Analysis of Speculative Prefetching

Michael Angermann
michael.angermann@dlr.de
Institute for Communications and Navigation,
German Aerospace Center (DLR),
Oberpfaffenhofen, Germany

Prefetching is a potential method to reduce waiting time for retrieving data over wireless network connections that commonly occur in mobile scenarios. These connections typically exhibit low bandwidth as well as high latency. This article analyzes fundamental principles in speculative prefetching. Based on a simplified system model, the influence of parameters and strategies on prefetching performance is investigated, both analytically and in simulation. The reduction of the mean waiting time is used as performance metric. An example is given that makes use of the Zipf distribution to demonstrate the influence of the documents' probability distribution and its parameters. A low-complexity algorithm that performs optimally under the given assumptions is presented.

I. Introduction

Radio spectrum is a notoriously scarce resource. As a consequence connection speeds over wireless links are and will generally be lower than in the fixed parts of the network by orders of magnitude. Apart from the well established data compression techniques and conventional caching, prefetching of data gains interest, due to its potential to improve the performance of mobile applications. Prefetching relies on the ability to predict a future request and to proactively retrieve the necessary data over the limited network resource *before* the actual request is made, thereby reducing the waiting time for the user. A prefetching algorithm has to *estimate* the potential future requests (*speculative prefetching*). Its performance, in terms of the reduction in waiting time, depends on the degree of randomness of the true requests and the accuracy of the estimation. Previous research on algorithms and architectures for conventional caching [3] has been performed and the obtained results are a valuable information source for the investigation of prefetching. A model for speculative prefetching has been presented by Tuah et al. in [6]. If this model is assumed, an optimum algorithm requires solving a stretch knapsack problem (SKP). The model presented here is a variation of Tuah's and results in an algorithm with low computational complexity for achieving the theoretical optimum performance. In the work presented here some underlying principles that arise in prefetching are investigated. Our aim is to gain fundamental insight in these principles which will help to interpret and understand the results obtained in trace-driven simulations and real systems.

II. System Model

Predominantly traces of application-layer traffic, mostly HTTP-traffic, have been used to carry out performance analysis and comparisons between distinct prefetching algorithms and combinations of prefetching and caching [2][4][5]. Simulations based on real traces certainly are capable to achieve more realistic results than model-based analysis. Yet, it is difficult to isolate the effect of individual parameters that resulted in the particular used trace. As the character of network-traffic changes in dramatic speed and extend, it is necessary to analyze the performance of algorithms and strategies based on models, whose parameters can be arbitarily chosen to resemble potential future circumstances. We therefore introduce a probabilistic model for our analysis. The model is comprised of an entity (*client*) that acts as a discrete random source issuing requests, and other entities (*servers*) that deliver their responses over connections with finite bandwidth. We assume that the connections' bandwidth is exclusively dedicated to the transport of responses to this particular client. This models e.g. a scenario with a mobile device using a fixed bandwidth wireless link to connect to the conglomerate of servers in the internet, with the wireless link as the dominant bottleneck and the responses representing the dominant data volume. We assume that in a certain situation the client requests a document. The requested document will be randomly chosen out of an ordered set of N documents $d_i,\ i = 1 \ldots N$, with probability $P\{d_i\} = p_i$, size V_i and connection speed C_i for the retrieval of this document. The document will be requested at previously unknown time t_R, drawn randomly from a continuous probability density function $f_{t_R}(t_R)$. The connections are available for the prefetching attempt at $t = t_0$. The time

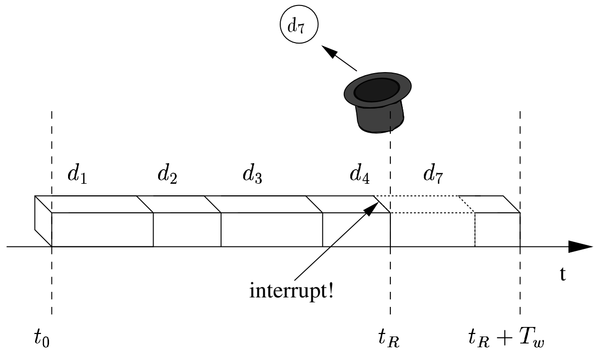

Figure 1: Document d_7 is chosen at request time t_R

between the request generation and the moment the corresponding document is completely retrieved is called *waiting time* T_w. If a new request is generated during a speculative retrieval for another document, the speculative retrieval is immediately interrupted, whereas in [6] the speculative prefetch completes before the actually requested document is retrieved.

