

**Научно-издательский центр «Открытие»
otkritieinfo.ru**

НАУКА XXI ВЕКА: НОВЫЙ ПОДХОД

**Материалы II молодёжной международной
научно-практической конференции
студентов, аспирантов и молодых учёных**

**28 сентября 2012 года
г. Санкт-Петербург**

Петрозаводск, 2012

УДК 001
ББК 72я431

НАУКА XXI ВЕКА: НОВЫЙ ПОДХОД

Материалы II молодёжной международной научно-практической конференции студентов, аспирантов и молодых учёных

Н 34 Наука XXI века: новый подход : материалы II молодёжной международной научно-практической конференции студентов, аспирантов и молодых учёных, 28 сентября 2012 года, г. Санкт-Петербург. - Петрозаводск : Петропресс, 2012. - 144 с. В надзаг.: Науч.-изд. центр «Открытие». - Библиогр. в конце ст. - ISBN 978-5-8430-0201-5.

Представлены материалы докладов II молодёжной международной научно-практической конференции студентов, аспирантов и молодых учёных «Наука XXI века: новый подход».

В материалах конференции представлены результаты новейших исследований в различных областях науки: биологии и экологии, географических наук и наук о Земле, информатики и медицины, сельскохозяйственных наук, архитектуры и строительства, технических, экономических, юридических наук, философии, филологии, истории. Сборник представляет интерес для научных работников, аспирантов, докторантов, соискателей, преподавателей, студентов — для всех, кто хотел бы сказать новое слово в науке.

© Научно-издательский центр «Открыт

ISBN 978-5-8430-0201-5

РАЗРАБОТКА МЕТОДА ПРЕДОТВРАЩЕНИЯ ВОЗНИКНОВЕНИЯ КОЛЛИЗИЙ ПРИ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКЕ ДАННЫХ В ДЕЦЕНТРАЛИЗОВАННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

**В. Н. Ткачѳв¹, А. Б. Анненков², В. Е. Саваневич¹,
А. Б. Брюховецкий³**

¹*Харьковский национальный университет радиоэлектроники, Харьков, Украина, vitalii.tkachov@kture.kharkov.ua;*

²*Запорожский институт экономики и информационных технологий, Запорожье, Украина;*

³*Национальный Центр управления и испытаний космических средств, г. Евпатория-19, Украина*

Децентрализованные вычислительные системы с параллельной обработкой данных появились во второй половине 80-х годов [1].

Подходы к предотвращению коллизий в различных операционных системах сложилась на фоне развития сетевых технологий. Первые методы предотвращения коллизий были основаны на принципах организации технологии Ethernet. Так, в технологии Ethernet, чтобы корректно обработать коллизию, все вычислительные станции одновременно наблюдают за возникающими в разделяемой среде сигналами. В UNIX системах было принято решение использовать метод блокировок [2]. В операционных системах семейства windows для предотвращения коллизиям ОС накладывает "замки" на порции данных [3]. Общим недостатком вышеуказанных методов является необходимость изменять структуру поступающих порций данных и отсутствие мультитиплатформенности.

Основной целью работы является разработка метода предотвращения возникновению коллизий при обработке данных, устраняющий вышеуказанные недостатки, быть платформонезависимым.

Порции данных от источника поступают во входной виртуальный буфер децентрализованной вычислительной системы, при этом условия поступления таковы, что источником может быть любая система хранения данных, телескоп,

виртуальная обсерватория, пользователи [4]. Система обработки данных, представляющая собой совокупность экземпляров программы, функционирующих на вычислительных станциях, к моменту начала обработки порций входных данных получает команду (сообщение) о начале работы.

На любой из вычислительных станций вычислительной системы установлены различные операционные системы, на базе которых функционируют экземпляры программы. Количество экземпляров программы равно количеству ядер процессоров вычислительных станций вычислительной системы. Каждый экземпляр программы в любой момент времени может выполнять любой из этапов обработки данных. Предполагается, что все структурные элементы вычислительной системы не выходят из строя во время цикла обработки данных.

