Аннотация:

Система LuNA предназначена для создания параллельных программ численного моделирования на суперкомпьютерах, чтобы специалист мог создавать эффективные параллельные программы и при этом не обладать высокой квалификацией в области параллельного программирования. Но система обладает некоторыми недостатками, из-за которых производительность программ для LuNA уступает аналогичным программам, написанным с использованием низкоуровневых средств параллельного программирования. В частности, это связано с тем, что из-за особенностей реализации система недостаточно эффективно управляет памятью, а также предоставляет ограниченный набор инструментов для ручного управления памятью.

Данная работа посвящена разработке директивных средств управления памятью для системы LuNA, которые будут способны приблизить расход памяти LuNA программ к ее расходу программами, написанными с использованием низкоуровневых средств параллельного программирования.

Тема работы: Разработка и реализация средств директивного управления памятью в системе LuNA

Состав коллектива:

Студент: Кудрявцев Андрей Александрович, Факультет информационных технологий, кафедра параллельных вычислений, 4-й курс, группа 19203

Научный руководитель: Власенко Андрей Юрьевич, к.т.н, доц. каф. ПВ ФИТ НГУ, н.с. ИВМиМГ СО РАН

Соруководитель: Перепёлкин Владислав Александрович, ст. преп. каф. ПВ ФИТ НГУ, н.с. ИВМиМГ СО РАН

Научное содержание работы

Современное состояние проблемы и постановка задачи:

Управление памятью – это важный аспект разработки любого программного обеспечения. От этого зависит не только работоспособность программы, но и ее эффективность, так как из-за недостаточно хорошего использования памяти могут ухудшиться нефункциональные характеристики программы. Например, возможно увеличение времени выполнения или потребление памяти. Последнее может привести к замедлению и даже прекращению работы не только данной программы, но и всей системы в целом. Существуют различные подходы к управлению памятью. Одним из них является низкоуровневое ручное управление. Подобный способ предоставляет возможность достичь оптимальной (или близкой к оптимальной) скорости работы, но вместе с этим требует от

программиста высокой квалификации и больших трудозатрат. Другим подходом является автоматическое управление памятью, например, сборка мусора. Этот вариант обычно уступает первому в скорости выполнения, но выигрывает в удобстве и скорости разработки программ. Стоит отметить, что такой способ зачастую тяжелее реализовать. В качестве третьего подхода можно выделить ручное управление памятью на достаточно высоком уровне абстракции. Этот вариант не требует высокой квалификации для использования, имеет приемлемую сложность реализации, может приближаться по скорости к низкоуровневым средствам, а также имеет потенциал для автоматизации. К последнему способу относится директивное управление памятью, то есть управление памятью, основанное на прямых предписаниях системе.

Проблема эффективного управления памятью не имеет общего решения, но возможны решения в частных случаях. Поэтому целесообразно будет рассмотреть проблему на примере системы фрагментированного программирования LuNA.

Задача заключается в разработке директивных средств управления памятью для системы LuNA, которые будут способны приблизить расход памяти LuNA-программ к ее расходу программами, написанными с использованием низкоуровневых средств параллельного программирования.

Подробное описание работы, включая используемые алгоритмы:

В качестве решения предлагается расширить модель вычислений системы LuNA механизмом, который позволяет записывать новое значение, принадлежащее выходному ФД, в участок памяти, занимаемый значением входного ФД. Назовем такой механизм уничтожающим потреблением, входной ФД – поглощаемым, а выходной ФД – поглощающим. Тогда модель вычислений может выглядеть следующим образом (Рисунок 1):

