Map Reduce for Algorithmic Trading

Akshaan Kakar, Alex Berard
Dept. of Computer Science
Columbia University
Email: ak3808@columbia.edu, alice.berard@columbia.edu

***Abstract*—Algorithmic Trading is an extremely competitive sector of financial markets. Developing trading algorithms involves the pivotal step of backtesting, where the performance of the algorithm is validated against large amounts of historical securities pricing data. These time series require large amounts of computational resources for storage, processing and visualization. In this paper, we describe the implementation of an algorithmic trading backtest engine that uses "Big Data" tools as a backbone to backtests for trading algorithms in an efficient and scalable fashion. We also show how this system can be extended to support live trading as well as a plethora of other features.**

***Keywords*-Algorithms, Finance, Trading, Big Data, Hadoop, Spark, Backtesting**

I. Introduction

With the computerization of order flows in financial markets, traders were afforded the ability to have computers place buy and sell orders on securities according to predefined strategies. The advent of such Algorithmic Trading strategies eventually developed into an extremely competitive sector of financial markets. The development of trading algorithms is now a formalized process that involves intricate research and mathematical models. Although a significant portion of this development effort is invested in the underlying mathematical theory, backtesting of algorithms is the most pivotal step in the process. Backtesting is the phase in which a trading algorithm is validated against large amounts of historical pricing data. Since trading algorithms may place orders extremely frequently and involve multiple securities in a single portfolio, large amounts of data must be stored, read and processed in order to run these tests. Moreover, the result must be visualized so that developers can gauge the performance of their algorithms quickly and conveniently.

There is an abundance of well developed and well maintained tools which have been built expressly to grapple with the large amounts of data that have now become commonplace. These tools leverage novel algorithmic and system level techniques to to deal with the space and time bottlenecks that come with large datasets. Since the problem of backtesting algorithmic strategies is inherently a 'Big Data' shaped problem, we decided to use a combination of such tools in order to to build an efficient and flexible system for the task.

II. Related Works

We modeled our implementation of the backtesting engine with some other tools and works in mind. One of the prominent platforms that we sought to emulate is Quantopian (CITE). The Quantopian platform allows in-browser implementation of trading algorithms in Python. It also allows the use of internal APIs as well as most external python modules for statistics, numerical algorithms etc. Although we have not extended our system to allow in-browser coding, we have also used python as the language of choice for trading algorithm implementation. Python is readable, flexible and popular, making it a good choice for algorithmic trading; a field where not everyone is well versed with computer programming. We also sought to emulate the charting and metric calculations performed by the Quantopian platform. The benchmark as well as the returns time series are plotted to provide a lucid understanding of the algorithm performance, without confusing the user with unnecessary detail. Also, important metrics such as the Sharpe Ratio, Maximum drawdown etc. are provided to the user for further insight.

III. System Overview

The main objectives in our implementation were to keep the platform efficient and easily extensible. In order to achieve these goals, we partitioned the system into independently functioning subsystems which we then integrated to realize the final backtesting engine. For the purpose of demonstration, we utilized the free historical pricing data from QuantCode (CITE). The dataset as well as the subsystems are described in detail in the following subsections.

A. QuantQuote dataset

We decided to use the QuantQuote free historical stock price data for the purpose of demonstration. This data consists of daily stock tick data for the 500 symbols that are listed on the Standard and Poor's 500 Index from 1998 to present. Although we used the dataset throughout the design and testing process, we structured the backtesting engine such that it is dataset agnostic. More specifically, we designed our subsystems so that they do not impose any tight constraints on input data formatting. Further descriptions of input data specifications are provided in the following subsection.

