**The Problem**

Getting into cryptocurrency daytrading starting out pretty rough? Wondering how to make a profit by buying and selling crypto? The best way to do this, with the appropriate hardware, is to do arbitrage. Arbitrage is when you buy an asset at a certain price in one marketplace and sell it for a profit in another marketplace. The goal of our app is to display opportunities for arbitrage by compiling different prices from different cryptocurrency marketplaces.

**The Data Solution**

This project will clearly involve data management to scrape, clean, store, and retrieve the data from the cryptocurrency marketplaces. It is important since it may make cryptocurrency marketplaces even more accessible and thus provide easier usage to people who are trying to learn about crypto.

Our project is a semi-standard term project of the course of [Database Systems/CompSci 516](https://courses.cs.duke.edu/spring22/compsci516/) at Duke University. A semi-standard project is basically developed by team members from project idea to end product.

**Design**

The design is relatively simple and can be accessed at the wireframe [here](https://lucid.app/lucidchart/08705157-421c-45b3-8788-79ed5dcb74e7/edit?invitationId=inv_13491946-1028-4370-aa29-3c144bbea015). It will have two main pages: one for the current arbitrage data and one for historical data. The main user interaction will be inputting request data into fields: the currency, the exchanges, and (for historical data) the date range.

**The Basic Features**

We plan to create a user portal in which you can search/add the currencies you wish to look at and compare, then display the various prices and some statistics from each marketplace for each cryptocurrency selected.

Additionally, we will have scheduled scraping so that the data can be updated periodically, showing users multiple potential arbitrage opportunities throughout the day. Thus, the four basic features will be:

1. Database (Perfectly Realized)
   * The scraped data is stored in a PostgreSQL database using PGAdmin4.
2. Data Gathering (Perfectly Realized)
   * We use Python’s requests and psycopg2 libraries to access the APIs of different exchanges and to connect the backend script and populate the new prices every hour (running on one of our home computers). The automation was done with Python’s apscheduler library, in particular using a BlockingScheduler.
3. Current Prices (Perfectly Realized)
4. Historical Data: In addition to recent prices, the user will be able to view tables and charts of historical arbitrage opportunities.

These two last points rely on two additional technical components. We use Python, in particular Flask, for database interactions dependent on user inputs. Additionally, we have developed a front-end with a good user experience to view all the crypto opportunities they desire.

**The Stretch Goals/Bonus Features**

To improve our project further, we want to make:

1. User accounts (not attempted at all)
2. Saving persistent watchlists (not attempted at all)
3. Benchmarking historical displays with other financial indices (not attempted at all)
4. Scrape and display news articles pertaining to crypto/finance (not attempted at all)

**Survey of previous work**

There are numerous studies analyzing arbitrage opportunities in cryptocurrency markets in the literature. For example, Markov and Schoar (2020) showed that the price of cryptocurrencies depends on local markets, and this variety created arbitrage opportunities on an hourly or daily basis. Coinmarketcap.com one of the world's most trusted & accurate sources for the crypto market described the reasons for price differences as: market inefficiency, low-volume exchanges, and different volumes of supply and demands of assets

As a result of the prices differences between crypto assets and marketplaces, traders seek different approaches to make a profit by buying at lower prices and selling at higher ones. However, the time between transactions and the transaction speed are important problems with the process, as one could potentially see an arbitrage opportunity which dissappears by the time their transactions could run. Therefore, traders and market makers use bots and trading algorithms to manage high-frequency transactions in milliseconds. With that, there are some tradeoffs, risks, and limits to creating an infrastructure to come up with high-frequency transactions.

One of the important limitations is the transaction fees of the markets. In other words, it may not be worth making a transaction between two markets because transaction fees may be higher than your expected profit (Boonpeam et al, 2021).

Secondly, lack of speed of database systems can be an issue for high-speed transactions. To compare thousands of the prices in dozens of markets in milliseconds makes this process very challenging. Moreover, there are some limitations to sending too many queries to the individual markets. Additionally, one may need to use the APIs of each market maker, which can cause issues with the integrity of one’s system if the inputs are not correctly formatted and validated.

