Deployment vs. Data Retrieval Costs for Caches in the Plane

Mihaela Mitici, Jasper Goseling, Maurits de Graaf and Richard J. Boucherie

***Abstract*—We consider the problem of finding the Pareto front of the expected deployment cost of wireless caches in the plane and the expected retrieval cost of a client requesting data from the caches. The data is allocated at the caches according to partitioning and coding strategies. We show that under coding, it is optimal to deploy many caches with low storage capacity. For partitioning, we derive a simple relation between the cost of the cache deployment and the cost of retrieving the data from the caches. Lastly, we show that coding results in a lower Pareto front than partitioning.**

I. Introduction

We consider wireless caches placed in the plane according to a homogeneous Poisson process. A client arriving at a random location in the plane is interested in retrieving a large data file that is stored at the caches. Since the storage capacity of the caches is limited, the file needs to be stored in a distributed fashion. Thus, the client needs to retrieve data fragments from several caches to recover the complete file.

Data fragments can be stored at the caches according to various strategies. We study two storage strategies: partitioning and coding. Partitioning is a storage strategy according to which the data is divided into equally-sized fragments. Replicas of the data fragments are stored at the caches. In the coding strategy, each cache stores a random linear combination of the fragments.

We focus on two cost measures and their Pareto front. The first cost measure is the deployment cost of the caches in the plane, defined to be proportional to the storage capacity of the caches and the density of the Poisson process according to which the caches are deployed. The second cost measure is the cost for a client to retrieve the file from the caches. This is the cumulative cost of obtaining individual data fragments from specific caches. The retrieval cost is increasing in the distance between the client and the contacted caches. Figure 1 (Partitioning) and Figure 2 (Coding) illustrate how the retrieval cost is affected by the choice of the density of the cache deployment, the storage capacity of the caches and the way the data is stored. One could reduce the retrieval cost by increasing the density of the caches, *i.e.*, the average number of caches per unit area. In this case, however, the deployment cost would increase. A conflict arises between the deployment of the caches in the plane and the cost of retrieving the data.

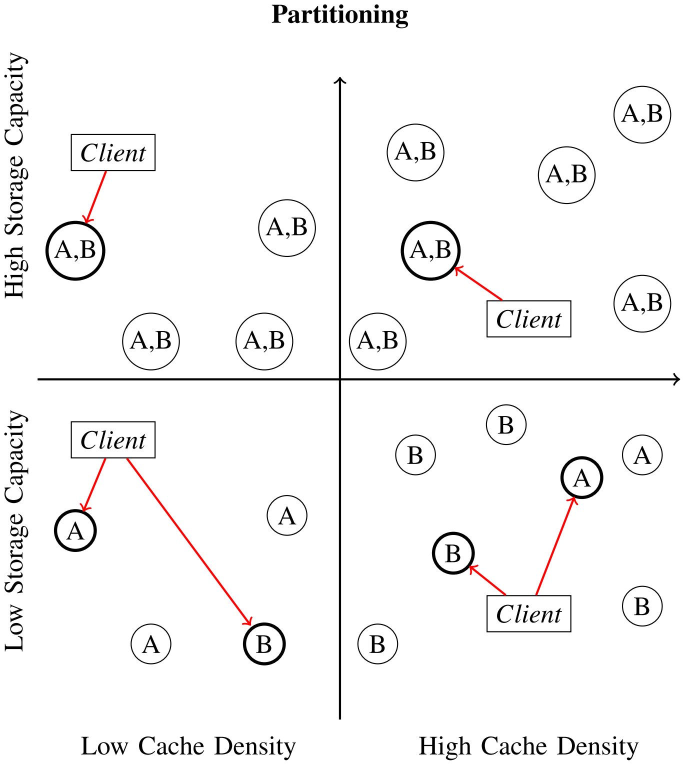

Fig. 1: Cache Deployment vs. Data Retrieval Costs under Partitioning. The data file consists of symbols *A* and *B*. A client needs to retrieve all symbols from the caches. Contacting far away caches increases the retrieval cost for the client. A high cache density 'brings' the cache closer to the client, but increases the deployment cost.

In the current work, we analyze the Pareto front of the deployment and the data retrieval costs. The parameters over which we optimize are: i) the density of the Poisson process according to which caches are deployed and ii) the storage capacity of an individual cache.

