

Flow graph

Реализовать с использованием графовой модели вычислений следующую задачу, разбив её на части и обеспечив максимальную потенциальную утилизацию ресурсов (изображение == матрица $M \times N$):

1. На вход подаются генерированные случайным образом изображения
2. Различные алгоритмы ищут || на изображении:
 - Набор максимальных элементов (максимальная яркость)
 - Набор минимальных элементов (минимальная яркость)
 - Набор элементов, равных заданному в командной строке значению (0-255)
3. По результатам предыдущего шага на изображении выделяются все найденные элементы (например, квадрат с максимальной яркостью вокруг точки)
4. Результат предыдущего шага передаётся на:
 - Расчёт обратного изображения (инверсия яркости)
 - Расчёт средней яркости изображения
5. Результат расчёта средней яркости выводится в файл в произвольном формате, если указан соответствующий флаг запуска приложения
6. Результат расчёта обратного изображения никуда не идёт

Предполагается использовать:

- broadcast_node
- join_node
- function_node
- limiter_node
- ...

Приложение принимает в качестве параметров:

1. “-b 123”: интересующее значение яркости, для шага № 2
2. “-l 5”: предел одновременно обрабатываемых приложением изображений
3. “-f log.txt”: имя файла журнала яркостей

Адекватные модификации задания (периодический вывод текущего изображения в консоль после расчёта обратного, работа с реальными изображениями и более реальными операциями...) допускаются. При наличии у студента задач, подходящих для применения flow graph - по согласованию через почту можно реализовать их вместо стандартного задания.

Ограничения на язык: имеющий framework, поддерживающий графовую модель вычислений, аналогичную Intel TBB Flow Graph

From:
<http://wiki.osll.ru/> - Open Source & Linux Lab

Permanent link:
http://wiki.osll.ru/doku.php/courses:high_performance_computing:tbb_flow_graph

Last update: **2016/11/11 07:15**