B. *Hadoop Data Warehouse*

Apache Hadoop (CITE) is a distributed framework that is designed for the storage and processing of large datasets. Hadoop is especially well suited to read-only, batch accessing of large amounts of data. Our system only requires the reading of financial time series data, without frequent writes or editing, making Hadoop a perfect choice for the data warehouse subsystem. We stored the finance symbol pricing data in the Hadoop Distributed File System (HDFS), which can be configured to run on any number of machines without a single point of failure. We structured the file system so that the time series data for each symbol were stored in a separate file. This simple, flat structure makes accessing the data extremely simple, since for a particular symbol, only a single file must be accessed. Each file was named after the symbol, to facilitate programatic access using symbol names directly, without the need for any lookup or translation. In the case that the data repository needs to be updated with newer pricing data, only a single file need be written to. This simple organization does not compromise on efficiency since Hadoop is designed to work well with a relatively small number of very large files, as opposed to small fragments of a larger dataset. A directory listing showing the file structure is shown in Figure 1.

Within each file, the time series were stored in a comma separated value (CSV) format, with time stamps and price values as the fields in each row. The CSV format is simple and widely used, making our system largely data agnostic. Any data in CSV format with the appropriate fields can be used with our data warehouse system since our engine does not impose and other restrictions on data formatting.

C. *Spark Algorithm Processing Engine*

Apache Spark is a tool that was developed for processing general large-scale data efficiently (CITE). Spark come with support for multiple languages and platforms and an also integrate seamlessly with the Hadoop Distributed File System. We chose Spark to be the workhorse for the main trading algorithm processing. In order to leverage the features offered by Spark, we used pySpark, which is the Spark API in the Python scripting language. Our algorithm processing engine is responsible for detecting all the trading symbols that are involved in a user-defined trading strategy, and to retrieve the corresponding data from the Hadoop data store. After retrieving the required time series, the engine then runs the rules specified in the strategy against these series and computes the basic returns for the overall portfolio. Once the returns have been computed for the entire data, further portfolio performance metrics are computed. These include the mean return, standard deviation, maximum drawdown and Sharpe Ratio. The final returns time series and the metrics are then written to temporary output files on disk for use further downstream in the processing pipeline.

D. *Javascript Server Back End*

We sought to build a simple web app which could concisely present the results of algorithm backtesting to the user. In order to do this, we implemented a very basic server using the node.js JavaScript runtime.

E. *Visualization*

In order to visualize the performance of user defined trading algorithms viz. the benchmark, we built a JavaScript front end for our web application. In order to produce detailed and informative plots, we utilized the Highcharts and Highstock charting libraries.
The schematic in Figure 2. shows an outline of the system design.

IV. Algorithm (you can change this section name)

Show algorithm and tools you have used to solve your problem.

V. Software Package Description

Describe the software package that is going to be open sourced. Show some screen shots of how user use it and or UI.