Data replication and coding for caches have been studied in [1]–[6]. In [1], the authors consider the optimal number of replicas of data such that the distance between a requesting node and the nearest replica is minimized. Data sharing among multiple caches such that the bandwidth consumption and

This work was performed within the project RRR (Realisation of Reliable and Secure Residential Sensor Platforms) of the Dutch program *IOP Generieke Communicatie*, IGC1020, supported by the *Subsidieregeling Sterktes in Innovatie*, and is partly supported by the *Netherlands Organisation for Scientific Research* (NWO), grant 612.001.107.

M. Mitici is with the Department of Applied Mathematics, University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands (email: m.a.mitici@utwente.nl)

J. Goseling is with the Department of Applied Mathematics, University of Twente, and with Faculty of EEMCS, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands. (email: j.goseling@utwente.nl)

M. de Graaf is with Thales B.V. Nederland and with the Department of Applied Mathematics, University of Twente (email: maurits.degraaf@nl.thalesgroup.com)

R.J. Boucherie is with the Department of Applied Mathematics, University of Twente.(email: r.j.boucherie@utwente.nl)

the data retrieval delay are minimal, is considered in [2]. In [3] coded data allocation at the caches is investigated such that any sufficiently large subset of caches can provide the complete data. In [4] coding strategies for networks of caches are presented, where each user has access to a single cache and a direct link to the source. The authors show how coding helps in reducing the load on the link between the caches and the source. The impact of non-orthogonal transmissions for coding strategies is considered in [5]. The authors derive scaling results are on the best achievable transmission rates. In this paper, we assume that the data transmissions from the caches to the client are orthogonal, by separating them in time, frequency or coding space.

The geometry of the cache deployment under partitioning and coding has been studied in [7]. The authors consider caches randomly placed in the plane and show that, for a general data retrieval cost function, the coding strategy outperforms partitioning. Closed form expressions for the retrieval costs under partitioning and coding are derived. The retrieval cost is seen as an increasing function of the distance between the client and the contacted caches. Thus, the deployment of the caches in the plane directly influences the retrieval process. This has motivated us to investigate in [8] the Pareto front of the cache deployment and data retrieval costs.

We computed in [8] the Pareto front of the two costs by relaxing the integrality condition on the cache storage size and considered arbitrary storage capacities. By doing so, we have obtained a superset of the Pareto front for integer capacities.

In this paper, we extend our analysis from [8] to integer storage capacity for the caches and provide an exact description of the Pareto front. From a practical point of view, the proposed model characterizes more closely the wireless cache system and provides valuables insight into methods to tune the cache density and storage capacity such that a targeted performance is achieved. In addition, we have removed the constraints on the cache density, as used in [8], thereby obtaining a concise description of the Pareto front.

We show that under partitioning, the Pareto front depends only on the ratio of the deployment density and the storage capacity of the caches. Thus, when deploying caches, one has some flexibility in either increasing the density or the capacity of the caches. For the coding strategy, however, we show that it is optimal to deploy many caches with low storage capacities. We also show that the optimal coding strategy results in a lower Pareto front than partitioning. In conclusion, even though the optimal coded strategy leads to lower costs, it has the disadvantage that it provides less flexibility in deployment.

The remainder of this paper is organized as follows. In Section II we define the model and formulate an exact problem statement. We compute the Pareto front for the retrieval time and the cache deployment cost in Section III. In Section IV we provide final remarks and conclusions.

II. Problem Statement

We consider a data file of n symbols, with $n > 1$ fixed, which is stored at the caches. The symbols are elements of a finite field $\mathbb{F}_q$.

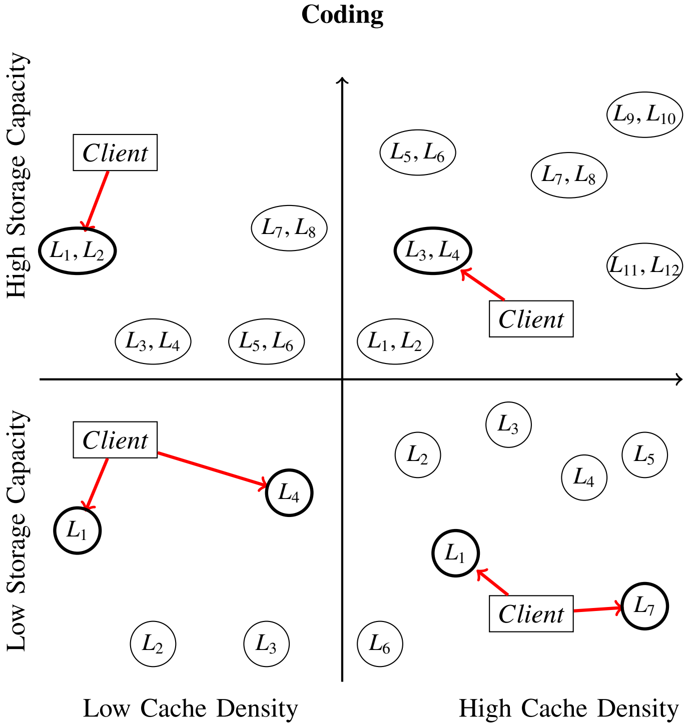

Fig. 2: Cache Deployment vs. Data Retrieval Costs under Coding. Caches store linear combinations $L_i, i = 1, .., 12$, where $L_i = \alpha_i A + \beta_i B$.