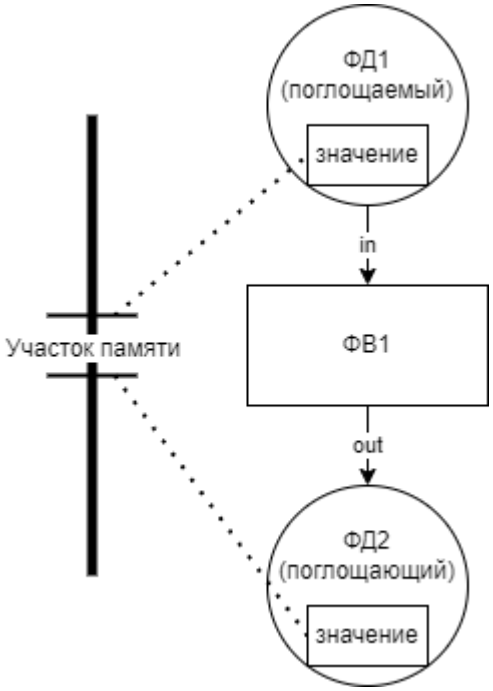


Рисунок 1 – Пример расширенной модели вычислений

Причем для каждого ФД должен существовать только один ФВ, в котором он является поглощаемым. Это необходимо, так как в противном случае каждое уничтожающее потребление одного и того же поглощаемого ФД может 16 приводить к неявной модификации поглощающих ФД из предыдущих уничтожающих потреблений. Поглощающий ФД одновременно является выходным, а значит, из-за принципа единственного присваивания ФД, для него всегда существует только один ФВ, в котором он поглощающий.

Помимо этого, предлагаемое решение предоставляет системе информацию о том, на каком этапе вычислений выполняется уничтожающее потребление и какие ФД в нем участвуют. Также решение предусматривает выполнение проверки того, что имеющаяся в системе информация об уничтожающем потреблении не противоречит реальному выполнению уничтожающего потребления.

Также предлагаемое решение не исключает возможность автоматического использования, так как ответственность за перенос буфера и генерацию системной информации в перспективе можно возложить на систему.

Предлагаемое решение было реализовано в системе LuNA с учетом особенностей ее реализации в несколько этапов:

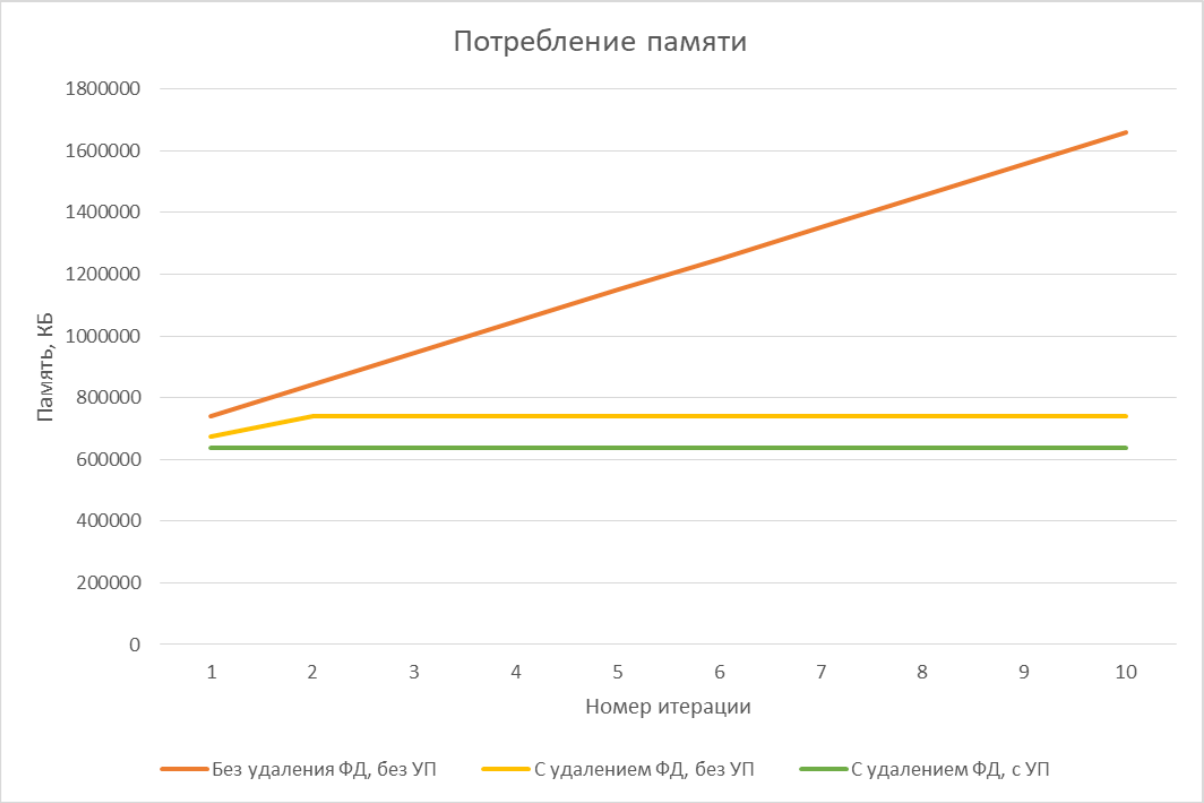
1. Расширение интерфейса класса DF.
2. Расширение грамматики языка LuNA.
3. Модификация транслятора системы LuNA.

