МГУ им. М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Автоматизации Систем Вычислительных Комплексов

Задание по курсу "Распределённые системы"

Выполнила:
Бурдюгова Мария Витальевна
421 группа

Москва, 2024 год

Содержание

1 Постановка задач

Доработать MPI-программу, реализованную в рамках курса “Суперкомпьютеры и параллельная обработка данных”. Используя параллельный ввод-вывод (MPI-IO), добавить контрольные точки для продолжения работы программы в случае сбоя. Реализовать один из 3-х сценариев работы после сбоя:

а) продолжить работу программы только на “исправных” процессах;

б) вместо процессов, вышедших из строя, создать новые MPI-процессы, которые необходимо использовать для продолжения расчетов;

в) при запуске программы на счет сразу запустить некоторое дополнительное количество MPI-процессов, которые использовать в случае сбоя.

Подготовить отчет о выполнении задания, включающий описание алгоритма, детали реализации, а также временные оценки работы алгоритма.

2 Алгоритм

Реализованный сценарий:

а) продолжить работу программы только на “исправных” процессах.

В коде искусственно создается сбой с помощью `raise(SIGKILL)`,что позволяет протестировать механизм восстановления.

Полный код и команды для компиляции и запуска можно найти по ссылке:

https://github.com/murlinmurlo/parallel-programming/tree/main/distributed

2.1 Распределение данных

Матрица делится между процессами. Каждый процесс обрабатывает свою часть данных, определяемую диапазоном строк (start и end). Дополнительные процессы остаются в резерве и не участвуют в вычислениях.

2.2 Контрольные точки

На каждой итерации данные текущего состояния (локальная часть матрицы `local_A`) сохраняются в файл `checkpoint.dat` с использованием MPI-IO. Каждый процесс записывает свою часть данных в файл, учитывая смещение (offset), чтобы избежать конфликтов записи.

2.3 Обработка сбоев

Если процесс завершился сбоем, вызывается пользовательский обработчик ошибок `handler`. Обработчик использует функцию `MPIX_Comm_shrink` для создания нового коммуникатора, исключающего сбойные процессы. После восстановления коммуникатора работа перераспределяется между оставшимися процессами.

2.4 Восстановление из контрольной точки

После сбоя данные восстанавливаются из файла `checkpoint.dat` с использованием MPI-IO (функция `MPI_File_read_at`). Программа возвращается к последней сохраненной итерации с помощью `longjmp`.

2.5 Основной цикл вычислений

Выполняются численные расчеты с использованием метода релаксации. На каждой итерации:

- Обновляются значения матрицы (функция `relax`).
- Вычисляется ошибка (функция `resid`).
- Проверяется условие завершения (достижение максимального числа итераций или минимальной ошибки).

2.6 Завершение работы

После завершения вычислений результаты проверяются функцией `verify`.