Caches are placed in the plane according to a homogeneous Poisson process with density λ, where $\lambda > 0$ is a parameter over which we optimize. The caches have limited storage capacity which we express in terms of an integer k, the second optimization parameter. A cache stores n/k symbols. To ensure integral cache capacity, we impose that k divides n, denoted by $k|n$. Thus, $1 \leq k \leq n$ and n/k is an integer.

A client arriving at a random location in the plane is interested in retrieving the file from the caches. We assume that the client has complete knowledge about the content and the location of the caches. The client requests data from a set of k caches that ensure the recovery of the file.

Under partitioning (P), the file is divided into k different fragments, each of n/k symbols. Each cache selects uniformly at random a fragment to store. A client requests fragments from k closest caches such that all k fragments are distinct.

Under coding (C), each cache stores a random linear combination of the k fragments. The closest set of k caches is chosen to decode the file. There is a positive probability that k random linear combinations are not linearly independent and, therefore, do not provide the entire data. In this case, the client needs to request data from caches that are located further away. It was demonstrated in [7] that this has a negligible impact on the retrieval cost. Therefore, we restrict our attention to the cost of retrieving the data from the k nearest caches.

The cost measures, which are a function of the model parameters k and λ, are defined as follows:

i) The expected data retrieval cost, denoted by C_r^A, where $A \in \{P, C\}$

Let the cost of retrieving data from k caches located at distances $\delta_1, \ldots, \delta_k$ be

$$\mathbf{C}_r^A(\delta_1, .., \delta_k) = \frac{1}{n} \sum_{i=1}^{k} \frac{n}{k} \delta_i^{2\alpha}, \tag{1}$$

where $\alpha \geq 1/2$ is an arbitrary, but fixed, parameter denoting the path loss exponent in the wireless medium. In (1), $\frac{n}{k}\delta_i^{2\alpha}$ is the cost of retrieving n/k symbols from a cache at distance δ_i away from the client. We normalize the retrieval cost by n. Hence, the cost depends only on the parameters λ and k. We are interested in the expected cost $C_r^A(k, \lambda)$, where the expectation is over the randomness in the spatial Poisson process.

ii) The expected deployment cost of the caches in the plane per unit area, denoted by C_d.

The cost of deploying a single cache is proportional to the cost of storing n/k symbols and the deployment density $\lambda > 0$ of the caches in the plane. We again normalize the cost by n. Therefore, the expected deployment cost per unit area is defined as:

$$C_d(k, \lambda) = \frac{\lambda}{k} , \tag{2}$$

where the expectation is over the randomness in the spatial Poisson process and over the randomness of the storage strategy.

We consider the multi-objective optimization problem which aims at minimizing the expected deployment cost $C_d(k, \lambda)$ and the expected retrieval cost $C_r^A(k, \lambda)$ under the storage strategy $A \in \{P, C\}$.

We will make use of the gamma function, which for $x > 0$ is represented as $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ and the digamma function, $\psi(x) = \frac{d}{dx} \ln \Gamma(x) = \int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt$ (see, for instance, [9]).

III. Analysis

In general, a single point simultaneously minimizing two conflicting objectives does not exist, in which case the multi-objective problem does not have a unique optimal solution. Therefore, we characterize the Pareto front [10] of the expected cache deployment and data retrieval costs. More precisely, we analyze which sets of objective values are achievable and non-dominated. We refer to [11] for an extensive survey on methods to compute optimal Pareto solutions. In this paper, the specific structure of the objective functions allows us to directly determine their Pareto front.