Полученные результаты:

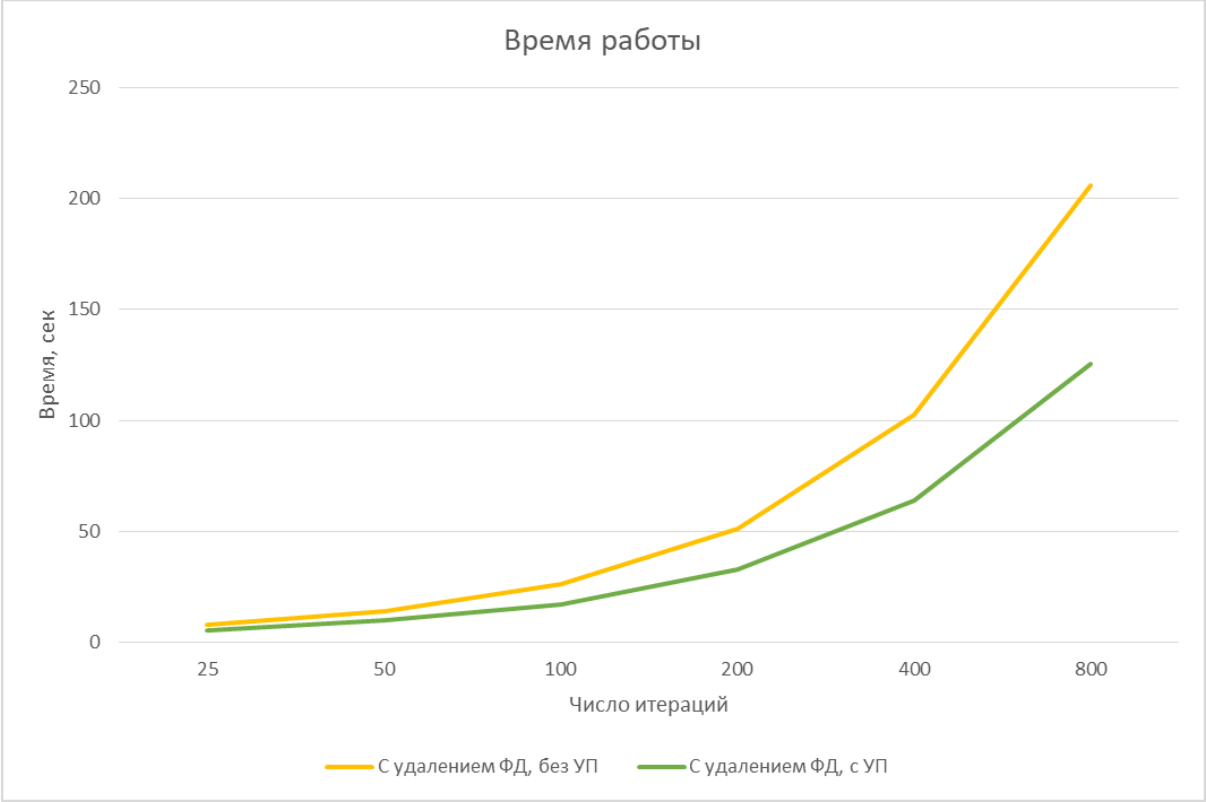
Реализация уничтожающего потребления была протестирована локально и на кластере. В результате была проверена работоспособность