III. Analysis

In our analysis we will answer the following general questions: What criterion should we apply to determine the sequence of speculatively retrieved documents? Do we prefer small documents over large documents? How does the distribution of probabilities influence the achievable reduction in waiting time T_w?

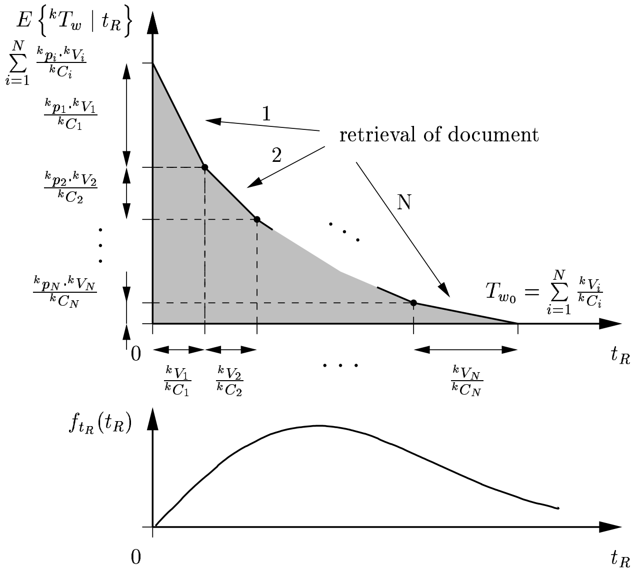

Figure 2: Expected value for waiting time $E\{T_w \mid t_R\}$ and arbitrary probability density function $f_{t_R}(t_R)$ of the request time t_R.

The expected waiting time without any prefetching is

$$E\{T_w\}_{\textit{no prefetch}} = \sum_{i=1}^{N} p_i \cdot \frac{V_i}{C_i}. \quad (1)$$

We can express any sequence of the N possible documents d_i as a permutation of the initial ordered set. The number of possible permutation of an N-element set is $N!$. We write ${}^k element_i$ and ${}^k value$ to denote the i-th element of a set or a value calculated for a particular permutation k. The index k is used to denote the position of the particular permutation among any defined order (e.g. lexicographic) of all possible permutations. Hence, ${}^k p_i$ and ${}^k V_i$ are the i-th elements of the ordered set of the documents' probabilities and sizes *after* these sets have been ordered according to the k-th permutation with permutations numbered in the specified order.

We calculate $E\{{}^kT_w \mid t_R\}$, the expected value for T_w for a chosen permutation k, under the condition that the request arrives at a certain time t_R

$$E\{{}^kT_w \mid t_R\} = E\{T_w\}_{\textit{no prefetch}} - E\{{}^kT_{\textit{gain}} \mid t_R\}, \quad (2)$$

with

$$E\{{}^kT_{\textit{gain}} \mid t_R\} = \sum_{i=1}^{N} \left({}^kp_i \cdot \min\left\{ \max\left\{0, t_R - t_0 - \left(-\frac{{}^kV_i}{{}^kC_i} + \sum_{j=1}^{i} \frac{{}^kV_j}{{}^kC_j} \right) \right\}, \frac{{}^kV_i}{{}^kC_i} \right\} \right). \quad (3)$$

We try to give an intuitive explanation for Eqn. 3: The total time available for prefetching is $t_R - t_0$. It is necessary to distinguish between three distinct cases: a) no attempt has been made to retrieve the chosen document before t_R. In this case no time is gained, but the retrieval starts immediately at t_R. Hence, the minimum time gain is 0. b) the chosen document is currently being retrieved. Subtracting the time invested on the other documents from the total time available for prefetching yields the time gained. c) the document has already been completely retrieved. Then the complete time $\frac{{}^kV_i}{{}^kC_i}$ necessary to retrieve it is gained, but not more. The three cases are represented by the min / max operation. Finally the expected value is calculated by weighting with the documents' probabilities ${}^kp_i$ and summing up over all $i = 1 \ldots N$.

