Новосибирский государственный университет
*НАСТОЯЩАЯ НАУКА

Новосибирский государственный университет, НГУ
09.04.01 Информатика и вычислительная техника.
Технология разработки программных систем.

РАЗРАБОТКА СРЕДСТВ ОБЕСПЕЧЕНИЯ ОТКАЗОУСТОЙЧИВОСТИ В СИСТЕМЕ LUNA НА ОСНОВЕ КОНТРОЛЬНЫХ ТОЧЕК

Выполнил: Закиров Владимир Константинович, магистратура ФИТ НГУ, группа 21223
Научный руководитель: Малышкин Виктор Эммануилович, д.т.н., профессор, зав. каф. ПВ ФИТ НГУ
Соруководитель: Перепёлкин Владислав Александрович., ст. преп. каф. ПВ ФИТ
Рецензент: Загорулько Юрий Алексеевич, к.т.н., зав. лаб. ИСИ СО РАН

22.06.2023

* Актуальность

Преимущества исполнения задач на суперкомпьютере - большая вычислительная мощность, одновременная работа с большим количеством пользователей.

Проблема - узлы могут выходить из строя.

Важную роль при исполнении задач численного моделирования на суперкомпьютерах играют **средства отказоустойчивости**. Одним из таких средств является **контрольная точка** - сохраненное состояние системы, позволяющее продолжить выполнение программы позже без изменения поведения вычислений.

Проблему отсутствия квалификации написания параллельных программ решают системы **автоматизации программирования** - это совокупность программных средств, предназначенная для автоматизации написания программ. **LuNA** - система автоматизации программирования, позволяющая автоматически конструировать параллельные программы.

* Цель и задачи

Целью данной работы является создание механизма контрольных точек для системы LuNA.

Для достижения поставленной цели был сформирован следующий **список задач**:

1. Исследовать особенности исполнения фрагментированных программ, а также распределения ресурсов в системе LuNA
2. Разработать формат хранения контрольной точки для LuNA программ
3. Разработать и реализовать алгоритмы сохранения контрольной точки и восстановления с контрольной точки для системы LuNA

Степень разработанности проблемы

Примеры рассмотренных систем:

- Condor
  - Использует моментальные снимки
- Charm++
  - Базовой единицей параллельных вычислений в Charm++ является так называемый chare
- Disco
  - Подходы к автоматизации конструирования параллельных программ на основе частной модели вычислений MapReduce

Вывод из анализа: Рассмотренные варианты не предоставляют необходимый механизм, удовлетворяющий заявленным требованиям.

* Как работает система LuNA

Фрагментированное программирование - программа состоит из множества триплетов, состоящих из входных данных, некоторой операции без побочных эффектов и выходных данных.

Фрагмент данных - входные и выходные данные триплета.

Фрагмент вычислений - некоторая операция без побочных эффектов.

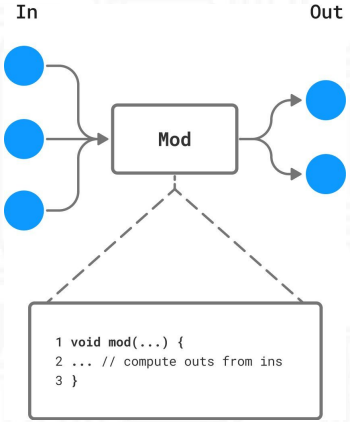

Рисунок 1 - визуализация триплета

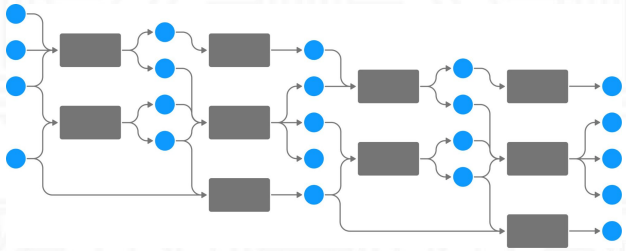

Рисунок 2 - визуализация фрагментированной программы

* Как работает система LuNA

Исполнительная система состоит из трех модулей:

1. RTS (Run Time System)
2. ThreadPool
3. MpiComm

На каждом узле системы будет запущен экземпляр исполнительной системы, состоящий из указанных модулей.

Реализовано на C++ с использованием MPI.