VI. Experiment Results

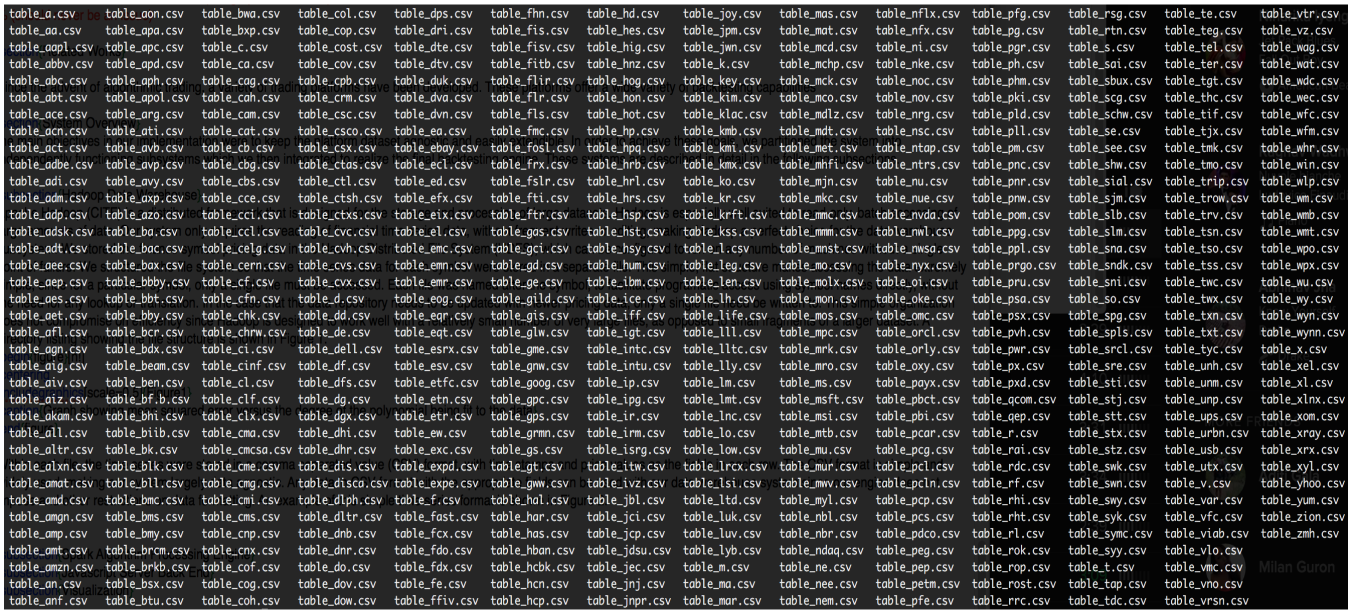
Describe the experiment results of your algorithm. Show how did you evaluate the performance of your algorithm.

VII. Conclusion

Drew a conclusion of your project, describe the contributions of each team member in percentage and discuss future works.

