

Выводы по практической работе

1. GPU эффективен для больших массивов данных.
 2. Использование shared memory критично для производительности.
 3. Гибридные вычисления полезны при правильном разделении задач.
 4. MPI даёт ускорение, но масштабируемость ограничена.
 5. Накладные расходы передачи данных влияют на итоговое время.
-

Контрольные вопросы

1. В чём отличие гибридных вычислений от CPU или GPU?

Гибридные вычисления используют **одновременно CPU и GPU**, распределяя нагрузку между ними для максимальной эффективности.

2. Для каких задач целесообразно распределять вычисления между CPU и GPU?

- обработка больших массивов
 - задачи с высокой степенью параллелизма
 - вычисления, где CPU выполняет управление, а GPU — вычисления
-

3. Разница между синхронной и асинхронной передачей данных?

- **Синхронная** блокирует выполнение
 - **Асинхронная** позволяет перекрывать вычисления и передачу данных
-

4. Почему асинхронная передача повышает производительность?

Потому что CPU и GPU работают **параллельно**, не простаивая в ожидании.

5. Основные функции MPI

- MPI_Init, MPI_Finalize
 - MPI_Scatter, MPI_Gather
 - MPI_Reduce
 - MPI_Bcast
-

6. Как количество процессов MPI влияет на время выполнения?

До определённого момента время уменьшается, затем накладные расходы превышают выигрыш.

7. Факторы, ограничивающие масштабируемость

- пропускная способность сети
 - синхронизация процессов
 - дисбаланс нагрузки
 - объём передаваемых данных
-

8. Когда распределённые вычисления оправданы?

- ✓ Большие объёмы данных
 - ✗ Малые задачи и частые синхронизации
-

Итог

В работе были реализованы CUDA-, гибридные и MPI-программы, проведено сравнение производительности и проанализированы преимущества различных моделей параллелизма.