



ТЕОРИЯ И ПРАКТИКА МНОГОПОТОЧНОГО ПРОГРАММИРОВАНИЯ

Тема 10

Некоторые свободные от блокировок соисполняемые примитивы и их число
консенсуса

Д.ф.-м.н., профессор А.Г. Тормасов
Базовая кафедра «Теоретическая и Прикладная Информатика», МФТИ

Тема

- Регистры как самые простые блоки для построения сложных объектов.
- Атомарный снимок памяти (снапшет).
- Очередь типа FIFO, стек LIFO, двусторонние очереди double ended queue, очереди с приоритетами, наборы set.
- Множественное присваивание. Чтение-модификация-запись. Сравнение с обменом. Расширенная очередь.
- Теорема о максимальной “мощности” примитива CAS.

Регистры

- Объекты
 - наблюдать состояние которых можно методом `read()`
 - изменять состояние которых можно методом `write()`
- Типы данных
 - булевские
 - целые
 - массив
- Не будем рассматривать «явно синхронизированные» реализации (явный вызов мутекса до и после операций)
- Рассмотрим только свободную от ожидания реализацию

Типы регистров

- Безопасный (safe)
 - Операция чтения, которая не перекрывается с операцией записи, вернет значение, записанное последней операцией записи
 - В противном случае операция чтения имеет право вернуть любое возможное значение, допустимое для регистра (не обязательно то, которое кем-то писалось)
- Обычный (regular)
 - Он является “безопасным” регистром для неперекрывающихся операций чтения-записи
 - Для перекрывающихся операций он имеет право вернуть любое значение, которое было записано “параллельно” процессу операции чтения
 - Замечание: являются согласованными по периодам покоя (но не наоборот)

Типы регистров...

- Атомарный (atomic)
 - недопустимость переупорядоченности записей
 - более ранняя операция чтения не может вернуть более позднее значение, которое вернет более поздняя операция чтения
- Реализации существенно зависят от числа читателей-писателей
- Число консенсуса: 1
 - Мало для создания сложных свободных от блокировок алгоритмов
- Пример использования
 - кластерная файловая система где диски на общей шине SCSI

СНИМОК ПАМЯТИ

- Несколько ячеек памяти (массив), которые надо скопировать единым образом даже если они параллельно модифицируются
- Модификация «снимка памяти» – считывание из массива только некоторых значений – «частичный снимок»
- Число консенсуса: 1
 - Мало для создания сложных свободных от блокировок алгоритмов

Очереди и наборы

- Однотипные объекты
 - Очередь типа FIFO
 - стек LIFO
 - двусторонние очереди double ended queue
 - очереди с приоритетами priority queues
 - наборы set
- Характеризуются также числом писателей-читателей ($xR \times W$, x из $\{S, M\}$)
- Число консенсуса: 2
 - Все эти примитивы нельзя реализовать при помощи регистров (атомарных чтения и записи)

Очереди и наборы

- Интересные приложения
 - Объект типа очередь используется как строительный кирпич для создания сложных систем коммуникаций
 - архитектура гиперкуб
 - Процессы, соединенные сокетами TCP
 - TBB, Fastflow и аналогичные библиотеки
- Из очереди может достать любой процесс (S/MWMMR), тогда как в библиотеках используются (interlock-free) объекты SRSW

Множественное присваивание

- «инверсия» примитива снимка памяти
- одновременная запись в набор элементов памяти
- (m,n) присваивание
 - $n \geq m > 1$
 - общий размер поля n
 - число одновременно присваиваемых записей m

Множественное присваивание

Теорема.

Не существует свободной от ожидания реализации (m,n) присваивания через регистры

Доказательство

конструирование протокола, решающего проблему консенсуса для 2 потоков, используя объект $(2,3)$ присваивания

Множественное присваивание

Теорема.

Атомарное $(n, n(n+1)/2)$ присваивание для $n > 1$ имеет число консенсуса как минимум n .

Доказательство также ведется путем конструирования протокола консенсуса для n потоков.

