Scheduling Instructions on Processors with Incomplete Bypass

Florent Blachot * Guillaume Huard Jonathan Pecero Erik Saule
Denis Trystram †

1 Introduction

New generation of embedded VLIW processors such as the ST200 implements a bypass mechanism between functional units. Thanks to this mechanism, the result of an arithmetic operation can be directly fetched as an operand of another operation. This eliminates the usual data dependence latency between the two operations. Nevertheless, when production costs enforce strong contraints on die area, an incomplete bypass is implemented to save space. An incomplete bypass is a partition of functional units into small groups in which the bypass is complete. It implies that, depending on placement, the latency between dependent operation is either null or a fixed number of cycles. An upcoming revision of the ST200 will use an incomplete bypass.

In this work, we model the problem of scheduling instructions for the ST200 processor as a problem of scheduling unitary tasks on a hierarchical architecture with communication delays. We propose a guaranteed heuristic to solve it. Our heuristic is within a factor from the optimal that depend on the number of groups induced by the incomplete bypass mechanism. We prove that this bound is tight and preliminary experiments demonstrate the effectiveness of the proposed heuristic.

2 Model

As usual, a program is represented by a directed graph $G = (T, E)$ where the set of vertices T is a set of n tasks. This application graph has to be scheduled on P, a set of mM processors (i.e. our functional units). Those processors are organized in M groups of size m also named hierarchies in the litterature [1]. The communication delay between two processors in the same hierarchy is negligible, while it takes ρ units of time if the two processors belong to different hierarchies. The k-th processor of the l-th hierarchy is denoted $P_{lk} \in H_l$.

A solution to the problem of scheduling an application graph on the processors is made of two functions $\pi : T \to P$ and $\sigma : T \to \mathbb{N}^+$ that give respectively a processor assignement and an

*This work is supported by STMicroelectronics Grenoble. 12, rue Jules Horowitz - BP 127 F-38019 Grenoble, France.

†{florent.blachot | guillaume.huard | jonathan.pecero | erik.saule | denis.trystram}@imag.fr. Laboratoire d'Informatique de Grenoble. ENSIMAG - antenne de Montbonnot ZIRST 51, avenue Jean Kuntzmann 38330 Montbonnot Saint Martin, France.

execution date for each task of the graph. This solution has to meet resources and precedence constraints. We are aiming at minimizing the *Makespan* of the schedule.

This problem is denoted in the standard 3-fields notation $\alpha \mid \beta \mid \gamma$ extended by Giroudeau to the hierarchical problem as $P_M(P_m)|prec, p_j = 1, c = (\rho, 0)|C_{max}$ [1]. Notice that the upcoming ST200 revision is a particular case where $M = 2$, $m = 3$ and $\rho = 2$.

3 A first heuristic

List Scheduling on the whole machine has a performance ratio of $2 + \rho - \frac{1}{mM}$. But the ρ factor is large and we aim at getting rid of it. To achieve this goal we first characterize a good class of heuristics that is guaranteed within a factor that do not depend on ρ. We first need to introduce the notion of idle time due to communications before characterizing schedules that use a main hierarchy.

In a schedule (π, σ), an **idle time** is a pair (P_{kl}, t) such that $\nexists i \in T, \pi(i) = P_{kl}$ *and* $\sigma(i) = t$.

An idle time (P_{kl}, t) is said to be **due to communication** if $\exists i \in T, \exists j$ successor of $i, \sigma(j) > t$ *and* $\pi(j) \in H_k$ *and* $\pi(i) \notin H_k$ *and* $\sigma(i) \in [t - \rho; t - 1]$

Schedules that do not contain idle time due to communication and that comply with the Graham criterion are very interesting in the hierarchical case. As the following proposition states, they have a performance guarantee that depend on the number of hierarchies. The proof of this proposition is inspired by [2] and is not presented here.

Proposition 1 *A schedule without idle time due to communication and complying with the Graham criterion on at least one hierarchy is* $M + 1 - \frac{1}{m}$ *optimal.*

Notice that scheduling the whole graph only on the processors of a single hierarchy using classical list scheduling produces a schedule without idle time due to communications and complying with the Graham criterion on one hierarchy. Let call this scheduling algorithm GSingle, it is $M + 1 - \frac{1}{m}$ optimal.

Corollary 2 *In the* ST200 *case (*$M = 2$ *and* $m = 3$*), A schedule without idle time due to communication and complying with the Graham criterion is* $\frac{8}{3}$ *optimal.*

We can notice that this ratio is better than $\frac{23}{6}$ derived from general list scheduling with communication delays.

