Technical Report 1999-12-03
Internetworking and Media Communications Research Laboratories
Department of Math & Computer Science, Kent State University
http://medianet.kent.edu/technicalreports.html

ORDERING PREFETCH IN TREES, SEQUENCES AND GRAPHS

Javed I. Khan

Technical Report 1999-12-03

Internetworking and Media Communications Research Laboratories
Department of Math & Computer Science
Kent State University, 233 MSB, Kent, OH 44242
December 1999

javed@kent.edu

ABSTRACT

Compared to hardware prefetching, the prefetching in Web systems faces quite high branching factor. Decision points mostly bifurcate the control flow tree in hardware due to the predominant if-then like program constructs. In contrast, in web there is no limit on the number of links in a page. In the case of hardware quite often all the parallel branches are prefetched- and in some cases condition can be pre-evaluated to determine the prefetch path. Neither is practical for web systems. There is critical for web systems to carefully evaluate all prefetch options. This report contains few analytical results, which show how to rank prefetch paths in various hyper linked graphs shown in Fig-A-D. It seems in most cases optimum prefetch path should depend both on the link transition probability as well as loading time of the component –rather than just the former.

1 Optimization Criterion

In a hyper-graph G, Lets $U=(a_1,a_2,a_3\ldots a_l)$, where $u_i \in G$, is the *anchor sequence*-- the sequence of nodes followed by a user. Let's Γ is the *loading sequence* in which the nodes are loaded in the cache (Clearly, $U \subseteq \Gamma \subseteq$ {nodes in G}). Let p_i is the estimated probability that a user traverses a node n_i in roaming sphere G, and $T_{L,i}$ is the time the node a_i is fetched and $T_{P,i}$ is the time spent by the user in that node. Thus, we define an overall penalty function-- the expected cumulative *read-time lag*:

$$T(\Gamma \mid U) = \sum_{i}^{U} p_i \max\left\{[T_{L,i} - (T_{L,i-1} + T_{P,i-1})],0\right\} \qquad ...(1)$$

The objective is to find the loading sequence Γ that will minimize the expected penalty $E\{T(\Gamma|U)\}$. It is important to note that this function optimizes with respect to all probable transitions of U, weighted by their transition probability.

2 Some Analytical Results

Theorem-1 (Branch Decision)*: Let $A=n_c$ is the current anchor point with direct transition paths to a set of candidate nodes n_1, n_2, $n_{3,--}$ n_n, such that T_i is the estimated loading times of node n_i, and $Pr[a_{n+1}=n_i \mid a_n=A]$ is the conditional link transition probability, then the average delay is minimum if the links are prefetched in-order of the highest priority Q_i, where:*

$$Q_i = \frac{\Pr[a_{n+1} = n_i \mid a_n = A]}{T_i} \qquad .....(A.1)$$

Proof: Let us consider two pre-fetching sequences Seq1= [n_o, n_1, $n_2 \ldots n_{m-1}$, $\mathbf{n_m}$, $n_{m+1}, \ldots n_{r-1}$, $\mathbf{n_r}$, $n_{r+1} \ldots, n_N$] and Seq2=[n_o, n_1, $n_2 \ldots n_{m-1}$, $\mathbf{n_r}$, $n_{m+1}, \ldots n_{r-1}$, $\mathbf{n_m}$, $n_{r+1} \ldots, n_N$], where m<r. These two sequences are identical, except only two of the nodes n_m and n_r have been switched their positions. We also note that the nodes loaded prior to the m^{th} node and the nodes loaded after the r^{th} nodes are identical in both the sequences. We use the following expressions to denote the cost function due to these two nodes:

$$C_{prev.} = \sum_{j=0}^{m-1} \left\{ p_j \sum_{i=0}^{j} T_i \right\} \text{ , and } C_{post.} = \sum_{j=r+1}^{N} \left\{ p_j \sum_{i=0}^{j} T_i \right\},$$

