Разработка параллельного алгоритма генерации состояний при проверке моделей для систем с неразделяемой памятью

При создании параллельных алгоритмов или сетевых протоколов возникает задача их верификации — поиска возможных взаимоблокировок, непредвиденных сценариев работы и нарушения условий работоспособности. Одним из подходов к решению этой задачи является автоматическая формальная верификация полным перебором состояний системы, называемая *проверкой модели* [1].

Одним из наиболее распространенных способов описания таких моделей является язык Promela. Модель в последнем описывается в виде набора процессов, имеющих доступ к общим переменным и использующим очереди для передачи сообщений. Проверяемые утверждения выражаются в виде простых утверждений о значениях переменных или при помощи составления из таких утверждений LTL-формул [1].

Поскольку при проверке модели обходится весь граф состояний, который в общем случае имеет циклы, необходимо хранить в памяти множество посещенных состояний.

Актуальность. Для моделей, описывающих взаимодействие процессов или участников сетевого протокола, наблюдается комбинаторный рост числа состояний. Модели среднего размера, например, модель RIP-протокола для системы из 4 маршрутизаторов, насчитывает порядка 10^8 состояний, что делает невозможным хранение их в памяти одной машины, а использование swap-памяти увеличивает время проверки на несколько порядков.

Наиболее распространенным средством проверки моделей является Spin [2], строящий множество состояний последовательным образом на одной машине, и, следовательно, неприменимый для проверки моделей такого размера. Имеется один экспериментальный аналог, реализующий параллельную генерацию — DiVinE, разработанный в университете Брно [3].

Постановка задачи. Разработка параллельного алгоритма генерации состояний для проверки модели с распределенным хранением состояний в различных узлах кластера.

Для решения поставленной задачи было сделано следующее:

- создан прототип верификатора моделей, описываемых на языке Promela, с поддержкой утверждений в виде предположений о значениях переменных (assert);
- предложено несколько способов распределения состояний между узлами, проведено их экспериментальное сравнение.

Основной проблемой распределенной генерации множества состояний является выбор подходящей функции распределения состояний между узлами [4]. Такая функция должна быть однозначно вычислимой по самому состоянию и обеспечивать равномерное распределение состояний между узлами.

Тривиальным решением является является использовать хэш-код самого состояния в качестве индекса хранящего его узла. При этом решении достигается высокая равномерность распределения (до 99.9%). Однако, новое состояние с большой вероятностью будет принадлежать не тому узлу, где оно было сгенерировано и число удаленных вызовов между узлами будет приближаться к числу переходов.

При альтернативном подходе индексом узла является хэш-код от состояния первых M процессов моделируемой системы, т.е. значений их локальных переменных и номеров текущих инструкций. В этом случае удаленный вызов делается лишь в результате переходов, затрагивающих состояния этих процессов. Значение M выбирается с учетом числа вычислительных узлов и процессов в моделируемой системе. Слишком маленькие

значения приводят к неравномерному распределению, при больших растет число удаленных вызовов.

Основные результаты. Реализован параллельный генератор состояний на основе алгоритма параллельного поиска в ширину с использованием указанной функции распределения.

Эксперименты на кластере из 80 узлов, имеющих 4 Гбайт памяти каждый, показывают пригодность созданного генератора для проверки моделей с числом состояний до 10^{11}. В качестве моделей использовалась задача об обедающих философов и алгоритм распределенного выбора лидера с различным числом сторон.

Проведена экспериментальная проверка предложенного распределения состояний, из которой следует, что при подходящем выборе M выигрыш во времени генерации состояний составляет до 80% в сравнении с тривиальным решением при неравномерности распределения между узлами меньше 30%.

Научная новизна работы Полученные результаты позволяют использовать автоматическую формальную верификацию для проверки моделей бóльшего размера, чем это возможно при использовании существующих средств.

Список литературы

1. *Кларк, Э. М.* Верификация моделей программ. Model Checking / Э. М. Кларк, О. Грамберг, Д. Пелед. — МЦНМО, 2002.

2. *Holzmann, Gerard J.* The model checker SPIN / Gerard J. Holzmann // *IEEE Transactions on Software Engineering*. — 1997. — Vol. 23. — Pp. 279–295.

3. *Barnat, J.* Parallel Breadth-First Search LTL Model-Checking / J. Barnat, L. Brim, J. Chaloupka // 18th IEEE International Conference on Automated Software Engineering (ASE'03). — IEEE Computer Society, 2003. — Oct. — Pp. 106–115.

4. *Lerda, Flavio.* Distributed-Memory Model Checking with SPIN / Flavio Lerda, Riccardo Sisto // Theoretical and Practical Aspects of SPIN Model Checking, 5th and 6th International SPIN Workshops, Trento, Italy, July 5, 1999, Toulouse, France, September 21 and 24 1999, Proceedings / Ed. by Dennis Dams, Rob Gerth, Stefan Leue, Mieke Massink. — Vol. 1680 of *Lecture Notes in Computer Science*. — Springer, 1999. — Pp. 22–39.

Название доклада	Разработка параллельного алгоритма генерации состояний при проверке моделей для систем с неразделяемой памятью
ФИО, ученая степень	Коротков Иван Андреевич
Должность, организация	аспирант, МГТУ им. Н. Э. Баумана
Адрес организации	1005005, Москва, 2-ая Бауманская ул., д. 5
Телефон	+7 (916) 213–40–98
Адрес эл. почты	twee@tweedle-dee.org