4 A new solution

GSingle only uses a M-th of available resources. Thus, it is likely to produces almost systematically worst case schedule scenario. In this section, we propose the BHPST algorithm that produces schedules without idle time due to communications and complying with Graham criterion while making use of all the available resources.

BHPST works by choosing a master hierarchy and applying a modified list scheduling on all the resources. The list scheduling iterates over schedule time while maintaining a list of ready tasks. A task is scheduled on a non master hierarchy only if it is ready for all the hierarchies. If at any time the master hierarchy is idle, recent schedule decisions (in the last ρ units of time)

are examined: if there exists a task scheduled recently on a non master hierarchy, it is brought back to the master hierarchy. This light schedule modification eliminates any possibility of generating an idle due to communications on the master hierarchy. Furthermore, because it uses a list scheduling as a basis, our heuristic comply with the graham criterion, at least on the master hierarchy. Thus, the proposed heuristic comply to **Proposition** 1 and is $M + 1 - \frac{1}{m}$ of optimal. When $M \leq \rho$, this bound is tight.

We have implemented the two proposed heuristic (GSingle and BHPST) and a modified version of classical List Scheduling algorithm on two Machines with communication delays. More precisely, an adapted version of Earliest Task First algorithm [3]. We have tested the three different heuristics over a set of Traced graphs [4]. The traced graphs represent some of numerical parallel application programs obtained via a parallelizing compiler. Preliminary results showed that BHPST outperform GSingle algorithm and is slightly better to classical List Scheduling with communications when run over this set of Traced graphs.

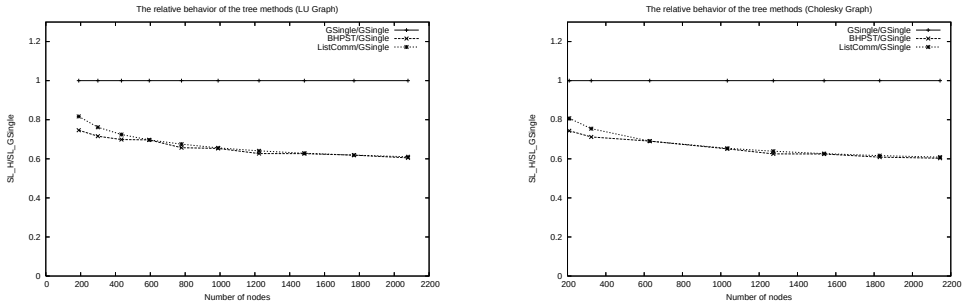

Figure 1: Simulation results for the LU factorization and the Cholesky factorization graphs. The y-axis represents the ratio between the makespan of the three heursitics and the makespan obtained by GSingle.

5 Concluding remarks and future works

In this article we proposed a new model to represent processor architectures that use an incomplete bypass mechanism. This new model can be directly used with classical algorithms for scheduling task with communication delays. Nevertheless, we proved that in the hierarchical case, it is possible to improve on the general approximation guarantee. We showed that a simple list scheduling algorithm applied on only a part of the machine was actually a $M + 1 + \frac{1}{m}$ approximation (where the processor has M hierarchies of m units). We improved this algorithm by proposing BHPST, a heuristic with the same guarantee that makes use of all the available resources. Finally, preliminary experiment results show the effectiveness of BHPST.

Future works will focus on the establishment of a link between guarantees related to communication delays and guarantees related to the number of hierarchies. We aim at improving the classical bound that depend on communication delays by taking advantage of the hierarchical structure of the architecture.

References

[1] E. Bampis and R. Giroudeau and J-C. König (2003). An approximation algorithm for the precedence constrained scheduling problem with hierarchical communications. *Theor. Comput. Sci.*. 290(3):1883-1895. ISSN 0304-3975. Elsevier Science Publishers Ltd. Essex, UK.

[2] R. L. Graham (1969). Bounds on Multiprocessing Timing Anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416-429. March, 1969.

[3] J.J. Hwang, Y. Ch. Chow, F. D. Angers and Ch. Y. Lee (1989). Scheduling precedence graphs in systems with Interprocessor communication times. *SIAM J. Comput.*, 18(2):244-257, April

[4] Kwok Y.K and Ahmad I. (1999). Benchmarking and comparison of the task graph scheduling algorithms. *Journal of Parallel and Distributed Computing*, 59(3):381-422.