

A Thing That Could Be a Library

GROUP AWESOME!



AALBORG UNIVERSITY STUDENT REPORT

Jonathan Hastrup

Morten Mandrup Hansen



AALBORG UNIVERSITY
STUDENT REPORT

Department of Computer Science
Selma Lagerlöfs Vej 300
9220 Aalborg Ø

Title:

A Comparative Study of Programming
Languages for the GPU

Theme:

Programming Technology

Project period:

01/09/2017 -
05/01/2018

Project group:

DPW???F18

Members:

Jonathan Hastrup
Morten Mandrup Hansen

Supervisor:

Lone Leth Thomsen

No. of Pages: ??

No. of Appendix Pages: ??

Total no. of pages: ??+??

Completed: ??/??/2018

Abstract:

We did all the things

Reading Instructions

Left to right, top down. Pretty straight forward.

Definitions

This is a list of terms heavily used in this report, with their definitions.

STL The C++ Standard Template Library

1	Introduction	5
1.1	Motivation	5
1.2	Related Works	5
1.3	Problem Statement	6
2	Design	7
3	Implementation	8
4	Evaluation	9
4.1	Test	9
4.2	Performance	9
5	Reflection	10
6	Conclusion	11
7	Future Work	12
I	Appendix	13

This is a project, yay

1.1 Motivation

Why is this interesting

1.2 Related Works

/ meta here */*

1.2.1 Bolt

/ meta here */*

Goals

Bolt is designed to provide high performance library implementations for common algorithms, following the structure of STL. It is intended to make heterogeneous development easier.

It is designed to provide an application that can execute on either a CPU or any OpenCL capable unit.

Implementation

Bolt is a...

Bolt targets...

Programming Model

Bolt is modeled on STL and as such, follows the model of calling functions with iterators as arguments to instruct where input, and output is located.

An example is shown in Listing 1.1, where we sort a device vector. This is identical to the method shown in Listing 1.2, where a `std::vector` is sorted with `std::sort`.

```
1 //vector construction
2 bolt::cl::device_vector<int> input(1024);
3
4 //vector fill omitted
5
```

```
6 //inplace sort
7 bolt::cl::sort(input.begin(), input.end());
```

Listing 1.1: Bolt sort example

```
1 //vector construction
2 std::vector<int> input(1024);
3
4 //vector fill omitted
5
6 //inplace sort
7 std::sort(input.begin(), input.end());
```

Listing 1.2: STL sort example

Cognitive Dimensions of Notations

1.3 Problem Statement

What we exactly want to solve

2. Design

We make decisions, not war!

3. Implementation

We have all this pretty code to show, also that design decision back then, we hacked it up like this!

4. Evaluation

Here we do stuff like, saying it works, and it goes THAT fast!

4.1 Test

We actually did some tests, and respected the nice feedback from our previous examn.

4.2 Performance

Oh yeah? this one is actually THIS fast!

5. Reflection

Actually, starting with weird dummy texts in files, was a pretty good idea!

6. Conclusion

We did every single thing!

7. Future Work

Uhm, maybe, now we just need to make this project real!

Part I

Appendix