средства и его влияние на нефункциональные характеристики программ. Для снятия нефункциональных характеристик использовались утилиты time и top. Были получены следующие результаты:

Локальный запуск тестовой программы:



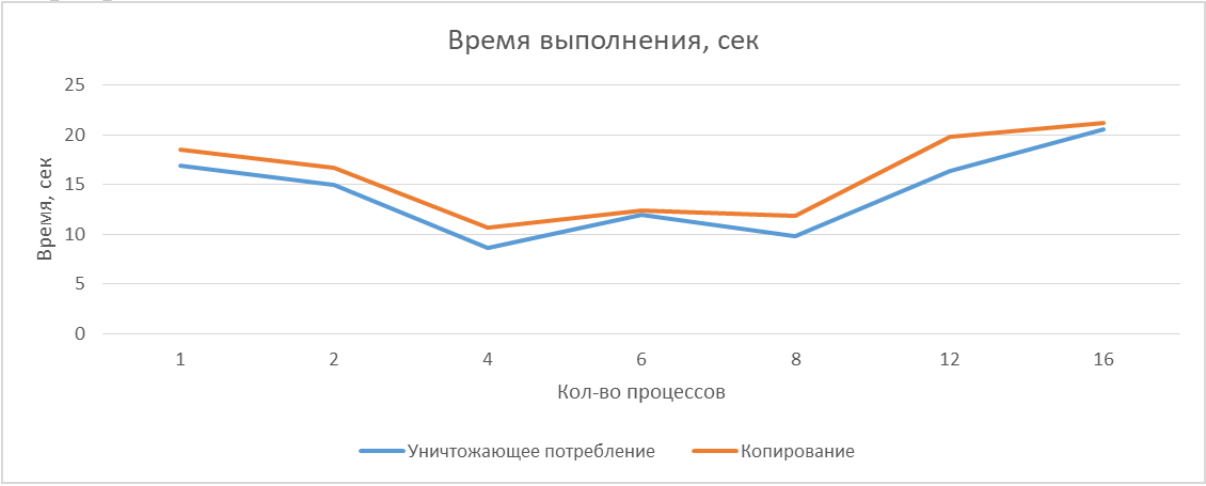
Без удаления фрагментов данных и без уничтожающего потребления потребление памяти закономерно постоянно растет. С удалением фрагментов данных и без уничтожающего потребления потребление памяти сохраняется на одном уровне, при этом память расходуется на данные исполнительной системы и 2 массива по 100МБ, т.к. команда top вызывается после выделения памяти. Таким образом, в определенный момент времени в памяти находятся два массива - со старыми и измененными данными. С удалением фрагментов данных и с уничтожающим потреблением потребление памяти на одном уровне, при этом память расходуется на данные исполнительной системы и один массив размером 100 МБ. В этом случае в каждый момент времени в памяти расположен только один массив.