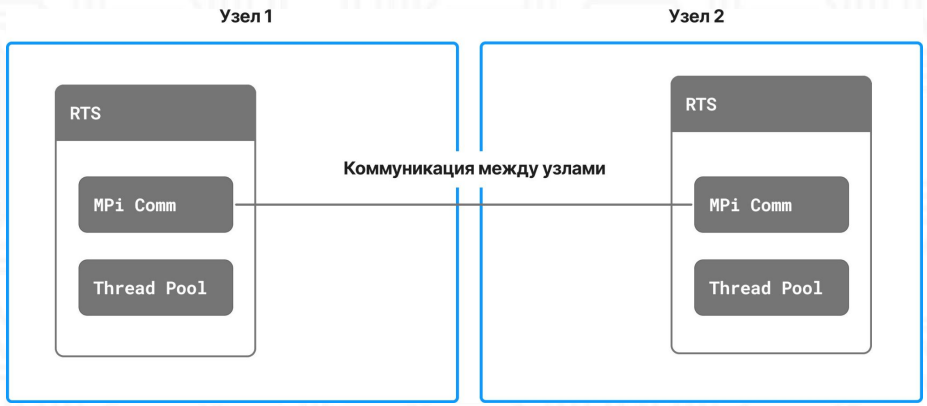

Рисунок 3 - пример исполнительной системы, запущенной на двух узлах

* Формат контрольной точки

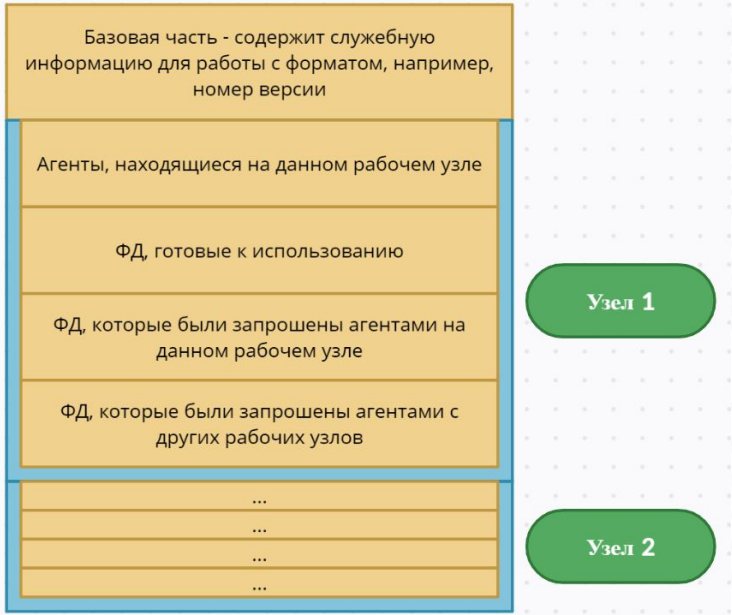

Рисунок 4 - формат контрольной точки

Формат checkpoint:

Номер версии
Количество узлов
Размер сериализованных узлов
(
Агенты:
- **Количество Агентов**
- **Массив Агентов**

Агенты для перезапуска:
- **Количество номеров**
- **Массив порядковых номеров сохраненных ранее агентов**

Полученные данные:
- **Количество полученных ФД**
- **Массив:**
  - **Идентификатор ФД**
  - **Допустимое количество использования ФД**
  - **ФД**

Запросы на данные:
- **Количество запросов**
- **Массив:**
  - **Идентификатор ожидаемого ФД**
  - **Количество ожидаемых Агентов**
  - **Массив:**
    - **Порядковый номер сохраненного ранее Агента**
    - **Идентификатор узла, запросившего ФД**

Отложенные запросы:
- **Количество запросов**
- **Массив:**
  - **Идентификатор ожидаемого ФД**
  - **Количество ожидающи**
  - **Массив: Порядковый номер сохраненного ранее Агента**

)

Предлагаемая идея решения

Для решения был адаптирован алгоритм Ченди-Лэмпорта.

Предлагаемый алгоритм:

1. Один из процессов (процесс-инициатор) инициирует создание контрольной точки, отправляя широковещательное сообщение
2. Процессы, получившие сообщения делают следующее:
   a. Запускают процесс остановки, чтобы убедиться, что процессы на время сохранения своего состояния не будут получать сообщения и, следовательно, не изменят своего состояния
3. С помощью адаптированного алгоритма Дейкстры—Шольтена процесс-инициатор убедится, что обмен сообщениями больше не происходит и отправит широковещательное сообщение с целью сохранения своего состояния
4. Процессы, получившие сообщения, делают следующее:
   a. Сохраняют свое состояние
   b. Оповещают об этом процесс-инициатор
5. Процесс-инициатор после получения всех уведомлений возобновляет работу системы

Примечание: модуль, реализующий алгоритм Дейкстры—Шольтена, уже реализован в системе и мной не разрабатывался.

Остановка системы

Основная идея остановки работы системы заключается в том, что при остановке поступления новых задач на выполнение, задачи со временем закончатся.

Для остановки был выбран следующий алгоритм действий:

1. Инициировать остановку и перестать добавлять задачи в модуле RTS
2. Определить остановку подсистемы выполнения задач ThreadPool
3. Определить остановку коммуникационной подсистемы (MPIComm)

Модификация системы

Для реализации механизма контрольных точек в существующей исполнительной системе LuNA был внесен ряд модификаций. Необходимость их внесения вызвана несоответствием следующим требованиям:

1. Все структуры данных должны быть сериализуемы
2. Уникальные объекты в каждый момент времени должны храниться только на одном рабочем узле

Для этого был модифицирован механизм отправки запросов на ФД, добавлены недостающие сериализующие методы, изменен способ хранения запросов.