Eqn. 3 is illustrated in Fig. 2, with $t_0 = 0$, for one arbitray permutation k. The upper curve shows the expected value for the waiting time. It is composed of N linear segments. Each segment corresponds with the retrieval of one particular document. For requests arriving as early as $t_R = 0$, no improvement is possible yet. The expected value for the waiting time is then

$$E\{{}^kT_w \mid t_R = 0\} = E\{T_w\}_{\textit{no prefetch}}, \quad (4)$$

which is the same for all permutations. Independently of the chosen permutation the expected value for the waiting time reaches zero when all documents have been prefetched after

$$T_{w_0} = \sum_{i=1}^{N} \frac{{}^kV_i}{C_i}. \quad (5)$$

As t_R is distributed according to $f_{t_R}(t_R)$ the expected value of the unconditioned waiting time is

$$E\{T_w\} = \int_0^\infty f_{t_R}(t_R) \cdot E\{{}^kT_w \mid t_R\}\, dt_R. \quad (6)$$

In order to minimize $E\{T_w\}$ it is sufficient to choose the permutation that minimizes the gray-shaded area in Fig. 2. This can be proved, using the facts that $E\{{}^kT_w \mid t_R\}$ is monotonically decreasing for all permutations k, and that $f_{t_R}(t_R) \geq 0 \; \forall \; t_R$. With this in mind and the observation that the slope of every segment depends only on the probability p_i of its document we can derive a simple two-step algorithm:

1. sort the documents with respect to their probability p_i.
2. sequentially fetch all documents.

It is important to notice that the optimum sequence only depends on the documents' probabilities *not* on their size or connection speed. With this result we are now able to answer two of the three questions we asked when beginning our analysis: The probabilities p_i should be applied as the criterion for determining the sequence. In order to minimize waiting time, it is not necessary or beneficial to prefer small documents over large documents.

III.A. Example

To acquire an understanding for the influence of the probabilities and permutations we analyze an example. The parameters in the example shall be $N = 6$, $C_i, V_i = 1 \ \forall\, i = 1 \ldots N$. The probabilities p_i shall be distributed according to Zipf's Law[1] [1], with $\alpha = 1$ ($p_1 \approx 0.408$, $p_2 \approx 0.2041$, $p_3 \approx 0.1360$, $p_4 \approx 0.1020$, $p_5 \approx 0.0816$, $p_6 \approx 0.0680$).
Fig. 3 shows the resulting curves for all $6! = 720$ permutations, each representing a possible prefetching decision.

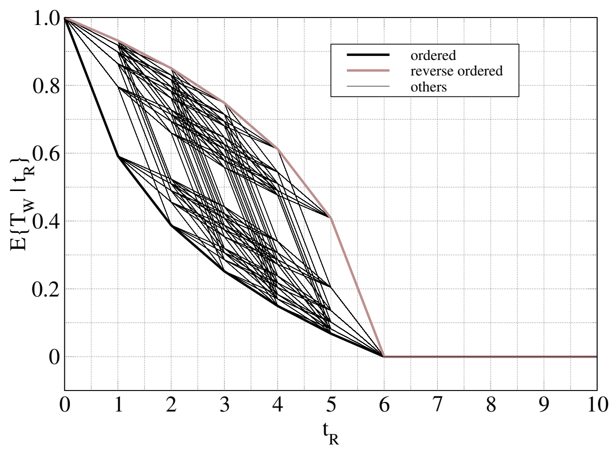

Figure 3: Influence of permutations ($C_1 \ldots C_6 = 1$, $V_1 \ldots V_6 = 1$, Zipf Distribution, $\alpha = 1$).

Common sense suggests that the more pronounced the differences in the documents' probabilities p_i are, the better the performance of pefetching can get. The distribution according to Zipf's Law facilitates a quantitative analysis of this conjecture. By varying α we can adjust the distribution from being absolutely flat ($\alpha = 0$) to a stronger pronouncing of the likelier documents when α rises to 1.0 (or higher). This effect is depicted in Fig. 4. Decreasing α reduces the distance between best and worst strategy. Fig. 5 shows the ordered (best) and reverse ordered (worst) strategies. We can see that for the case $\alpha = 0$, when all N documents have the same probability $p_i = 1/N$, no difference between the best and the worst strategy exists.