Acknowledgment

The authors would like to thank... more thanks here



```
table_a.csv     table_aon.csv   table_bwa.csv   table_col.csv   table_dps.csv   table_fhn.csv   table_hd.csv    table_joy.csv   table_mas.csv   table_nflx.csv  table_pfg.csv   table_rsg.csv   table_te.csv    table_vtr.csv
table_aa.csv    table_apa.csv   table_bxp.csv   table_cop.csv   table_dri.csv   table_fis.csv   table_hes.csv   table_jpm.csv   table_mat.csv   table_nfx.csv   table_pg.csv    table_rtn.csv   table_teg.csv   table_vz.csv
table_aapl.csv  table_apc.csv   table_c.csv     table_cost.csv  table_dte.csv   table_fisv.csv  table_hig.csv   table_jwn.csv   table_mcd.csv   table_ni.csv    table_pgr.csv   table_s.csv     table_tel.csv   table_wag.csv
table_abbv.csv  table_apd.csv   table_ca.csv    table_cov.csv   table_dtv.csv   table_fitb.csv  table_hnz.csv   table_k.csv     table_mchp.csv  table_nke.csv   table_ph.csv    table_sai.csv   table_ter.csv   table_wat.csv
table_abc.csv   table_aph.csv   table_cag.csv   table_cpb.csv   table_duk.csv   table_flir.csv  table_hog.csv   table_key.csv   table_mck.csv   table_noc.csv   table_phm.csv   table_sbux.csv  table_tgt.csv   table_wdc.csv
table_abt.csv   table_apol.csv  table_cah.csv   table_crm.csv   table_dva.csv   table_flr.csv   table_hon.csv   table_kim.csv   table_mco.csv   table_nov.csv   table_pki.csv   table_scg.csv   table_thc.csv   table_wec.csv
table_ace.csv   table_arg.csv   table_cam.csv   table_csc.csv   table_dvn.csv   table_fls.csv   table_hot.csv   table_klac.csv  table_mdlz.csv  table_nrg.csv   table_pld.csv   table_schw.csv  table_tif.csv   table_wfc.csv
table_acn.csv   table_ati.csv   table_cat.csv   table_csco.csv  table_ea.csv    table_fmc.csv   table_hp.csv    table_kmb.csv   table_mdt.csv   table_nsc.csv   table_pll.csv   table_se.csv    table_tjx.csv   table_wfm.csv
table_act.csv   table_avb.csv   table_cb.csv    table_csx.csv   table_ebay.csv  table_fosl.csv  table_hpq.csv   table_kmi.csv   table_met.csv   table_ntap.csv  table_pm.csv    table_see.csv   table_tmk.csv   table_whr.csv
table_adbe.csv  table_avp.csv   table_cbg.csv   table_ctas.csv  table_ecl.csv   table_frx.csv   table_hrb.csv   table_kmx.csv   table_mhfi.csv  table_ntrs.csv  table_pnc.csv   table_shw.csv   table_tmo.csv   table_win.csv
table_adi.csv   table_avy.csv   table_cbs.csv   table_ctl.csv   table_ed.csv    table_fslr.csv  table_hrl.csv   table_ko.csv    table_mjn.csv   table_nu.csv    table_pnr.csv   table_sial.csv  table_trip.csv  table_wlp.csv
table_adm.csv   table_axp.csv   table_cce.csv   table_ctsh.csv  table_efx.csv   table_fti.csv   table_hrs.csv   table_kr.csv    table_mkc.csv   table_nue.csv   table_pnw.csv   table_sjm.csv   table_trow.csv  table_wm.csv
table_adp.csv   table_azo.csv   table_cci.csv   table_ctxs.csv  table_eix.csv   table_ftr.csv   table_hsp.csv   table_krft.csv  table_mmc.csv   table_nvda.csv  table_pom.csv   table_slb.csv   table_trv.csv   table_wmb.csv
table_adsk.csv  table_ba.csv    table_ccl.csv   table_cvc.csv   table_el.csv    table_gas.csv   table_hst.csv   table_kss.csv   table_mmm.csv   table_nwl.csv   table_ppg.csv   table_slm.csv   table_tsn.csv   table_wmt.csv
table_adt.csv   table_bac.csv   table_celg.csv  table_cvh.csv   table_emc.csv   table_gci.csv   table_hsy.csv   table_l.csv     table_mnst.csv  table_nwsa.csv  table_ppl.csv   table_sna.csv   table_tso.csv   table_wpo.csv
table_aee.csv   table_bax.csv   table_cern.csv  table_cvs.csv   table_emn.csv   table_gd.csv    table_hum.csv   table_leg.csv   table_mo.csv    table_nyx.csv   table_prgo.csv  table_sndk.csv  table_tss.csv   table_wpx.csv
table_aep.csv   table_bbby.csv  table_cf.csv    table_cvx.csv   table_emr.csv   table_ge.csv    table_ibm.csv   table_len.csv   table_molx.csv  table_oi.csv    table_pru.csv   table_sni.csv   table_twc.csv   table_wu.csv
table_aes.csv   table_bbt.csv   table_cfn.csv   table_d.csv     table_eog.csv   table_gild.csv  table_ice.csv   table_lh.csv    table_mon.csv   table_oke.csv   table_psa.csv   table_so.csv    table_twx.csv   table_wy.csv