We also use $T_p = \sum_{i=0}^{m-1} T_i$ to denote cumulative load time of nodes loaded before $\mathbf{n_m}$. Let 's also denote $p_i = \Pr[a_{n+1} = n_i \mid a_n = A]$. Thus, the expected cost factor for sequence 1 is given by:

$$\begin{aligned} C_{seq1} &= C_{prev} + p_m \cdot (T_p + T_m) + p_{m+1} \cdot (T_p + T_m + T_{m+1}) ..... \qquad .....(A.2) \\ &+ p_{r-1} \cdot (T_p + T_m + T_{m+1} ... + T_{r-1}) + p_r \cdot (T_p + T_m + T_{m+1} ... + T_{r-1} + T_r) + C_{post} \\ &= C_{prev} + p_m \cdot (T_p + T_m) + \sum_{j=m+1}^{r-1} \left\{ p_j \cdot (T_p + T_m + \sum_{i=m+1}^{j} T_i) \right\} + p_r \cdot (T_p + T_m ... + T_r) + C_{post} \\ &= T_m \cdot \left\{ \sum_{j=m+1}^{r-1} p_j \right\} + p_r \cdot (T_m ... + T_{r-1}) + K_1 \end{aligned}$$

Where:

$$K_1 = C_{prev} + p_m \cdot (T_p + T_m) + \sum_{j=m+1}^{r-1} \left\{ p_j \cdot (T_p + \sum_{i=m+1}^{j} T_i) \right\} + p_r \cdot (T_p + T_r) + C_{post}$$

In a similar way, the cost for the sequence 2 is given by:

$$\begin{aligned} C_{seq2} &= C_{prev} + p_r \cdot (T_p + T_r) + p_{m+1} \cdot (T_p + T_r + T_{m+1}) ..... \qquad .....(A.3) \\ &+ p_{r-1} \cdot (T_p + T_r + T_{m+1} .... + T_{r-1}) + p_m \cdot (T_p + T_r + T_{m+1} ... + T_{r-1} + T_m) + C_{post} \\ &= C_{prev} + p_r \cdot (T_p + T_r) + \sum_{j=m+1}^{r-1} \left\{ p_j \cdot (T_p + T_r + \sum_{i=m+1}^{j} T_i) \right\} + p_m \cdot (T_p + T_m ... + T_r) + C_{post} \\ &== T_r \cdot \left\{ \sum_{j=m+1}^{r-1} p_j \right\} + p_m \cdot (T_{m+1} ... + T_r) + K_2 \end{aligned}$$

Where,

$$K_2 = C_{prev} + p_r \cdot (T_p + T_r) + \sum_{j=m+1}^{r-1} \left\{ p_j \cdot (T_p + \sum_{i=m+1}^{j} T_i) \right\} + p_m \cdot (T_p + T_m) + C_{post}$$

We now compare the estimated delay costs of these sequences, and show that under the given condition (A.1) one will be always lesser than the other. Both (A.2) and (A.3) can be expanded as following:

$$C_{seq1} = T_m \cdot (p_{m+1} + ... + p_{r-1}) + p_r \cdot (T_m ... + T_{r-1}) + K_1$$

$$C_{seq2} = T_r \cdot (p_{m+1} + ... + p_{r-1}) + p_m \cdot (T_{m+1} ... + T_r) + K_2$$

Rearrangement of the terms will show that $K_1 = K_2$. Additionally, if:

$$\frac{p_m}{T_m} \geq \frac{p_{m+i}}{T_{m+i}} \qquad .....(A.4)$$

Then $T_m \cdot p_{m+i} \leq p_m \cdot T_{m+i}$, and since r>m, $p_r \cdot T_{r-i} \cdot \leq T_r \cdot p_{r-i}$, for all positive i. Thus, the left terms of (A.2) are smaller than the middle terms of (A.3). Similarly, the middle terms of (A.2) are smaller than the left terms of (A.3). Thus, $T_m \cdot (p_{m+1} + ... + p_{r-1}) \leq p_m \cdot (T_{m+1} ... + T_r)$ and $p_r \cdot (T_m ... + T_{r-1}) \leq T_r \cdot (p_{m+1} + ... + p_{r-1})$. Thus, $C_{seq1} \leq C_{seq2}$ is always true under condition A.4, which is the priority given is equation (A.1) of individual nodes (proved).

