

Процессный подход к моделированию и верификации параллельных программ

А. М. Миронов¹

В докладе излагается новая математическая модель параллельных программ и приводится пример ее применения для верификации параллельной программы перемножения матриц.

Ключевые слова: параллельные программы, распределенные процессы, верификация.

1. Введение

Параллельные программы – это программы, предназначенные для исполнения на многопроцессорных вычислительных системах (МПВС). Проблема разработки корректных и безопасных параллельных программ представляет в настоящее время исключительно высокую актуальность. Формальное обоснование свойств корректности и безопасности параллельных программ (называемое также **верификацией** параллельных программ) является сложной математической задачей. Существующие методы решения данной задачи пригодны лишь для достаточно ограниченного класса параллельных программ.

Одним из наиболее широко используемых языков для описания параллельных программ является язык MPI (Message Passing Interface). В настоящей работе вводится новая математическая модель параллельных программ, на основе которой можно решать задачи верификации параллельных программ, представленных на некотором подмножестве MPI. Введенная модель иллюстрируется применением к решению задачи верификации MPI-программы умножения матриц.

Наиболее важной особенностью предлагаемого подхода является возможность его применения для верификации параллельных программ, которые могут порождать неопределенное число процессов. Среди других подходов к моделированию и верификации таких программ следует отметить подход в [4]. В этой работе представлен инструмент ParTypes для моделирования и верификации параллельных программ, порождающих неопределенное число процессов. К сожалению, данный инструмент

¹ *Миронов Андрей Михайлович* — доцент, к.ф.м.н., кафедра математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: amironov66@gmail.com, телефон +7-(916)-462-20-36, 119991, Россия Москва, ГСП-1, Ленинские Горы, д.1, механико-математический факультет, комн. 12-01.

Mironov Andrew Mikhailovich — associate professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems

не работает для параллельных программ со свойством wildcard receives, которое имеет следующий смысл: в параллельной программе допускается использование действий приема сообщений от произвольного процесса (т.е. при выполнении действия такого типа номер процесса, от которого принимается сообщение, можно определить лишь после выполнения этого действия). В частности, при помощи подхода, лежащего в основе данного инструмента, невозможно верифицировать MPI программу для умножения матриц, рассматриваемую в настоящей статье. Существуют и другие подходы с использованием символьного исполнения и проверки модели, см. например [1]–[3], [5]–[9], однако все эти подходы пригодны лишь для анализа параллельных программ, порождающих заранее заданное количество процессов. Преимуществом данной работы является пример решения задачи верификации такой MPI-программы, которую невозможно верифицировать на основе использования других известных моделей параллельных программ.

2. Описание модели MPI-программы

Модель MPI-программы представляет собой распределенный процесс (РП) \mathcal{P} , который является семейством последовательных процессов (ПП) P_i ($i \geq 0$). Каждый ПП представляется графом, ребра которого помечены действиями. Действиями могут быть присваивания, условные переходы, и команды отправки и приема сообщений. Кроме того, в записи действий могут использоваться вспомогательные переменные, операции с которыми не являются исполняемыми операциями программы, а предназначены лишь для облегчения записи утверждений о программе.

Элементарные действия (ЭД) – это записи следующих видов:

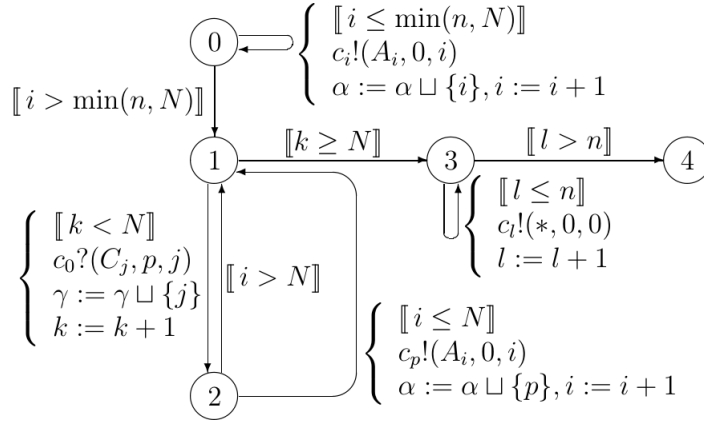
$c!e, \quad c?e, \quad e := e', \quad \llbracket \varphi \rrbracket$, где c – имя канала, e, e' – термы, φ – формула,

которые называются **посылкой** сообщения e в канал c , **приемом** сообщения e из канала c , **присваиванием**, и **условным переходом**, соответственно. **Действие** – это конечная последовательность ЭД, в которой имеется не более одного посылки или приема.