table_aet.csv   table_bby.csv   table_chk.csv   table_dd.csv    table_eqr.csv   table_gis.csv   table_iff.csv   table_life.csv  table_mos.csv   table_omc.csv   table_psx.csv   table_spg.csv   table_txn.csv   table_wyn.csv
table_afl.csv   table_bcr.csv   table_chrw.csv  table_de.csv    table_eqt.csv   table_glw.csv   table_igt.csv   table_lll.csv   table_mpc.csv   table_orcl.csv  table_pvh.csv   table_spls.csv  table_txt.csv   table_wynn.csv
table_agn.csv   table_bdx.csv   table_ci.csv    table_dell.csv  table_esrx.csv  table_gme.csv   table_intc.csv  table_lltc.csv  table_mrk.csv   table_orly.csv  table_pwr.csv   table_srcl.csv  table_tyc.csv   table_x.csv
table_aig.csv   table_beam.csv  table_cinf.csv  table_df.csv    table_esv.csv   table_gnw.csv   table_intu.csv  table_lly.csv   table_mro.csv   table_oxy.csv   table_px.csv    table_sre.csv   table_unh.csv   table_xel.csv
table_aiv.csv   table_ben.csv   table_cl.csv    table_dfs.csv   table_etfc.csv  table_goog.csv  table_ip.csv    table_lm.csv    table_ms.csv    table_payx.csv  table_pxd.csv   table_sti.csv   table_unm.csv   table_xl.csv
table_aiz.csv   table_bf.b.csv  table_clf.csv   table_dg.csv    table_etn.csv   table_gpc.csv   table_ipg.csv   table_lmt.csv   table_msft.csv  table_pbct.csv  table_qcom.csv  table_stj.csv   table_unp.csv   table_xlnx.csv
table_akam.csv  table_bhi.csv   table_clx.csv   table_dgx.csv   table_etr.csv   table_gps.csv   table_ir.csv    table_lnc.csv   table_msi.csv   table_pbi.csv   table_qep.csv   table_stt.csv   table_ups.csv   table_xom.csv
table_all.csv   table_biib.csv  table_cma.csv   table_dhi.csv   table_ew.csv    table_grmn.csv  table_irm.csv   table_lo.csv    table_mtb.csv   table_pcar.csv  table_r.csv     table_stx.csv   table_urbn.csv  table_xray.csv
table_altr.csv  table_bk.csv    table_cmcsa.csv table_dhr.csv   table_exc.csv   table_gs.csv    table_isrg.csv  table_low.csv   table_mu.csv    table_pcg.csv   table_rai.csv   table_stz.csv   table_usb.csv   table_xrx.csv
table_alxn.csv  table_blk.csv   table_cme.csv   table_dis.csv   table_expd.csv  table_gt.csv    table_itw.csv   table_lrcx.csv  table_mur.csv   table_pcl.csv   table_rdc.csv   table_swk.csv   table_utx.csv   table_xyl.csv
table_amat.csv  table_bll.csv   table_cmg.csv   table_disca.csv table_expe.csv  table_gww.csv   table_ivz.csv   table_lsi.csv   table_mwv.csv   table_pcln.csv  table_rf.csv    table_swn.csv   table_v.csv     table_yhoo.csv
table_amd.csv   table_bmc.csv   table_cmi.csv   table_dlph.csv  table_f.csv     table_hal.csv   table_jbl.csv   table_ltd.csv   table_myl.csv   table_pcp.csv   table_rhi.csv   table_swy.csv   table_var.csv   table_yum.csv
table_amgn.csv  table_bms.csv   table_cms.csv   table_dltr.csv  table_fast.csv  table_har.csv   table_jci.csv   table_luk.csv   table_nbl.csv   table_pcs.csv   table_rht.csv   table_syk.csv   table_vfc.csv   table_zion.csv
table_amp.csv   table_bmy.csv   table_cnp.csv   table_dnb.csv   table_fcx.csv   table_has.csv   table_jcp.csv   table_luv.csv   table_nbr.csv   table_pdco.csv  table_rl.csv    table_symc.csv  table_viab.csv  table_zmh.csv
table_amt.csv   table_brcm.csv  table_cnx.csv   table_dnr.csv   table_fdo.csv   table_hban.csv  table_jdsu.csv  table_lyb.csv   table_ndaq.csv  table_peg.csv   table_rok.csv   table_syy.csv   table_vlo.csv
table_amzn.csv  table_brkb.csv  table_cof.csv   table_do.csv    table_fdx.csv   table_hcbk.csv  table_jec.csv   table_m.csv     table_ne.csv    table_pep.csv   table_rop.csv   table_t.csv     table_vmc.csv
table_an.csv    table_bsx.csv   table_cog.csv   table_dov.csv   table_fe.csv    table_hcn.csv   table_jnj.csv   table_ma.csv    table_nee.csv   table_petm.csv  table_rost.csv  table_tap.csv   table_vno.csv
table_anf.csv   table_btu.csv   table_coh.csv   table_dow.csv   table_ffiv.csv  table_hcp.csv   table_jnpr.csv  table_mar.csv   table_nem.csv   table_pfe.csv   table_rrc.csv   table_tdc.csv   table_vrsn.csv
```