Теорема.

Число консенсуса для атомарного присваивания в m элементов равно $2m-2$.

- Таким образом, реализация множественной записи требует существенно большей «сложности» примитивов, чем множественного чтения.

Чтение-модификация-запись RMW

- `a++`; (возвращает старое значение!)

Определение

нетривиальный RMW регистр – тот, который модифицирует значение ячейки

Теорема.

Любой нетривиальный RMW регистр имеет число консенсуса как минимум 2

Доказательство.

Каждый поток пишет в разделяемы RMW регистр свой номер, и если возвращаемое значение не равно записанному, то работать должен другой поток, а если равно – то пишущий.

Чтение-модификация-запись

Теорема.

Пусть задана функция модификации значения $f(v)$ для нетривиального RMW регистра. Тогда он имеет число консенсуса равное 2, если

- $f_1(v)$ и $f_2(v)$ – коммутативны: $f_1(f_2(v)) = f_2(f_1(v))$; или
- $f_1(v)$ и $f_2(v)$ – перезаписываемы: $f_1(f_2(v)) = f_1(v)$ или $f_2(f_1(v)) = f_2(v)$.
- инструкция x86 `xadd` – коммутативна
- инструкция x86 `bts` – перезаписывающая

Чтение-модификация-запись

Доказательство.

Докажем, что для 3 потоков решить задачу невозможно.
частный случай: сначала выполняется атомарно операция RMW для того потока, который выиграл, потом последовательно для двух проигравших.

Последний поток: не знает, кто из двух других выиграл!

$f_1(f_2(v)) = f_1(v)$ то он не знает, выполнялась ли вообще операция f_2

$f_1(f_2(v)) = f_2(f_1(v))$ не может отличить последовательность операций

сравнение с обменом CAS

CAS (compare and set или compare and swap), «максимально мощный» примитив

```
value CAS( value *addr, value compare, value set);
```

Атомарно возвращает старое значение ячейки (*addr), при этом если оно равно compare, то по адресу addr записывается новое значение set

На x86: инструкция CMPXCHG

Теорема.

Регистр, предоставляющий возможность выполнить операцию CAS, имеет число консенсуса равное бесконечности

сравнение с обменом CAS

Доказательство.

Инициализация: `RMW -> -1`

каждый поток номер n (0 до $N-1$) вызывает

`m = CAS(&RMW, -1, n);`

m будет одинаковым для всех потоков по определению CAS (номером первого обратившегося)

Определение несколько избыточно, можно возвращать булев признак замены.

Следствие: Любой примитив соисполнения может быть реализован через CAS как свободный от ожидания

Следствие: CAS невозможно реализовать через атомарные чтения-записи, очереди и любые другие примитивы с конечным числом консенсуса

Расширенная очередь

- Типовая очередь, где добавлена операция peek – получение первого элемента очереди без ее изменения (неразрушающее чтение)

Теорема.

Число консенсуса расширенной очереди – бесконечность.

В любом потоке номер n (n из $\{0, N-1\}$)

push(n);

$m = \text{peek}()$; // m – выиграл!

Лемма.

Реализация расширенной очереди только через операции атомарного чтения или записи, так же как и через очереди и другие примитивы с конечным значением числа консенсуса, а также их комбинации – невозможна.

Выводы

- Рассмотрели типовые примитивы и их модификации, доступные на большинстве платформ, и их числа консенсуса
- Для некоторых из них (множественное присваивание, RMW, CAS, расширенная очередь) доказаны значения их чисел консенсуса
- Определено, какие примитивы можно использовать для «взаимной реализации», а какие невозможно
- Доказательство того факта, что число консенсуса для атомарных примитивов чтения-записи равно 1, будет дано на следующем занятии

(с) А. Тормасов, 2010-11 г.

Базовая кафедра «Теоретическая и Прикладная Информатика» ФУПМ МФТИ
tor@cres.mipt.ru_

Для коммерческого использования курса просьба связаться с автором.