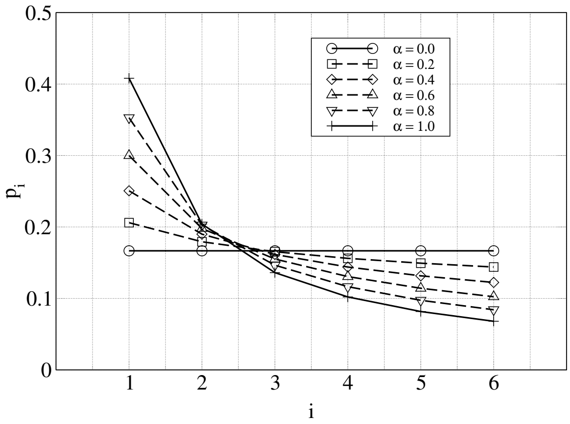

Figure 4: Influence of α, Zipf Distribution ($N = 6$).

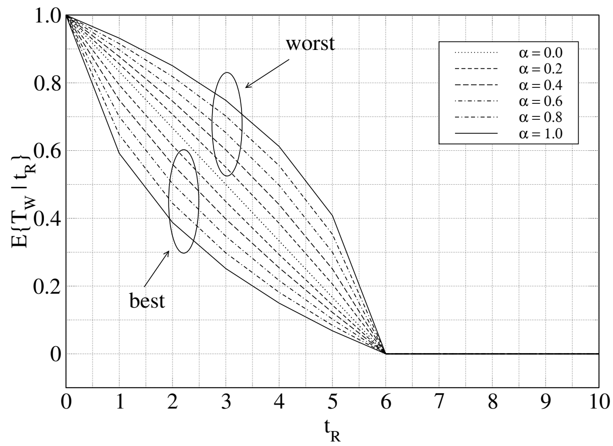

Figure 5: Influence of α ($C_1 \ldots C_6 = 1$, $V_1 \ldots V_6 = 1$, Zipf Distribution).

IV. Simulation

To augment and verify the analysis Monte-Carlo simulations are performed. The parameters (p_i, C_i, V_i) are chosen to match the example. In the first simulation the *prefetching controller* (PC) has a-priori knowledge of the documents' probabilities p_i. The actual waiting times $T_{w_j}, j = 1 \ldots M$ are measured for $M = 1000$ trials. The mean value $\overline{T}_w = \frac{1}{M} \sum_{j=1}^{M} T_{w_j}$ is plotted in Fig. 6 for several t_R. Simulation results and theoretical analysis show good consensus.
In the second simulation t_R is uniformly distributed from $[\,0, T_{w_0}]$ (This distribution simplifies interpretation of results. Other distributions e.g Poisson, Pareto are used for more realistic scenarios). Furthermore, the PC has *no* a-priori knowledge of p_i and uses the relative frequency of observed documents as estimations of p_i in this simulation. The learning behaviour and the influence of α are shown in Fig. 7. According to the theoretical analysis performance

[1] Zipf's Law states that the probability of the i-th most likely event is proportional to $1/i$. This is also called the *strict* Zipf's Law. Many interesting experiments show a slight modification of this law. They can be more adequately modelled with a probability proportional to $1/i^\alpha$ for the i-th most likely event. The value α then typically takes a value of less than unity. For $\alpha = 1$ this modified law is equivalent to the strict Zipf's Law.

of the PC should increase with the value of α. The PC exploits the asymmetries in the documents' probabilities. We can see for the particular setup of parameters the reduction of $\overline{T}_w$. It takes approximately 10 (20, 30) visits for $\alpha = 0.0$ ($\alpha = 0.5$, $\alpha = 1.0$) for the PC to gather enough observations for a good estimation of the probabilities p_i that is necessary to reach its asymptotic performance. With these observations and the results of the analysis depicted in Fig. 5 we have answered the third question on the influence of the distribution of probabilities on the achievable performance.

