Parallel Optimization of Signal Detection in Active Magnetospheric Signal Injection Experiments: Supplementary Material

Michael Gowanlock[1,2], Justin D. Li[2], Cody M. Rude[2], Victor Pankratius[2]

[1] *School of Informatics, Computing & Cyber Systems, Northern Arizona University, Flagstaff, AZ, U.S.A.*
michael.gowanlock@nau.edu
[2] *Massachusetts Institute of Technology, Haystack Observatory, Westford, MA, U.S.A.*
[jdli, cmrude, pankrat]@mit.edu

1. Performance Modeling

We develop performance models for the breadth-first and depth-first searches. The performance models are generalized, such that the response time can be estimated for pipelines of any size with given pruning requirements. This yields estimated response times that can be used when assessing whether the performance of a pipeline is sufficient for a given use case (e.g., a real-time space weather monitoring system).

1.1. Breadth-First Model

A performance model for the breadth-first approach for a pipeline, P with n stages is as follows. Let T_i be the time to execute an average parameter value at stage S_i, where $i = 1, \dots, n$. We adopt the former notation regarding the number of parameters at each stage, $|S_i|$, and the fraction of filtering between two stages is given as $f_{S_i S_j}$. The modeled execution time of the BFS becomes:

$$T^{BFS}(T, S) = |S_1|\, T_1 + |S_1| \sum_{i=2}^{n} T_i \prod_{k=0}^{i-2} f_{S_{i-k-1} S_{i-k}}\, |S_{i-k}|. \tag{1}$$

As an example, when $n = 2$ the modeled response time is $T^{BFS}(T, S) = |S_1|T_1 + |S_1||S_2|f_{S_1 S_2} T_2$.

Table 1 shows the modeled vs. measured response time for the BFS for four values of $f_{S_2 S_3}$ and $f_{S_3 S_4}$ on *SipleSignal* using scenario $E1$ (here, $f_{S_1 S_2} = 1$). These correspond to the corners of the heatmap in Figure 13. An approximate measurement of T_i can be calculated from Figure 12 (the response time divided by the total number of variants at each stage). In the sequential search ($t = 1$), the largest difference between the model and measured response times is 3.5%. In the parallel search ($t = 16$) the largest difference is 5%. Therefore, the performance model exhibits good accuracy.

1.2. Depth-First Model

The DFS as described in Section 5.6 uses a schedule of randomly selected variants to show the robustness of the pruning heuristic. As such, the chosen stages that are executed varies for each of the executions shown

Table 1: Breadth-first performance model. t denotes the number of threads. The estimated vs. measured response times are given and the percentage difference.

$f_{S_2S_3}$	$f_{S_3S_4}$	t	Model (s)	Measured (s)	Diff. (%)
0.1	0.1	1	94.47	97.07	2.71
1.0	0.1	1	98.41	101.91	3.50
0.1	1.0	1	99.90	97.24	2.69
1.0	1.0	1	103.83	107.27	3.25
0.1	0.1	16	7.96	7.91	0.55
1.0	0.1	16	8.30	8.16	1.65
0.1	1.0	16	8.39	7.98	5.01
1.0	1.0	16	8.74	8.63	1.16

Table 2: Parameters values of T_i for the Depth-First model. Times are measured for the execution of the exhaustive search of the individual stages using 1 thread (without overheads). This is scaled to 16 threads using the response time ratio of 1 to 16 threads in Table 1, $E1$. The overheads are included (the total time matches Table 1). T_i obtained by dividing by the number of parameters at each stage.

Stage (i)	Stage Total 1 Thread (s)	Stage Total 16 Threads (s)	Stage Total (16 Threads+ O/H) (s)	T_i ($\times 10^{-5}$ s)
1	0.059	0.006	0.007	41.4
2	97.610	10.139	11.024	2551.8
3	4.078	0.424	0.460	11.8
4	6.015	0.625	0.679	5.8
Total	107.761	11.193	12.170	

in Figure 14 and the distribution of the number of variants executed is influenced by n_t and f_{DFS} (see a single color in Figure 14). We parameterize the depth-first performance model as a function of n_p, which is the total number of non-pruned variants executed (conflating n_t and f_{DFS}).