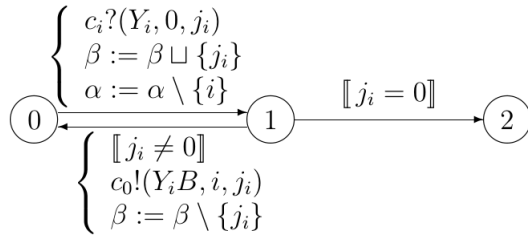
Например, РП $\mathcal{P}_\Pi = \{P_i \mid i = 0, 1, \dots\}$, моделирующий MPI-программу Π умножения матриц, имеет следующий вид. Входными переменными Π являются A, B (матрицы-сомножители), N (число строк в A), n (число порожденных процессов). Выходная переменная – C (матрица-результат, после выполнения программы значение C должно быть равно произведению AB). Вспомогательные переменные – α, β, γ их значения имеют следующий смысл: для каждого состояния s РП \mathcal{P} α^s состоит из номеров каналов из $\{c_1, \dots, c_n\}$ с непустым содержимым

в состоянии s , β^s состоит из номеров строк матрицы A , для которых в состоянии s вычисляется их произведение на B , γ^s равно множеству номеров всех строк, которые P_0 записал в C до момента прихода РП \mathcal{P}_Π в состояние s .

ПП P_0 имеет следующий вид:



Последовательный процесс P_i для $i > 0$ имеет следующий вид:



Верификация РП \mathcal{P} заключается в доказательстве утверждения о том, что после завершения его выполнения верно равенство $C = AB$. Данное утверждение доказывается с помощью следующей теоремы.

Теорема 1. Для каждого состояния s РП \mathcal{P} верны утверждения:

- 1) $\alpha^s \subseteq \{1, \dots, i^s - 1\}$, $i^s - 1 \leq N \mid |\gamma^s| = k^s \leq N$,
- 2) если $at_{P_0}^s = 1$ и $k^s < N$, то $k^s < i^s - 1$, $[c_0]_2^s \cap \alpha^s = \emptyset$,
- 3) $\forall i = 1, \dots, n \mid [c_i]^s = 1$, если $i \in \alpha^s$, и 0 иначе,
- 4) $at_{P_i}^s = 1 \Rightarrow (i \notin \alpha^s) \wedge (j_i^s \notin \gamma^s)$,
- 5) $at_{P_0}^s = 2 \Rightarrow p^s \notin \alpha^s$, $p^s \in \{1, \dots, i^s - 1\}$,
- 6) $[c_1]_3^s \sqcup \dots \sqcup [c_n]_3^s \sqcup \beta^s \sqcup [c_0]_3^s \sqcup \gamma^s = \{1, \dots, i^s - 1\}$,
- 7) $\forall p \in \alpha^s \mid [c_p]^s$ имеет вид $\{(A_i, 0, i)\}$, где $i \in \{1, \dots, N\}$,
- 8) каждый элемент $[c_0]^s$ имеет вид $(A_i B, p, i)$, где $i \in \{1, \dots, N\}$,

$$9) \forall j \in \gamma^s \ C_j = A_j B, \ k^s = N \Rightarrow \gamma^s = \{1, \dots, N\}.$$

Все эти утверждения обосновываются индуктивно: они верны в начальном состоянии РП \mathcal{P} , и сохраняют свою истинность после каждого перехода РП \mathcal{P} . Также нетрудно доказать, что в определенном выше РП \mathcal{P} нет тупиковых состояний, и любое выполнение РП \mathcal{P} завершается после конечного числа шагов.