A. Partitioning

Theorem 1 ([7]). *The expected cost of retrieving the data file under the partitioning (P) strategy is:*

$$C_r^P(k, \lambda) = \left(\frac{k}{\lambda \pi} \right)^\alpha \Gamma(\alpha + 1) . \tag{3}$$

Our first contribution is the following result:

Theorem 2. *The Pareto front of the partitioning strategy is described by the following set of points:*

$$\left\{ (x, y) \mid x > 0, \; y = \frac{\Gamma(\alpha + 1)}{(\pi x)^\alpha} \right\} .$$

Proof. Let $x = C_d(k, \lambda), x > 0$ and $y = C_r^P(k, \lambda)$. Now, using (2) and (3), we can write the expected retrieval cost as a function of the expected deployment cost as follows:

$$y = \frac{\Gamma(\alpha + 1)}{(\pi x)^\alpha},$$

where $x > 0, \alpha \geq 1/2$.

This concludes the proof of Theorem 2. □

The above results demonstrate that the Pareto front of the costs $C_r^P(k, \lambda)$ and $C_d^P(k, \lambda)$ only depends on the ratio of the optimization parameters $x = \lambda/k$.

B. Coding

Theorem 3 ([7]). *The expected cost of retrieving the data file under the coding (C) strategy is:*

$$C_r^C(k, \lambda) = \frac{1}{k} \left(\frac{1}{\lambda \pi} \right)^\alpha \frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)\Gamma(k)} . \tag{4}$$

We firstly state the following lemma:

Lemma 1. $v\psi(v + s) - v\psi(v) - s < 0$, *for* $1 \leq v \leq n$ *and* $s > 1$.

Proof. Let $h(s, t) = \dfrac{(1 - e^{-st})}{(1 - e^{-t})s}$. Then

$$\begin{aligned} v\psi(v + s) - v\psi(v) - s &= v \int_0^\infty \frac{e^{-vt}(1 - e^{-st})}{1 - e^{-t}} \, dt - s \\ &= \int_0^\infty \frac{ve^{-vt}(1 - e^{-st})}{(1 - e^{-t})s} s \, dt - s \\ &= \int_0^\infty h(s, t)(ve^{-vt})s \, dt - s. \end{aligned}$$

First, we show that $h(s, t) < 1$ for $s > 1, t > 0$. This follows from

$$\frac{\partial h(s, t)}{\partial s} = \frac{e^{-st}(1 + ts) - 1}{s^2(1 - e^{-t})} < \frac{e^{-st} \cdot e^{st} - 1}{s^2(1 - e^{-t})} = 0,$$

where, for the last inequality we used $1 + ts < e^{st}$. Thus, $h(s, t)$ is a decreasing function in s and $\lim_{s \to 1} h(s, t) = 1$. Consequently, $h(s, t) < 1$.

Now, since $h(s, t) < 1$ and $\int_0^\infty ve^{-vt} = 1$,

$$v\psi(v + s) - v\psi(v) - s = \int_0^\infty h(s, t) \cdot ve^{-vt} \cdot s \, dt - s < 0.$$

□

Theorem 4. *The Pareto front of the coding strategy is described by the following set of points:*

$$\left\{ (x, y) \mid x > 0, \; y = \frac{\Gamma(\alpha + 1 + n)}{(\alpha + 1)\Gamma(n)(\pi x)^\alpha n^{\alpha + 1}} \right\} .$$

Proof. Let $x = C_d(k, \lambda)$, with $x > 0$, and $y = C_r^C(k, \lambda)$. Then, using Theorem 3, the expected retrieval cost as a function of the expected deployment cost is as follows:

$$y = \frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)(\pi x)^\alpha \Gamma(k) k^{(1+\alpha)}} . \tag{5}$$

Let

$$g(k, \alpha) = \frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)\pi^\alpha \Gamma(k) k^{(1+\alpha)}} .$$

Then $y = g(k, \alpha)x^{-\alpha}$. We now show that $g(k, \alpha)$ is minimized when $k = n$. Computing the gradient of $g(k, \alpha)$ with respect to k gives

$$\frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)\pi^{\alpha}\Gamma(k)k^{(2+\alpha)}}[k\psi(k + 1 + \alpha) - k\psi(k) - 1 - \alpha],$$

which by Lemma 1 is negative. It follows that $g(k, \alpha)$ is decreasing in k. Therefore, y is minimized for $k = n$.

Taking $k = n$ in (5) gives the desired result. □

The above results show that under coding, it is Pareto optimal to always take $k = n$, i.e. to fragment the data as much as possible. Note that this result satisfies the integrality constraint of the storage capacity of the caches.