We present three different performance models and time estimates. In the performance evaluation of the depth-first approach, we generated a schedule with a random ordering of variants to show the robustness of the pruning heuristic. The last stage (S_4 for the Siple Station Experiment signal processing chain) defines the traversal of a single variant. If we select a variant at S_n (the leaf level) at random without replacement, it will influence the probability that previous stages in its traversal, $S_i < S_n$ (i.e., a stage closer to the root), will be executed again.

In the first model, we ignore the above effect and assume that non-leaf nodes are randomly sampled with replacement (they can be sampled again), although leaf nodes (at S_n) are always sampled without replacement. Since this potentially overestimates the amount of reuse in the non-leaf stages, it may underestimate the time. Let N_i be the number of nodes at stage S_i (i.e., $N_i = \prod_{j=1}^{i} |S_j|$). The model is as follows:

$$T_{middle}^{DFS}(T, N, n_p) = n_p\, T_n + \sum_{i=1}^{n-1} N_i \left(1 - \left(\frac{N_i - 1}{N_i}\right)^{n_p}\right) T_i. \tag{2}$$

An alternative to the above model is to assume that all non-leaf stages (S_i, where $i < n$) are executed exactly once before any can be selected again. This is a worst-case scenario, as it minimizes the amount of data reuse among variants. The model is as follows:

$$T_{upper}^{DFS}(T, N, n_p) = \sum_{i=1}^{n} argmin[N_i, n_p]\, T_i. \tag{3}$$

Finally, the last model assumes that the search maximizes selecting the non-leaf nodes that have already been executed; increasing the probability that a stage is executed again. This is the opposite of the $T_{upper}^{DFS}(T, N, n_p)$ model. Thus, this represents the lower bound on the execution time. The model is as follows:

$$T_{lower}^{DFS}(T, N, n_p) = n_p\, T_n + \sum_{i=1}^{n-1} T_i \left(\left\lceil n_p \frac{N_i}{N_n} \right\rceil \right) \tag{4}$$

We validate the three models on $E1$. The values of the average stage times, T_i, are shown in Table 2, derived from measuring the sequential response times at each stage. These times are then reduced based on the speedup, overheads, and N_i. The three models are compared to the depth-first results in Figure 14 (b). From Figure 1, we see that the T_{upper}^{DFS} and T_{lower}^{DFS} models bracket the expected response time. T_{middle}^{DFS} matches the measurements in the region that is of greatest interest (aggressively pruning, where the fraction pruned is $\gtrsim 0.985$). Note that when the fraction pruned is < 0.8, then the response time in some of the trials exceeds the exhaustive search time. This is due to the load imbalance that occurs when there is a high fraction of variants that are executed, which does not occur in the exhaustive search. Furthermore, load imbalance is not considered in the performance models, which explains why some of the response times exceed T_{upper}^{DFS}.

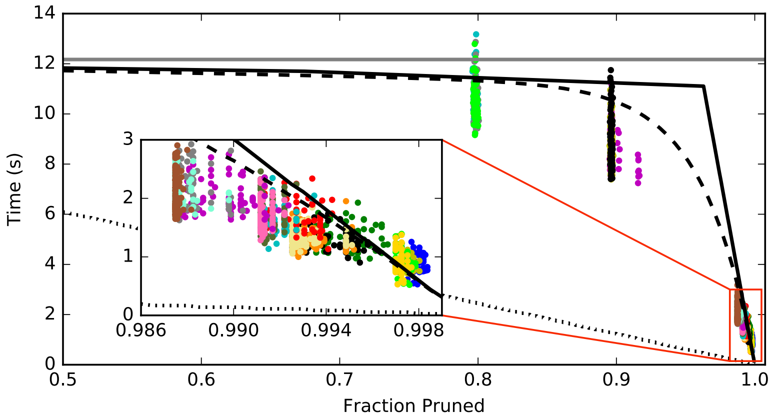

Figure 1: Comparison between the DFS models (black dashed: T_{middle}^{DFS}, black solid: T_{upper}^{DFS}, black dotted: T_{lower}^{DFS}) plotted on results from Figure 14 (b). The horizontal gray line corresponds to the the exhaustive search time.