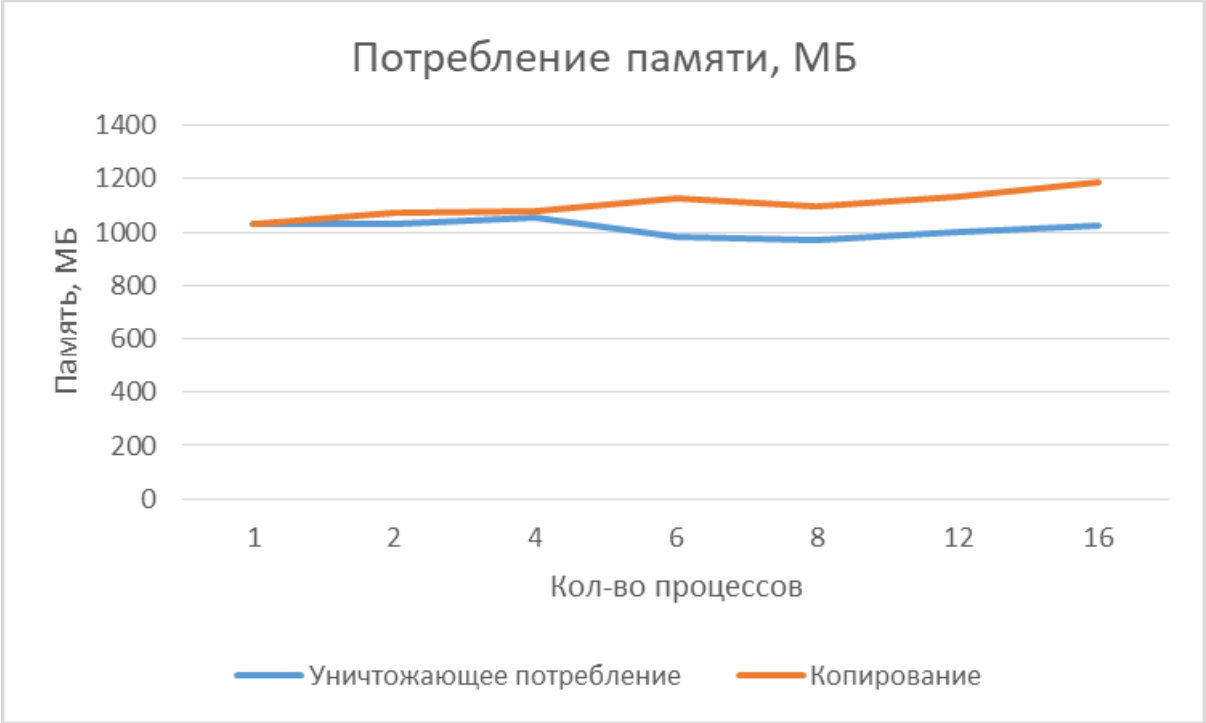


Таким образом, вариант с использованием уничтожающего потребления исполняется на 30-40% быстрее.

Запуск реальной задачи на кластере:

Была выбрана прикладная задача трехмерной фильтрации двухфазной жидкости при наличии скважин. Математическая модель данной задачи представляет собой СЛАУ, решаемую с помощью итерационного метода сопряженных градиентов. Из-за применения итерационного метода использование уничтожающего потребления должно положительно сказаться на нефункциональных характеристиках данной программы.





Эффект от использования кластера:

С помощью кластера удалось провести тестирование реализованного средства на прикладной задаче в среде с распределенной памятью, а также экспериментально показать, что использование реализованного средства положительно влияет на нефункциональные характеристики программ, написанных для системы LuNA.

Перечень публикаций, содержащих результат работы:

1. А.А. Кудрявцев, В.Э. Малышкин, Ю.Ю. Нуштаев, В.А. Перепёлкин, В.А. Спирин Эффективная фрагментированная реализация краевой задачи фильтрации двухфазной жидкости //Проблемы информатики. – 2023 (Отправлена в журнал).

2. Кудрявцев А. А. Разработка и реализация средств директивного управления памятью в системе LuNA / Кудрявцев А. А. //Информационные технологии : Материалы 61-й Междунар. науч. студ. конф. 17–26 апреля 2023 г. / Новосиб. гос. ун-т. — Новосибирск : ИПЦ НГУ, 2023 (Принято в печать).