<u>Corollary-2.1 (Sequence Decision):</u> *If two nodes are in a sequence, then the preceding node has to be loaded first.*

Proof: below we provide a direct proof of this intuitive corollary. Let nodes n_1 and n_2 are in a sequence, n1 preceding n2. We compare the costs for two sequences Seq1= [.. n_1, n_2…] Seq2= […, n_2, n_1…]. Since, n_2 is traversed only after n_2 the state probabilities $p_2 \leq p_1$. The response delay count begins for the second node, after the first node is loaded and then read i.e. at time (T_1+R_1) . Thus, the cost of first sequence is given by:

$$C_{seq1} = p_1 \cdot T_1 + p_2 \cdot \max\{T_1 + T_2 - (T_1 + R_1), 0\} = p_1 \cdot T_1 + p_2 \cdot \max\{T_2 - R_1), 0\} \qquad .....(B.1)$$

Similarly, the cost for the second sequence is given by:

$$C_{seq2} = p_1 \cdot (T_1 + T_2) + p_2 \cdot \max\{T_1 - (T_1 + T_2 + R_1), 0\} \qquad .....(B.2)$$

$$= p_1 \cdot (T_1 + T_2)$$

Since, the second node is loaded before the first node so it will be immediately available after the first node is read. If $\max\{T_2 - R_1), 0\} = 0$ then, $0 \leq p_1 \cdot (T_1 + T_2)$. Thus, $C_{seq1} \leq C_{seq2}$. On the other hand, if, $\max\{T_2 - R_1), 0\} = T_2 - R_1$,then also, $p_1 \cdot T_1 + p_2 \cdot (T_2 - R_1) \leq p_1 \cdot (T_1 + T_2)$, or $p_2 \cdot (T_2 - R_1) \leq p_1 \cdot T_2$. Since, $p_2 \leq p_1$ and $T_2 - R_2 \leq T_2$. Thus, for this case also $C_{seq1} \leq C_{seq2}$. Note: If node 2 can be reached via a second path, then state probability p2<=p1 may not be true (proved).

Theorem-2 (Tree Decision)*: If the sequence $\{n_1, n_2, n_3,...n_d\}$ are the nodes in the path in a tree from the current anchor $A=n_0$ to a candidate node n_d, at depth d and T_d is it's estimated loading time, then the priority Q_d is given by the product of the conditional transition probabilities along the path such that:*

$$Q_d = \frac{\prod_{i=c}^{d} \Pr[a_{i+1} = n_{i+1} \mid a_i = n_i]}{T_d} \quad \text{.....(C.1)}$$

Proof: Without loss of generality, the question we will address in this proof is that should we bring a node from depth d before fetching a node from depth 1? Consequently, we compare two pre-fetching sequences Seq 1 = […. n_1, $\mathbf{n_2}$, $\mathbf{n_3}$,….] and Seq 2= […. n_1, $\mathbf{n_3}$, $\mathbf{n_2}$,..]. Here node 1 is the highest priority node at depth 1, and $\mathbf{n_2}$ is one of its lower priority sibling in the same depth and thus yet not fetched, and $\mathbf{n_3}$ is the new node at depth d exposed by the fetching of node 1. For removing some clatter in this proof, without loosing the generality of proof, we also this time assume that the set of nodes between these switched nodes is empty. Lets also assume that $p_{i=2} = \Pr[a_{n+1} = n_2 \mid a_n = A]$ and $p_3 = \prod_{i=c}^{d} \Pr[a_{i+1} = n_{i+1} \mid a_i = n_i]$.

Thus, the cost corresponding to these sequences are respectively given by the following expressions.

$$C_{seq1} = p_1 \cdot T_1 + p_2 \cdot (T_1 + T_2) + p_3 \cdot \max\{T_1 + T_2 + T_3 - (T_1 + R_1), 0\} \quad \text{.....(C.2)}$$

$$C_{seq2} = p_1 \cdot T_1 + p_2 \cdot (T_1 + T_2 + T_3) + p_3 \cdot \max\{T_1 + T_3 - (T_1 + R_1), 0\} \quad \text{.....(C.3)}$$

In both of the cases, we begun counting the response delay for n_3 after n_1 is loaded and read. The proof that, $C_{seq1} \le C_{seq2}$ requires that we show under condition (1b):

$$p_3 \cdot \max\{T_2 + T_3 - R_1, 0\} \le p_2 \cdot T_3 + p_3 \cdot \max\{T_3 - R_1, 0\} \quad \text{.....(C.4)}$$

Below, we show it by considering the following three cases.