* Тестирование

Цель тестирования:

- Проверить работоспособность остановки системы, когда система запущена на нескольких узлах
- Проверить работоспособность и время работы механизма сохранения и восстановления с контрольной точки на одном узле

Тест проверки работоспособности остановки системы.

Для тестирования использовался решатель модельного трехмерного уравнения в единичном кубе.

Остановка системы инициировалась, когда системы была запущана на 1-16 ядер.

После остановки работа системы возобновлялась и завершалась корректно.

* Тестирование

Тест проверит работоспособность и время работы механизма сохранения и восстановления с контрольной точки на одном узле.

Для тестирования создания контрольной точки использовалась задача умножения матриц.

- Исполнительная система запускалась на одном узле
- Узел инициировал начало создания контрольной точки
- Система корректно завершала свою работу
- Система запускалась на одном узле, но уже с контрольной точки

Сохраняемые параметры	Время с начала инициации контрольной точки, до восстановления работы
Агентов - 1893 Запросов - 185 Готовых данных - 202	0.006 сек. 0.026 сек. - время работы всей программы
Агентов - 1820 Запросов - 401 Готовых данных - 457	0.01 сек. 0.026 сек. - время работы всей программы.
Агентов - 2004 Запросов - 900 Готовых данных - 1782	0.022 сек. 0.66 сек. - время работы всей программы.
Агентов - 11841 Запросов - 1290 Готовых данных - 2561	0.05 сек. 7.13 сек. - время работы всей программы.

Таблица 1 - время работы алгоритма

Апробация

- Perepelkin V., Malkhanov V., Zakirov V. Preliminary Results on Fault Tolerance Support in LuNA System // Bull. Nov. Comp. Center, Comp. Science, 2022. Iss. 46. 2022. p. 13.
- Закиров В.К. Создание механизма сохранения и восстановления контрольных точек для системы LuNA. // Секция “Информационные технологии”, подсекция “Параллельные вычисления”, 61-й Междунар. науч. студ. конф. (МНСК) 17–26 апреля 2023 г. / Новосиб. гос. ун-т. — Новосибирск: ИПЦ НГУ, 2023.

Заключение

В системе программирования LuNA был разработан механизм контрольных точек. Для единообразного представления контрольных точек был разработан формат контрольной точки. Разработанный метод был реализован в рамках исполнительной системы в LuNA, было произведено его тестирование. Разработанный механизм сохранения и восстановления с контрольной точки может быть использован для обеспечения отказоустойчивости в системе LuNA.

На защиту выносятся следующие результаты:

- Формат контрольной точки
- Алгоритм сохранения контрольной точки и восстановления с контрольной точки в системах фрагментированного программирования
- Реализация данного алгоритма в исполнительной системе в LuNA для одного узла

*Дальнейшие планы

Дальнейшие планы:

- Добавить реализацию сохранения контрольной точки для нескольких узлов и возможность запускать работу исполнительной системы на другом количестве узлов, используя контрольную точку
- Изучить способы по инициации создания контрольной точки: как часто нужно ее создавать
- Добавить в языке LuNA возможность инициировать создание контрольной точки

Спасибо за внимание

*Остановка ThreadPool

1. Блокировка.
2. Если задач нет:
   a. **Увеличить счетчик определения остановки.**
   b. **Если счетчик равен количеству потоков:**
      i. **ThreadPool остановлен, оповестить RTS.**
   c. Снять блокировку.
   d. Ожидать поступления задач.
   e. **Декрементировать счетчика определения остановки.**
3. Если задачи есть:
   a. Достать задачу.
   b. Снять блокировку.
   c. Выполнить задачу.

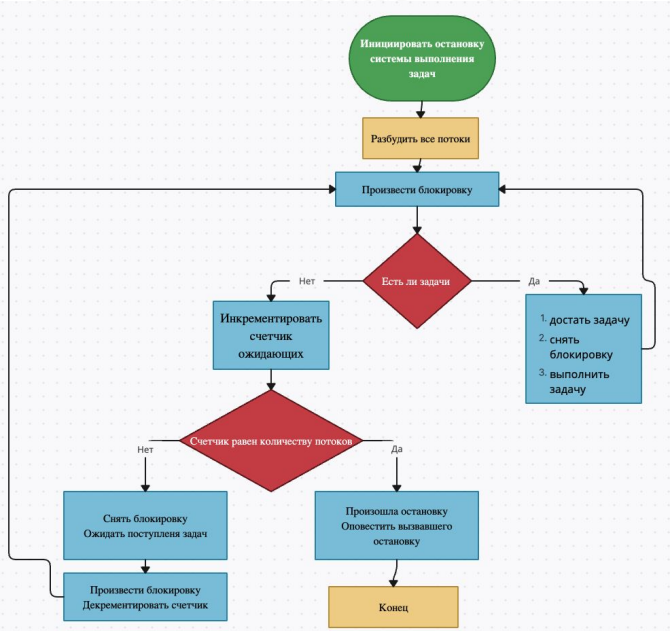

Рисунок 4 - остановка ThreadPool