

---

**発表概要**

---

**並列プログラムデバッグのための巻き戻し実行機構**

丸山 真佐夫<sup>†</sup> 山本 繁 弘<sup>††</sup>  
大野 和彦<sup>††</sup> 中島 浩<sup>††</sup>

再演実行を基礎とする従来の並列プログラムのデバッグ手法では、バグの原因をたどる過程で、頻りにプログラムの再実行をしなくてはならない。実行の行き過ぎによる再実行の回数を減らすためには、実行途中でブレークポイント設定が必要になり、ユーザの負担が大きくなる。そこで我々は、再演実行手法に基づきながら、プログラム先頭からの再実行をせずに並列プログラムを過去の状態に戻し、そこから実行を再開することを可能にする“巻き戻し実行”機構を提案する。本提案の巻き戻し実行機構では、並列プログラムを構成する任意のプロセスを、任意の受信イベントの時点に戻すことができ、これを基礎にすべてのプロセスまたは一部のプロセスだけをプログラムの途中から実行させることができる。我々は並列言語 Orgel に対して巻き戻し実行機構を実装し、性能評価を行った。その結果通常実行に対して、イベント順序保存実行で 7%、巻き戻しのための状態保存実行で 13% 増という、小さいオーバーヘッドで動作させることができた。

**A Rollback Execution Mechanism for Parallel Program Debugging**

MASAO MARUYAMA,<sup>†</sup> SHIGEHIRO YAMAMOTO,<sup>††</sup> KAZUHIKO OHNO<sup>††</sup>  
and HIROSHI NAKASHIMA<sup>††</sup>

In debugging a parallel program with conventional replay based method, the programmer has to rerun the program repeatedly from its beginning, because the code the programmer wants to examine next might have already gone beyond the breakpoint. To prevent the program from overrunning, the programmer must set breakpoints with much care in the complicated parallel program. Thus we propose a rollback mechanism, which allows the programmer to rerun the program halfway of it. Using this mechanism, the programmer may rollback any process of the target program to any receive event on its event graph. We applied our rollback mechanism to a parallel programming language named Orgel, and evaluate the overhead of logging and rollback. The result shows that execution time of event logging and computational state saving mode for rollback are only 7% and 13% larger than normal execution respectively.

(平成 13 年 7 月 26 日発表)

---

<sup>†</sup> 木更津工業高等専門学校  
Kisarazu National College of Technology  
<sup>††</sup> 豊橋技術科学大学工学部  
Toyohashi University of Technology