Ad-hoc Big-Data Analysis with Lua

And LuaJIT

Alexander Gladysh <ag@logiceditor.com>
@agladysh

Lua Workshop 2015
Stockholm

Outline

Alexander Gladysh

- CTO, co-owner at LogicEditor
- In löve with Lua since 2005

The Problem

- You have a big dataset to analyze
- that makes casual analysis tools explode or be too slow
- and you don't have resources to set up and maintain (or pay for) Hadoop, Google Big Query etc.
- but you have some processing power available.

Goal

- Pre-process the data so it can be handled by R or Excel or your favorite analytics tool (or Lua!).
- If the data is dynamic, then *learn to* pre-process it and build a data processing pipeline (which is outside of the scope of this talk).

An approach

- Use Lua!
- And (semi-)standard tools, available on Linux.
- Go minimalistic while exploring,
- Then move to an industrial solution that fits your newly understood requirements
- Or roll your own ecosystem ;-)

Start small!

- Always run your scripts on small representative excerpts from your datasets, not only while developing them locally, but on actual data-processing nodes too.
- Saves time and helps you learn the bottlenecks.
- Sometimes large run still blows in your face though.

Discipline!

- Many moving parts, large turn-around times, hard to keep tabs.
- Keep journal: Write down what you run and what time it took.
- Store actual versions of your scripts in a source control system.
- Don't forget to sanity-check the results you get!

LuaJIT?

- Up to a point:
- 2.1 helps to speed things up,
- FFI bogs down development speed.
- Go plain Lua first (run it with LuaJIT),
- then roll your own ecosystem as needed ;-)

Hardware?

- As usual, more is better: Cores, cache, memory speed and size, HDD speeds, networking speeds...
- But even a modest VM (or several) can be helpful.
- Your fancy gaming laptop is good too ;-)

OS

- Linux (Ubuntu) Server.
- Approach will, of course, work for other setups.

Data layout

- Ideally, have data copies on each processing node, using identical layouts.
- Fast network should work too.

Data format

- TODO

The Tools

- parallel
- sort, uniq, grep
- cut, join, comm
- pv
- compression utilities
- LuaJIT

Why Lua?

Perl, AWK are traditional alternatives to Lua, but, if you're not very disciplined and experienced, they are much less maintenable.

Pipeline

- TODO

Advice

Pre-sort everything!

Advice

Monitor resource utilization

Compression

- gzip: default, bad
- lxc: fast, large files
- pigz: fast, parallelizable
- xz: good compression, slow
- be on lookout for new formats!

Heavy-weights

TODO Code

```
TODO
```

Questions?

Alexander Gladysh, ag@logiceditor.com