

Исполнитель: Масыч Матвей, ФИТ 4 курс

Руководитель: Перепелкин Владислав Александрович

Дата доклада: 29.01.2021

Зимняя школа 2021

Реализация репликации фрагментов данных для
повышения их доступности

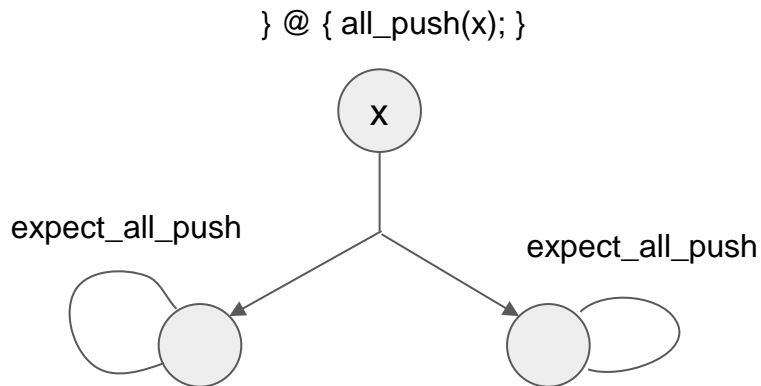
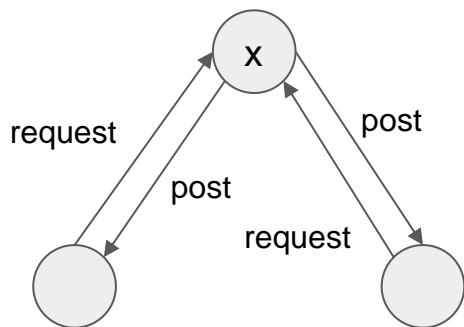
Постановка задачи

В целях оптимизации параллельных программ зачастую используется репликация данных. Репликация увеличивает доступность данных, тем самым снижая время их получения, затрачивая при этом дополнительную память.

Задача: реализовать для системы LuNA возможность репликации фрагментов данных.

Идея решения

Создать новую рекомендацию системы LuNA для указания реплицируемых фрагментов данных и реализовать алгоритмы runtime-системы для поддержки этой рекомендации.



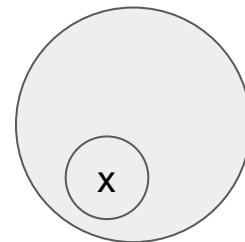
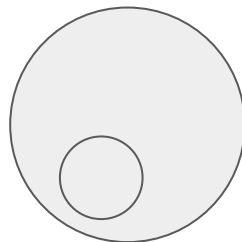
Реализация

В процессе работы для поддержки предложенной рекомендации были реализованы методы:

1. `CF::all_push(Id, DF);`
2. `CF::expect_all_push(Id);`
3. `RTS::all_push(Id, DF);`
4. `RTS::expect_all_push(Id);`
5. `RTS::_all_push(Id, DF);`
6. `Comm::bcast(tag, buf, buf_size);`
7. `CF::all_destroy(Id);`
8. `RTS::all_destroy(Id);`
9. `RTS::_all_destroy(Id);`

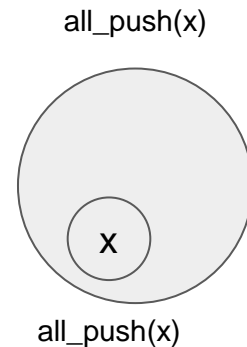
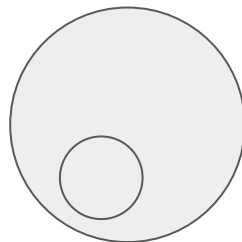
Реализация

- Фрагмент вычислений вычисляет фрагмент данных x ;



Реализация

- Фрагмент вычислений вызывает метод `all_push(x)`, который в свою очередь вызывает одноименный метод у runtime-системы;



Реализация

- Runtime-система вызывает у себя метод `_all_push(x)`, а также делает бродкаст по всем узлам, отправляя фрагмент данных `x`;
- При получении бродкаста, остальные узлы также вызывают у себя эту функцию с фрагментом данных;



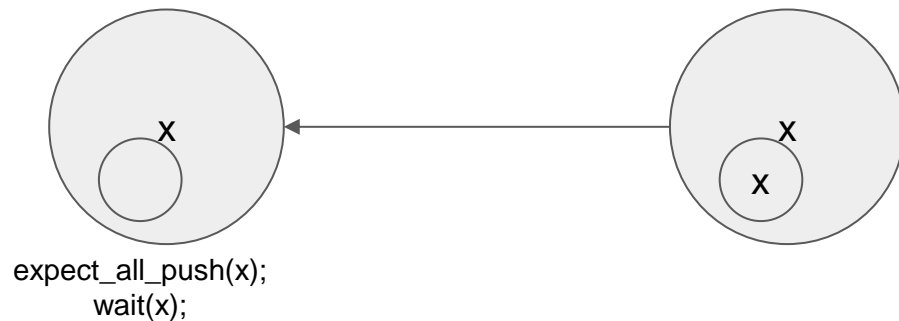
Реализация

- Метод `_all_push(x)` сохраняет фрагмент данных `x` на узле;



Реализация

- Фрагмент вычислений, может потребовать у узла нужный ему фрагмент данных, вызывая метод `expect_all_push(x)`;
- Затем фрагмент вычислений ожидает данные методом `wait(x)`;



Реализация

- Фрагмент вычислений получает данные, если они присутствуют на узле;
- Если данных на узле нет, то фрагмент продолжает их ожидать, находясь в очереди.



Тестирование

- На языке LuNA была описана программа с большим количеством фрагментов вычислений, которые используют один и тот же фрагмент данных.
- Было подготовлено два варианта программы: с использованием новых функций массовой репликации фрагмента и без.
- Был проведен замер времени работы программ:
 - 1 узел:
 - без: 0.725 sec;
 - с: 0.694 sec.
 - 4 узла:
 - без: 9.504 sec.
 - с: 1.687 sec;

Тестирование

```
#!/user/bin/luna
import c_init(int, name) as init;
import c_check(int) as check;

sub main()
{
    df x;

    cf a: init(1, x) @ {
        locator_cyclic: 1;
        all_push: x;
    };

    for i=2..100000 {
        check(x) @ {
            locator_cyclic: i;
        };
    }
}
```

Заключение

В результате работы была предложена новая рекомендация системы LuNA для репликации данных. Были реализованы алгоритмы runtime-системы, поддерживающие данную рекомендацию.

Планируется добавить поддержку предложенной рекомендации в компилятор системы LuNA, реализовать похожую рекомендацию, которая бы реплицировала данные только на указанных узлах.