If maximum hit rate performance of global system is $h^*_{\mathcal{L}}(T)$ and that of local cache is $h^*_l(T)$

$$h^*_{\mathcal{L}}(T/L) = h^*_l(T), \quad \forall T > 0.$$

Model for correlated popularities

We propose here a model for correlated local popularities (μ_m^l).

- Content m with feature vector X_m iid uniform in $[0,1]$.
- Location l with feature vector Y_l iid uniform in $[0,1]$.
- $K(x,y) = g(|x-y|)$, where g is continuous, strictly decreasing on $[0,1/2]$, symmetric and 1-periodic.

Popularity of content m at cache l is

$$\mu_m^l = \mu_m^{\mathcal{L}} \frac{K(X_m, Y_l)}{\sum_{l' \in \mathcal{L}} K(X_m, Y_{l'})}, \quad \forall m, l.$$

where $\mu_m^{\mathcal{L}}$ is aggregate popularity of m drawn from a power-law dist.

Local is More Accurate - Known Popularities

Theorem Assuming known popularities (or equiv. assume $T \to \infty$), the hit rate performance of local learning is higher than the aggregate global learning. Furthermore, as the number of edge caches $L \to \infty$ the maximum expected hit rate is

$$\lim_{L\to\infty} h_l^*(\infty) = \frac{1}{\overline{\mu}}\mathbb{E}\Big[\mu_m^l \mathbf{1}(\mu_m^l \geq \theta^l)\Big]$$

$$= \int \left\{ 2 \int_0^{g^{-1}\left(\frac{L\theta^l}{\mu_m^{\mathcal{L}}}\right)} g(t)\, dt \right\} \frac{\mu_m^{\mathcal{L}}}{L}\, dZ_m,$$

where θ^l is the unique value satisfying

$$\gamma_c = \mathbb{P}\Big(\mu_m^l \geq \theta^l\Big) = 2\int g^{-1}\left(\frac{L\theta^l}{\mu_m^{\mathcal{L}}}\right) dZ_m.$$

and $\mu_m^{\mathcal{L}} = \overline{\mu}(1-\alpha)Z_m^{-\alpha}$.

Our solution

Remember that we stored content if

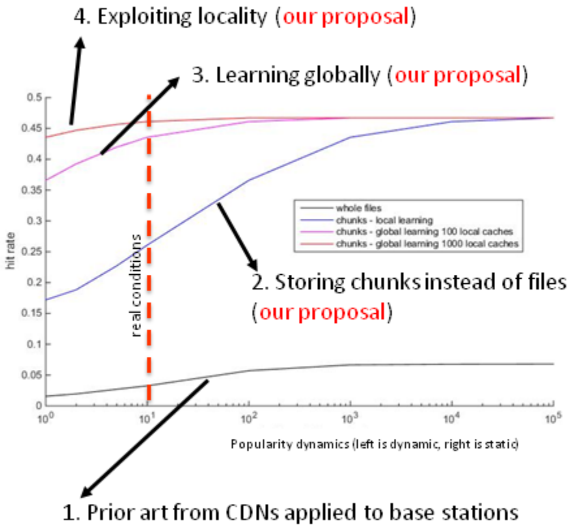
$$N_m \geq \widetilde{N}(\tau) = \min\{k \in \mathbb{N} : \mathbb{E}[\mu_m | N_m = k, \tau_m = \tau] \geq \theta(\gamma_c)\}$$

- Cluster locations to define thresholds and use clustered ABT.
- Score-gated LRU: $1 \geq \beta_1 > \gamma_c$, assuming a larger virtual cache, never cache content such that

$$N_m \leq \min\{k \in \mathbb{N} : \mathbb{E}[\mu_m | N_m = k, \tau_m = \tau] \geq \theta(\beta_1)\}$$

- Perform LRU on the rest
- Prefetch content that is deemed superpopular by global controller, i.e. $0 \leq \beta_2 < \gamma_c$, assuming a smaller virtual cache, cache everywhere content such that

$$N_m \geq \min\{k \in \mathbb{N} : \mathbb{E}[\mu_m | N_m = k, \tau_m = \tau] \geq \theta(\beta_2)\}$$


4. Exploiting locality (our proposal)
3. Learning globally (our proposal)
2. Storing chunks instead of files (our proposal)
1. Prior art from CDNs applied to base stations
hit rate
real conditions
Popularity dynamics (left is dynamic, right is static)
whole files
chunks - local learning
chunks - global learning 100 local caches
chunks - global learning 1000 local caches
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
10^0
10^1
10^2
10^3
10^4
10^5

Conclusion

- Models and algorithmic solutions that led to the design of an LRU score-gated algorithm with prefetching.
- Numerous systems assumptions are unrealistic (no-replacement cost, bandwidth): [Maggi et al '15].
- A more general problem: how to solve the small-sample problem? How to cluster "Cluster caches"?