Musketeer

All for one, One for All
So many systems...

- Hadoop, Metis, Dryad, Spark, Naiad, CIEL, Giraph, PowerGraph, GraphChil, X-Stream... list goes on.
- Often high-level languages are used like Hive and Lindi.
How to choose the best one?

- Depends on workflow, input data size and scale of parallelism available.
- Very cumbersome to write native code for each.
- What if there was a way to write high-level code and then pick which system to use...?
Enter Musketeer!

- Musketeer is a project to port high-level front ends like Hive and Lindi to any backend.
- Being built here in the Cambridge computer lab!
- Takes a job written in e.g. Hive and generates an intermediate workflow DAG.
- This DAG is then executed on the backend system of choice.
Musketeer has comparable performance.

- Obviously not as efficient as a native implementation, but much better than alternatives.
Currently no support for Apache Giraph

- I plan to add Apache Giraph to collection of backends.
- Ionel Gog wants to implement this but is working on something else.
- He said he’ll be there if I have questions.
- Might take me until January 14th!
- Sounds fun.
Why I think I’ll be able to do it.

- Intermediate DAGs are already generated from high level code.
- Other vertex-centric systems have already been implemented. PowerGraph + GraphChi.
Plan Bs

- Further Benchmarking of Musketeer.
- Triangle counting has not yet been done.
- I will implement a triangle counting algorithm in native GraphLab, GraphChi and Spark and benchmark with a Musketeer implementation.
Current progress

● Playing around with Apache Hadoop + Hive.
● Playing around with HiveIO - hive front-end for Giraph.
● Possible to write Hive and run on Hadoop, but not possible to then run immediately on a different system.
Questions...