Figure 1. HDFS directory listing showing the flat directory structure for the symbol price data from the QuantQuote dataset.

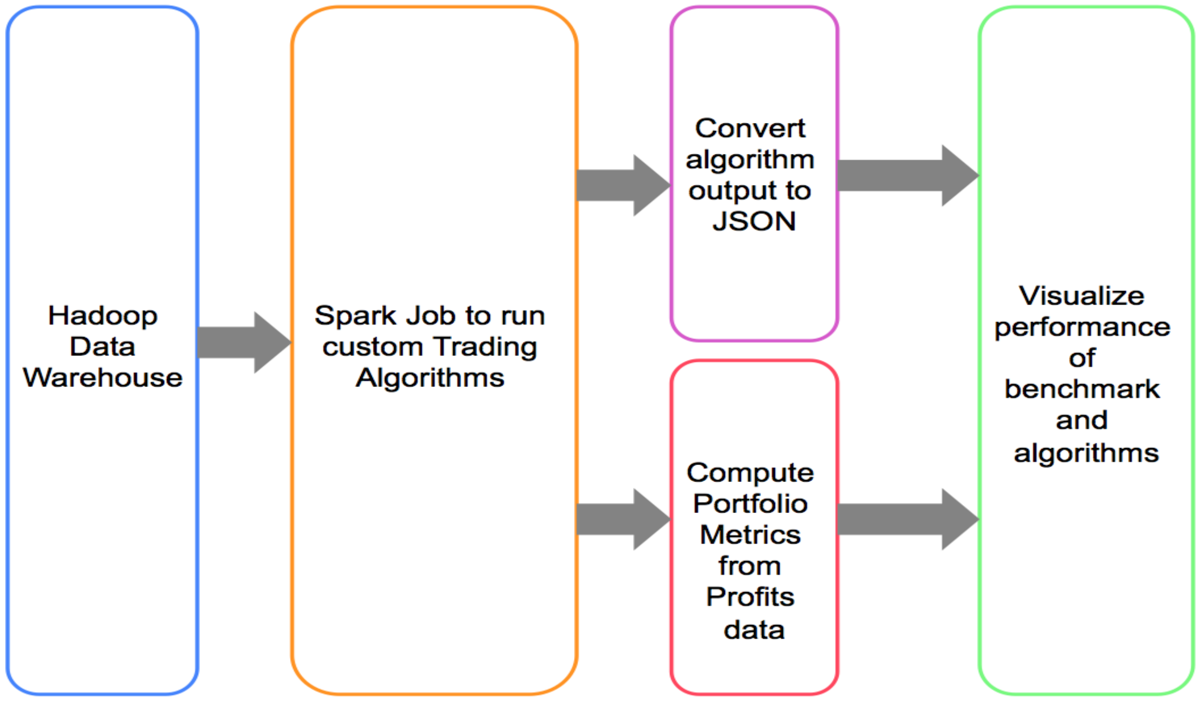

Hadoop Data Warehouse
Spark Job to run custom Trading Algorithms
Convert algorithm output to JSON
Compute Portfolio Metrics from Profits data
Visualize performance of benchmark and algorithms

Figure 2. Schematic depicting the overall system organization

Appendix

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.