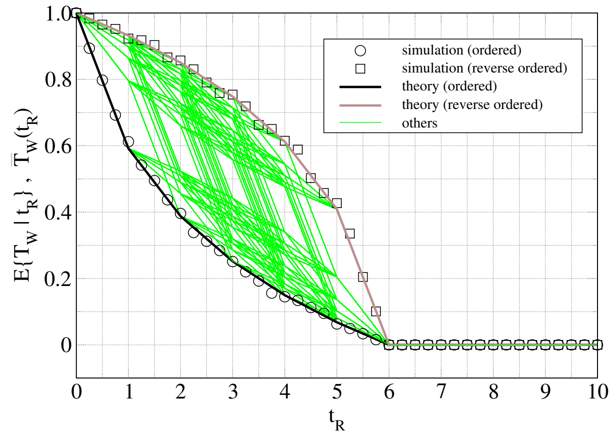

Figure 6: Influence of permutations, simulation result ($C_1 \ldots C_6 = 1, V_1 \ldots V_6 = 1$, Zipf Distribution, $\alpha = 1$).

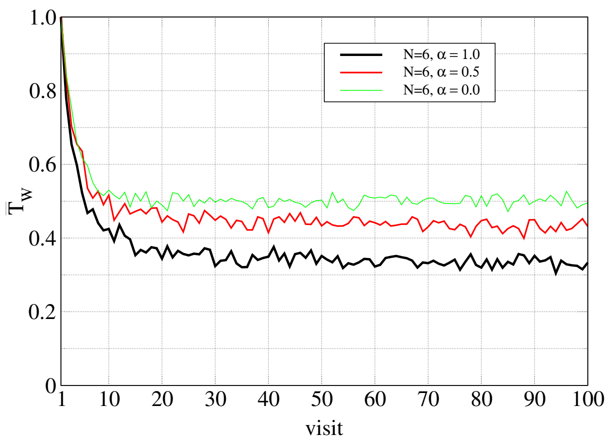

Figure 7: Influence of α, simulation result ($C_1 \ldots C_6 = 1$, $V_1 \ldots V_6 = 1$, Zipf Distribution), t_R equally distributed .

V. Conclusions and Further Work

The mean waiting time is minimized by prefetching with an algorithm with low computational complexity. The performance of a PC with a-priori knowledge of the probabilities p_i, that follows the proposed algorithm, constitutes an upper bound on the achievable reduction in waiting time. A PC's strategy, to use the relative frequencies of occurrance as estimates for the documents' probabilities, is optimal in the maximum-likelihood sense.

For a typical of today's mobile scenarios, i.e. WWW-browsing over a circuit-switched and air-time charged personal communication system the proposed model and algorithm is resembling reality fairly well.

Interesting extensions to the current model are time-varying document probabilities $p_i(t)$ and individual probability density functions $f_{t_R}(t_R \mid d_i)$ of the request time t_R for each document d_i.

It is also desirable to enhance the model to allow the analysis of other scenarios where the bottleneck connection is used by a *multitude of users simultaneously*. For minimizing the global waiting in this broader scenario it is sometimes necessary to refrain from prefetching documents with lower probabilities p_i. We further intend to improve the PC to obtain and exploit additional information (soft information) on the reliability of its estimation. This problem can be treated by Bayesian inference.

Based on the extensive work on architectures and protocols (for an overview and excellent starting point see [3]) the implementation and deployment of efficient systems remains the utmost objective.

Acknowledgements

The author would like to thank the editor and the reviewers for their valuable insights, comments and encouragement. Their competence and dedication cannot be esteemed highly enough.

References

[1] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications, *Proc. of the IEEE Infocom '99*, New York, NY, March 1999.

[2] Mark Crovella and Paul Barford. The network effects of prefetching, *Proc. of the IEEE Infocom '98*, San Francisco, CA, March/April 1998.

[3] B. D. Davison. A web caching primer, *IEEE Internet Computing*, 5(4), July/August 2001.

[4] Li Fan, Pei Cao, Wei Lin, and Quinn Jacobson. Web prefetching between low-bandwidth clients and proxies: Potential and performance, *Proc. of SIGMETRICS '99*, Atlanta, GA, May 1999.

[5] Thomas M. Kroeger and Darrell D.E. Long. Exploring the bounds of web latency reduction from caching and prefetching, *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Monterey, California, December 1997.

[6] N. J. Tuah, M. Kumar and S. Venkatesh. A performance model of speculative prefetching in distributed information systems, *Intl. Parallel Processing Symposium*, San Juan, Puerto Rico, April 1999.

[7] A. Venkataramani, P. Yalagandula, R. Kokku, S. Sharif, and M. Dahlin. The potential costs and benefits of long term prefetching for content distribution, Technical Report TR-01-13, Department of Computer Science, University of Texas at Austin, June 2001.