В основе разрабатываемого метода лежит использование сообщений-запросов и сообщений-состояний, являющихся своего рода «блокираторами».

Для каждой порции данных вычислительный метод предотвращения коллизии при параллельной обработке данных заключается в следующем.

1. На нулевом этапе обработки экземпляр программы ищет во входном буфере очередную порцию данных.

2. При отсутствии во входном буфере порции данных, экземпляр программы выполняет поисковый цикл сообщений состояния по хранилищам порций данных на всех этапах обработки.

- 2.1. При обнаружении экземпляром программы в хранилище порций данных сообщения состояния, которое позволяет проводить обработку серий, он ищет сообщение-запрос на обработку этой серии порций данных.

- 2.1.1. При отсутствии сообщения-запроса о серии порции данных, переход на пункт 8.2.

- 2.1.2. При наличии сообщения-запроса о серии порции данных, переход на пункт 1.

- 2.2. При обнаружении экземпляром программы в хранилище порций данных сообщения состояния, которое не

позволяет проводить обработку серий, он продолжает поиск в очередном хранилище порций данных.

2.2.1. При отсутствии задач экземпляра программы делает уведомление внешней программе о переходе в режим работы «Ожидание».

2.2.2. При поступлении уведомления от внешней программы о выходе из режима «Ожидание» до истечения заданного времени, экземпляр программы переходит к пункту 1.

2.2.3. При отсутствии уведомления от внешней программы о выходе из режима «Ожидание» по истечению времени, экземпляр программы завершает работу.

3. При наличии во входном буфере порции данных, экземпляр программы ищет в очереди сообщений-запросов очередное сообщение-запрос о порции данных.

3.1. При наличии сообщения-запроса о порции данных, переход к пункту 1.

3.2. При отсутствии сообщения-запроса о порции данных, экземпляр программы, создает в очереди сообщений-запросов сообщение-запрос.

4. Функционирующий на вычислительной станции, экземпляр программы, ожидает наперед заданное время.

5. Экземпляр программы ищет в очереди сообщений-запросов сообщение-запрос о порции данных.

5.1. При обнаружении экземпляром программы нескольких сообщений-запросов, данная порция данных будет перемещена этим экземпляром программы при условии, что номер ее сообщения запроса меньше номеров других сообщений-запросов.

5.2. При тождестве номеров сообщений-запросов, порция данных будет перемещена этим экземпляром программы при условии, что номер вычислительной станции, на которой он функционирует, меньше номеров других вычислительных станций.

5.3. Остальные экземпляры программы, переходят к пункту 1.

6. Функционирующий на вычислительной станции, экземпляр программы перемещает порцию данных своей серии

из входного буфера в хранилище порций данных для первого этапа обработки.

7. Экземпляр программы создает сообщение состояния о формировании новой серии порций данных с целью предотвращения обработки несформированной серии другим экземпляром программы.

7.1. Экземпляр программы проверяет каждую входящую порцию данных на предмет выявления последней из этой серии.

7.1.1. При обнаружении не последней порции данных из серии, экземпляр программы переходит к пункту 1.

7.1.2. При обнаружении последней порции данных из серии, или, если серия содержит единственную порцию данных, экземпляр программы создает сообщение состояния о сформированной серии порций данных.

7.2. Экземпляр программы удаляет сообщение состояния о формировании серии порций данных.

8. Экземпляр программы ищет в очереди сообщений-запросов сообщение-запрос о серии порций данных.

8.1. При наличии сообщения-запроса о серии порций данных, переход к пункту 1.

8.2. При отсутствии сообщения-запроса, экземпляр программы, создает в очереди сообщений-запросов соответствующее сообщение-запрос.

9. Функционирующий на вычислительной станции, экземпляр программы, ожидает наперед заданное время.

10. Экземпляр программы ищет в очереди сообщений-запросов сообщение-запрос о серии порции данных.