Case-1: Let us consider that case, when, $\max\{T_2 + T_3 - R_1, 0\} = T_2 + T_3 - R_1$, and $\max\{T_3 - R_1, 0\} = T_3 - R_1$. Then the above reduces that we show that:

$p_3(T_2 + T_3 - R_1) \le p_2 \cdot T_3 + p_3 \cdot (T_3 - R_1)$, i.e. $p_3 \cdot T_2 \le p_2 \cdot T_3$, which is the case if $\frac{p_2}{T_2} \ge \frac{p_3}{T_3}$

Case 2: Let us consider that $\max\{T_2 + T_3 - R_1, 0\} = 0$. Then we are required to show that $p_3 \cdot \max\{T_2 + T_3 - R_1, 0\} = 0 \le p_2 \cdot T_3 + p_3 \cdot \max\{T_3 - R_1, 0\}$, which is always the case.

Case 3: Let us consider the only remaining case where, $\max\{T_2 + T_3 - R_1, 0\} = T_2 + T_3 - R_1$, but $\max\{T_3 - R_1, 0\} = 0$. Then, $T_3 - R_1 \le 0$ and we require to show that, $p_3 \cdot (T_2 + T_3 - R_1) \le p_2 \cdot T_3$.

Since, $T_3 - R_1 \leq 0$, then to show $p_3 \cdot (T_2 + T_3 - R_1) \leq p_2 \cdot T_3$, thus it should be sufficient to show that $p_3 \cdot T_2 \leq p_2 \cdot T_3$. Which is the case if $\frac{p_2}{T_2} \geq \frac{p_3}{T_3}$.Thus, for all situations $C_{seq1} \leq C_{seq2}$ (proved).

<u>Theorem-3 (Graph Decision)</u>*: For general graph $G(V_G,E_G)$ the node priority can be determined by computing order-n Markov state probability p_i. For a node n_i, with the estimated loading time T_i the priority function can be computed as:*

$$Q_i = \frac{p_i}{T_i} \quad \text{.....(D.1)}$$

Proof: Let us consider that a general graph, with link transition probabilities p_{ij} and state probabilities p_n. Let the current anchor node is n_0. We again consider two nodes, both un-fetched, but one n_2 at depth 1 from current anchor that can be called immediately, and another n_3 somewhere deep in the graph, and which can be accessed via multiple paths and at least one preceding node of which have been fetched. Since, n_3 is now in the pre-fetch set, a root of n_3 must also be in the pre-fetch set with higher priority than n_2. Let this node be n_1. Let L (0,3) = {l_1, l_2, l_3 .. l_n} is the set of all the paths through which n_3 can be reached from n_0. Consequently, we consider two pre-fetching sequences Sequence 1 = […. n_1, **n_2, n_3**,….] and Sequence 2= […. n_1, **n_3, n_2**,..]. Also, for reducing some clatter in this proof, without loosing the generality of proof, we also this time assume that the set of nodes between these switched nodes is empty. Let $\Pr(l=l_i \mid ao=n_o)$ is the probability of traversing path 0-2 via path l_i. Let $R(l_i)$ is the cumulative rendering time of all nodes in path l_i except n3. Then the cost of sequence 1 and sequence 2 are respectively given by:

$$C_{seq1} = p_2 \cdot (T_1 + T_2) + \sum_{l}^{L} \Pr[l = l_{03}^i \mid A_n = n_0] \cdot \max\{T_1 + T_2 + T_3 - R(l_{03}^i), 0\} \quad \text{.....(D.2)}$$

$$C_{seq2} = p_2 \cdot (T_1 + T_2 + T_3) + \sum_{l}^{L} \{\Pr[l = l_{03}^i \mid A_n = n_0] \cdot \max[T_1 + T_3 - R(l_{03}^i), 0]\}. \quad \text{.....(D.3)}$$

Note, there are other nodes involved along each path, each of which must be loaded and read before n_3 is reached. In a more precise sense, only after the current node has been requested by the user then the delay counter for the current node should begin. Thus D.3 overestimates the penalty by not issuing credit for the time needed in loading the preceding nodes along the path, which may or may not be already in the cache. (However, credit is issued for reading times). For fairness, however, we have taken out this credit in both the sequences. Now we consider the following cases:

Case A:
Now, if $\max\{T_1 + T_2 + T_3 - R(l_{03}^i), 0\} = 0$, then also true is $\max\{T_1 + T_3 - R(l_{03}^i), 0\} = 0$. Thus, $C_{seq1} \leq C_{seq2}$ always.