To sum up the current literature: price differences between crypto assets in different markets offer great opportunities for profit, although there are some limitations and challenges including creating high-frequency infrastructure, sourcing data, and ensuring arbittrage opportunities can actually lead to profits. Our project will provide a database system to show the arbitrage opportunities and track those on a historical basis rather than making a transaction.

Previous work such as Pionex, Coinrule, Bitsgap, Cryptohopper are platforms that allow users to look at market arbitrage opportunities and consist of bots that can perform trading. These sites update their arbitrage database based on data from third party sources that scan data across multiple sources such as CoinMarketCap. This data however, may not always be accurate. The most accurate way to compare arbitrage opportunities is to look at data directly from the markets. A practical implementation of this method is hard due to its lack of scalability. Our application aims to provide arbitrage opportunities across various markets for free and use both market data and third-party sources to provide this information so that users can make a more informed decision.

**Evaluation of Our System**

We believe our system runs very smoothly and the user experience is more than just adequate. Additionally, we have attempted something that has no open-source code out there (other companies are profiting off of this idea by offering bots/automated trading) and so we have created all our algorithms and ideas originally, using only the libraries in Python and the PostgreSQL database. When the computer is online (i.e., when our home wifi does not crash), the scheduler runs quickly: within seconds, it queries all the different exchanges for the various currencies we display, and stores them in a local PGAdmin4 database.

**Future Directions**

We are very proud of the progress we have made on our project. In particular, we hit all our main goals and achieved the user experience we desired. In the future, we would ideally implement more of the bonus features that would help the user experience (saved watchlists in particular, for avid traders of only certain coins). Also, we would like to add more coins and exchanges in the future, as more exchanges release their API’s.

Additionally, we would like to shorten the interval between data-collections (each hour was our goal which we met, but it would ideally be every five seconds). This would require the data-collection script to run on a faster backend server, and on a computer dedicated to that purpose (the home computer that is being used currently performs other tasks as well). The main issue currently is that with a local computer, if the processing speed is too low or if the internet cuts out, then the API calls will fail, and so the scheduler will just skip that hour. This is a hole in the database that would need to be corrected by hand. With a dedicated server, this would not be an issue as its whole purpose would be to run this one script. In this sense, this is a resource issue rather than a programming issue: if we had the resources to store and quickly query a large database and to have an always-online computer/backend server, this issue would be solved.

The pursuit of a free crypto-arbitrage website (where we do not force users to pay for bots) is badly needed in the speculators’ community, especially as virtual commodities and currencies become much more mainstream (the rise of NFTs, etc.). For this reason, we hope with more resources, this idea gets fleshed out and shared with the crypto community.

**References**

[1] Varian, Hal R. "The arbitrage principle in financial economics." Journal of Economic Perspectives 1.2 (1987): 55-72.

[2] Makarov, Igor, and Antoinette Schoar. "Trading and arbitrage in cryptocurrency markets." Journal of Financial Economics 135.2 (2020): 293-319.

[3] How to Benefit From Crypto Arbitrage. https://coinmarketcap.com/ Accessed 7 Feb 2022.

[4] Boonpeam, Naratorn, Warodom Werapun, and Tanakorn Karode. "The arbitrage system on decentralized exchanges." 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, 2021.

[5] Corradi, Anna. “Has Top Cryptocurrency Website CoinMarketCap Completely given up on Data Accuracy?” Forbes, Forbes Magazine, 23 July 2019, https://www.forbes.com/sites/annacorradi/2019/07/23/has-top-cryptocurrency-website-coinmarketcap-completely-given-up-on-data-accuracy/?sh=37a42a9772d9.

[6] “Crypto Arbitrage Guide - What It Is and How to Find It - algotrading101 Blog.” Quantitative Trading Ideas and Guides - AlgoTrading101 Blog, 14 May 2021, https://algotrading101.com/learn/crypto-arbitrage-guide/.