

Варианты расчётно-графических заданий по курсу «программирование графических процессоров»

Вариант на оценку «удовлетворительно»:

1. Провести анализ производительности программ, реализующих алгоритмы линейной алгебры с использованием библиотек Thrust, cuBLAS и «сырого» кода на CUDA C.

Варианты на оценки «хорошо» и «отлично»:

Во всех вариантах заданий для получения оценки «хорошо» требуется реализовать алгоритм решения задачи оптимизации в зависимости от варианта. Для получения оценки «отлично» нужно применить реализованный алгоритм оптимизации к задаче обучения логистической регрессии. Теоретический материал по логистической регрессии и алгоритмам оптимизации находятся в директории с этим файлом.

Для тестирования алгоритмов оптимизации можно воспользоваться следующими функциями:

Функция	Диапазон	Глобальный минимум
$f(X) = -\sum_{i=1}^6 (-x_i \sin(\sqrt{ x_i }))$	$-500 \leq x_i \leq 500$	$f(X) = 2513.9$ $X = \{420.9, 420.9, 420.9, 420.9, 420.9, 420.9\}$
$f(X) = \sum_{i=1}^6 x_i^2$	$-5.12 \leq x_i \leq 5.12$	$f(X) = 0$ $X = \{0, 0, 0, 0, 0, 0\}$
$f(X) = (x_1 - x_2)^2 + ((x_1 + x_2 - 10)/3)^2$	$0 \leq x_i \leq 10$	$f(X) = 0$ $X = \{1, 1\}$

Каждый реализуемый алгоритм нужно распараллелить на GPU при помощи CUDA C. Разрешается использовать библиотеки cuBLAS, cuRAND и Thrust.

Программа на оценку «хорошо» должна выводить для каждой функции из таблицы выше среднеквадратическую ошибку 1000 экспериментов на каждую функцию.

Программа на оценку «отлично» должна выводить процент правильно классифицированных объектов. В файле с исходными данными 100000 строк по 8 значений, каждая строка – вектор признаков объекта, а последнее значение строки – класс объекта. Исходные данные должны делиться случайным образом на две части 1 к 8. Большая часть будет служить обучающей выборкой, меньшая тестируемым множеством, для объектов которого производится классификация. Для тестирования нужно провести 1000 экспериментов и взять среднее значение процента правильно классифицированных объектов.

Отчёт по РГЗ должен содержать постановку задачи, схему разработанных параллельных алгоритмов, результаты работы программы, анализ эффективности распараллеливания алгоритмов на GPU при помощи профилировщика, код программы.

Оцениваться будет точность реализованных алгоритмов и эффективность реализации.

Номер варианта выбирается по формуле $N \bmod 6 + 1$, где N – порядковый номер по журналу.

Варианты заданий:

1. Реализовать алгоритм оптимизации методом роя частиц. Обучить логистическую регрессию путём минимизации функции эмпирического риска. data1.txt – файл с исходными данными.
2. Реализовать алгоритм оптимизации методом искусственной пчелиной колонии. Обучить логистическую регрессию путём минимизации функции эмпирического риска. data2.txt – файл с исходными данными.
3. Реализовать алгоритм оптимизации методом дифференциальной эволюции. Обучить логистическую регрессию путём минимизации функции эмпирического риска. data3.txt – файл с исходными данными.
4. Реализовать алгоритм оптимизации методом роя частиц. Обучить логистическую регрессию путём максимизации функции правдоподобия. data4.txt – файл с исходными данными.
5. Реализовать алгоритм оптимизации методом искусственной пчелиной колонии. Обучить логистическую регрессию путём максимизации функции правдоподобия. data5.txt – файл с исходными данными.
6. Реализовать алгоритм оптимизации методом дифференциальной эволюции. Обучить логистическую регрессию путём максимизации функции правдоподобия. data6.txt – файл с исходными данными.