Case B:

On the other hand, if $\max\{T_1 + T_3 - R(l_{03}^i), 0\} = 0$ then $\max\{T_1 + T_2 + T_3 - R(l_{03}^i), 0\} \le |T_2|$. Thus,

$$C_{seq1} \le p_2 \cdot (T_1 + T_2) + \sum_l^L \Pr[l = l_{03}^i \mid A_n = n_0] \cdot |T_2| = p_2 \cdot (T_1 + T_2) + p_3 \cdot |T_2| \cdot$$

$C_{seq2} = p_2 \cdot (T_1 + T_2 + T_3) \cdot$ Thus when, $p_3 \cdot T_2 \le p_2 \cdot T_3$, then also $C_{seq1} \le C_{seq2}$.

Case C: otherwise:

$$C_{seq1} = p_2 \cdot (T_1 + T_2) + p_3 \cdot (T_1 + T_2 + T_3) - \sum_l^L \Pr[l = l_{03}^i \mid A_n = n_0] \cdot R(l_{03}^i)$$

$$C_{seq2} = p_2 \cdot (T_1 + T_2 + T_3) + p_3 \cdot (T_1 + T_3) - \sum_l^L \Pr[l = l_{03}^i \mid A_n = n_0] \cdot R(l_{03}^i)$$

Thus, when, $p_3 \cdot T_2 \le p_2 \cdot T_3$, then also $C_{seq1} \le C_{seq2}$ (proved).

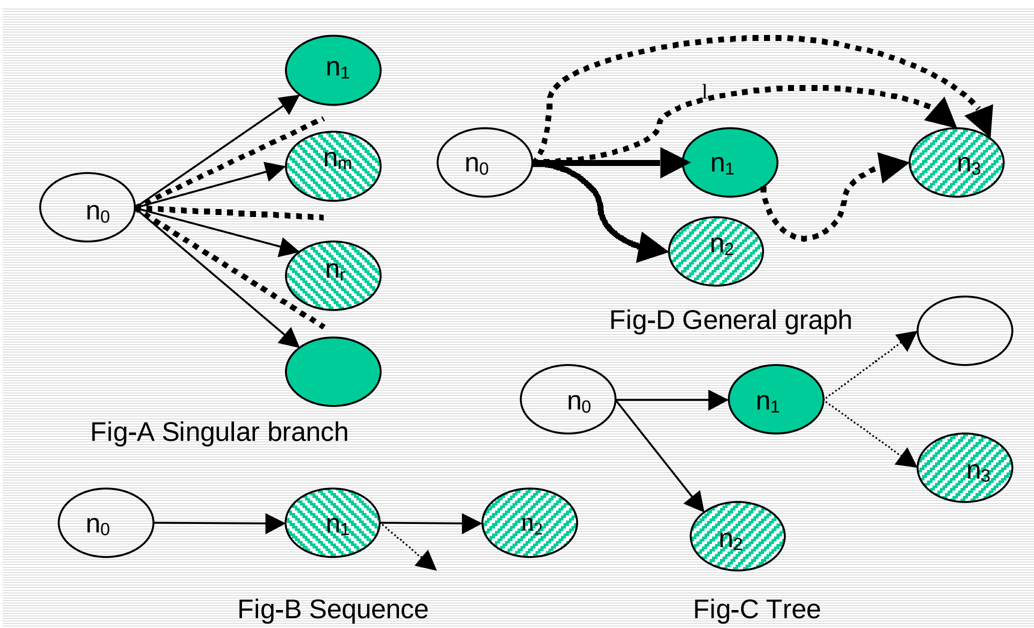

Fig-A Singular branch
Fig-B Sequence
Fig-C Tree
Fig-D General graph