Список литературы

- [1] V. Forejt, S. Joshi, D. Kroening, G. Narayanaswamy, S. Sharma, Precise Predictive Analysis for Discovering Communication Deadlocks in MPI Programs, In: ACM Transactions on Programming Languages and Systems, V. 39, Issue 4, 2017, 1–27.
- [2] Gopalakrishnan, G., Kirby, R.M., Siegel, S., Thakur, R., Gropp, W., Lusk, E., De Supinski, B.R., Schulz, M., Bronevetsky, G.: Formal analysis of MPI-based parallel programs. Communications ACM 54(12), 82-91, (2011).
- [3] W. Hong, Z. Chen, H. Yu, and J. Wang. Evaluation of model checkers by verifying message passing programs. Science China Information Sciences, volume 62, Article number: 200101 (2019).
- [4] H. A. Lopez, E. R. B. Marques, F. Martins, N. Ng, C. Santos, V.T. Vasconcelos, N. Yoshida, Protocol-based verification of message-passing parallel programs, Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications - OOPSLA 2015, p. 280-298.
- [5] Z. Luo, M. Zheng, and S. F. Siegel. Verification of MPI programs using CIVL. In EuroMPI. 6:1-6:11, 2017.
- [6] S. F. Siegel, Model Checking Nonblocking MPI Programs. In International Workshop on Verification, Model Checking, and Abstract Interpretation VMCAI 2007, pp 44-58.
- [7] Siegel, S., Mironova, A., Avrunin, G., Clarke, L.: Combining symbolic execution with model checking to verify parallel numerical programs. ACM Transactions on Software Engineering and Methodology, Volume 17, Issue 2, 2008, 1–34.
- [8] Sarvani S. Vakkalanka, Ganesh Gopalakrishnan, and Robert M. Kirby. Dynamic Verification of MPI Programs with Reductions in Presence of Split Operations and Relaxed Orderings. In International Conference on Computer Aided Verification CAV 2008, pp 66-79.

- [9] H. Yu, Z. Chen, X. Fu, J. Wang, Z. Su, J. Sun, C. Huang, and W. Dong. Symbolic Verification of Message Passing Interface Programs. In ICSE '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, p. 1248-1260, 2020.

Process approach to modeling and verification of parallel programs

Mironov A.M.

The report presents a new mathematical model of parallel programs and gives an example of its application for verification of a parallel program for matrix multiplication. *Keywords:* parallel programs, distributed processes, verification.

References

- [1] V. Forejt, S. Joshi, D. Kroening, G. Narayanaswamy, S. Sharma, Precise Predictive Analysis for Discovering Communication Deadlocks in MPI Programs, In: ACM Transactions on Programming Languages and Systems, V. 39, Issue 4, 2017, 1–27.
- [2] Gopalakrishnan, G., Kirby, R.M., Siegel, S., Thakur, R., Gropp, W., Lusk, E., De Supinski, B.R., Schulz, M., Bronevetsky, G.: Formal analysis of MPI-based parallel programs. Communications ACM 54(12), 82-91, (2011).
- [3] W. Hong, Z. Chen, H. Yu, and J. Wang. Evaluation of model checkers by verifying message passing programs. Science China Information Sciences, volume 62, Article number: 200101 (2019).
- [4] H. A. Lopez, E. R. B. Marques, F. Martins, N. Ng, C. Santos, V.T. Vasconcelos, N. Yoshida, Protocol-based verification of message-passing parallel programs, Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications - OOPSLA 2015, p. 280-298.
- [5] Z. Luo, M. Zheng, and S. F. Siegel. Verification of MPI programs using CIVL. In EuroMPI. 6:1-6:11, 2017.
- [6] S. F. Siegel, Model Checking Nonblocking MPI Programs. In International Workshop on Verification, Model Checking, and Abstract Interpretation VMCAI 2007, pp 44-58.

- [7] Siegel, S., Mironova, A., Avrunin, G., Clarke, L.: Combining symbolic execution with model checking to verify parallel numerical programs. *ACM Transactions on Software Engineering and Methodology*, Volume 17, Issue 2, 2008, 1–34.
- [8] Sarvani S. Vakkalanka, Ganesh Gopalakrishnan, and Robert M. Kirby. Dynamic Verification of MPI Programs with Reductions in Presence of Split Operations and Relaxed Orderings. In *International Conference on Computer Aided Verification CAV 2008*, pp 66-79.
- [9] H. Yu, Z. Chen, X. Fu, J. Wang, Z. Su, J. Sun, C. Huang, and W. Dong. Symbolic Verification of Message Passing Interface Programs. In *ICSE '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, p. 1248-1260, 2020.