Theorem 5. *The partioning Pareto front is dominated by the coding Pareto front, i.e., for each point on the partioning Pareto front there is a point on the coding Pareto front that dominates it.*

Proof. We know from Theorem 2 that any point on the partitioning Pareto front can be achieved with $k = n$ and corresponding density λ^*. Let $(C_d(n, \lambda^*), C_r^P(n, \lambda^*))$ be an arbitrary point on the partitioning Pareto front. We show that $(C_d(n, \lambda^*), C_r^C(n, \lambda^*))$ is on the coding Pareto front and that this point dominates $(C_d(n, \lambda^*), C_r^P(n, \lambda^*))$. It follows directly from Theorem 4 that $(C_d(n, \lambda^*), C_r^C(n, \lambda^*))$ is Pareto optimal. Let

$$\omega(z, n) = \frac{C_r^P(n, \lambda^*)}{C_r^C(n, \lambda^*)},$$

where $z = \alpha + 1$. It remains to show that $\omega(z, n) > 1$.

It follows from Theorems 1 and 3 that

$$\omega(z, n) = \frac{n^z z\Gamma(z)\Gamma(n)}{\Gamma(z + n)}.$$

Taking the first derivative

$$\frac{\partial\omega(z, n)}{\partial n} = \frac{zn^{z-1}\Gamma(n)\Gamma(z)[n\psi(n) - n\psi(n + z) + z]}{\Gamma(z + n)} > 0\,,$$

where $[n\psi(n) - n\psi(n + z) + z] > 0$ from Lemma 1. Hence, $\omega(z, n)$ is increasing in n and $\lim_{n\to1} \omega(z, n) = 1$. Therefore, $\omega(z, n) > 1$ for any $n > 1$ and any $\lambda^* > 0$. □

This result is a generalization of a result in [7], where the number of caches to be contacted and the deployment density with which the caches are deployed in the plane are fixed.

Figure 3 shows the Pareto front of the deployment and retrieval costs under partitioning and coding. The figure shows that optimizing for one of the objectives necessarily influences the other. For example, a low data retrieval cost for the client can be achieved at the expense of a more dense, yet more expensive, cache deployment.

IV. Conclusions

This paper has provided insight into the Pareto front of the expected deployment cost of the caches in the plane and the expected cost for a client to retrieve a large data file from the caches. The Pareto front shows to what extent one of the objectives can be improved at the expense of the other. This allows to optimize the overall cost of the system.

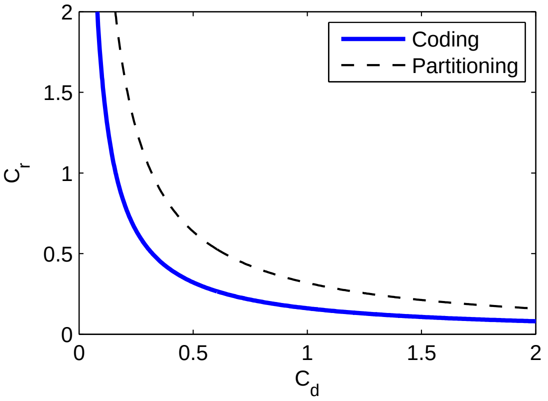

Fig. 3: The Pareto front for coding and partitioning storage strategies, $a = 2$.

For the partitioning strategy, we derived a simple relation for the Pareto points. For the coding strategy, we showed that it is optimal to maximize the data fragmentation. Lastly, we showed that storing data according to the optimal coding strategy results in a lower Pareto front than in the case of the partitioning strategy.

Future research directions include investigating the Pareto front of the expected retrieval and deployment cost for caches prone to failure as well as for storage strategies that consider transmission interferences and energy constraints.

References

[1] S. Jin and L. Wang, “Content and service replication strategies in multi-hop wireless mesh networks,” in *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2005, pp. 79–86.

[2] L. Yin and G. Cao, “Supporting cooperative caching in ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006.

[3] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, “Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes,” in *Proceedings of the International Symposium on Information Processing in Sensor Networks*, 2005, p. 15.

[4] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *arXiv preprint arXiv:1209.5807*, 2012.

[5] U. Niesen, D. Shah, and G. W. Wornell, “Caching in wireless networks,” *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6524–6540, 2012.

[6] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” *Information Theory, IEEE Transactions on*, vol. 56, no. 9, pp. 4539–4551, 2010.

[7] E. Altman, K. Avrachenkov, and J. Goseling, “Coding for caches in the plane,” *arXiv preprint:1309.0604*, 2013.

[8] M. Mitici, J. Goseling, M. de Graaf, and R. J. Boucherie, “Optimal deployment of caches in the plane,” in *Proceedings of the IEEE Global Signal and Information Processing*, 2013.

[9] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[11] G. P. Liu, J. F. Whidborne, and J.-B. Yang, *Multiobjective Optimisation and Control*. Research Studies Press, 2003.