10.1. При обнаружении экземпляром программы нескольких сообщений-запросов, данная серия порций данных будет обрабатываться этим экземпляром программы при условии, что номер ее сообщения запроса меньше номеров других сообщений-запросов .

10.2. При тождестве номеров сообщений-запросов, серия порций данных будет обрабатываться этим экземпляром программы при условии, что номер вычислительной станции, на которой он функционирует, меньше номеров других вычислительных станций.

10.3. Остальные экземпляры программы, переходят к пункту 1.

11. Функционирующий на вычислительной станции, экземпляр программы осуществляет очередной этап обработки серии порций данных.

11.1. Перед началом обработки серии порций данных экземпляр программы, создает в очереди сообщений состояния сообщение-состояние о процессе обработки данной серии порций данных. Это предотвращает обращения экземпляра программы с целью обработки этой же серии порций данных.

11.2. После окончания обработки серии порций данных экземпляр программы создает в очереди сообщений состояния сообщение об окончании обработки этой серии порций данных. Это предотвращает обращения экземпляра программы с целью повторной обработки этой же серии порций данных.

11.3. Экземпляр программы удаляет сообщение состояние о процессе обработки ω -ой серии порций данных.

12. Функционирующий на вычислительной станции, экземпляр программы выполняет проверку этапа обработки на предмет определения его окончности из количества этапов обработки порций данных.

12.1. При определении не последнего этапа обработки, экземпляр программы переходит к пункту 7.

12.2. При определении последнего этапа обработки, экземпляр программы формирует отчет-уведомление о завершении обработки серии порций данных на последнем этапе и переходит к пункту 1.

Экспериментальные исследования проводились на российском дистанционно управляемом телескопе автоматизированной обсерватории ISON-NM [5]. Вычислительная система состояла с одной четырехядерной вычислительной станции.

Результаты обработки серии кадров в обсерватории ISON-NM были доступны через 20 минут после окончания формирования последнего кадра серии. Это дает выигрыш во времени приоритета открытия объекта примерно в 10% случаев (50-100 зарегистрированных в MPC предварительных открытий

в год), что немаловажно в условиях сложившейся конкурентной борьбы между обсерваториями мира [5].

Выводы. В статье разработан метод предотвращения коллизий при параллельной обработке данных. Метод отличается от известных тем, что роль блокирующих механизмов выполняют сообщения-запросы и сообщения-состояния, в общем виде представлены файлами; обеспечивается поддержка масштабируемости вычислительной системы.

Предложенный метод позволяет повысить показатели оперативности при обработке данных. Метод апробирован на российском дистанционно управляемом телескопе автоматизированной обсерватории ISON-NM. Метод может быть применен также в других системах параллельной обработки данных, где задействовано больше одного экземпляра программы, то есть вычислительная система состоит как минимум с одной вычислительной станции с двухъядерным процессором. Дальнейшие работы над методом целесообразно посвятить исследованию его работы на предмет выявления ошибок в больших распределенных вычислительных системах.

Литература

1. Титоренко Г.А. Автоматизированные информационные технологии в экономике / Г.А. Титоренко. – М.: Компьютер, ЮНИТИ, 1998. – 400 с.

2. Стивенс У.Р. Unix. Взаимодействие процессов / У.Р. Стивенс. – СПб.: Питер, 2006. – 576 с.

3. Деревянко А.С. Операционные системы / А.С. Деревянко, М.Н. Солощук. – Харьков: НТУ "ХПИ", 2002. – 574 с.

4. Leon Alexander. Using a network telescope to analyse Internet traffic for network forensic readiness/ Leon Alexander, Adrie Stander, Jacques Ophoff. – Annual Conference on World Wide Web Applications. – 2012.

5. «Еще раз об OLDAS» [Электронный ресурс] // – Режим доступа:

http://neoastrasoft.com/news/information_about_oldas/?lang=ru
(06.02.2012).