NIBLS COIN

USA TECH + BALL

Dashawn Bledsoe

WHITEPAPER REPORT

Prepared by

Dashawn Bledsoe

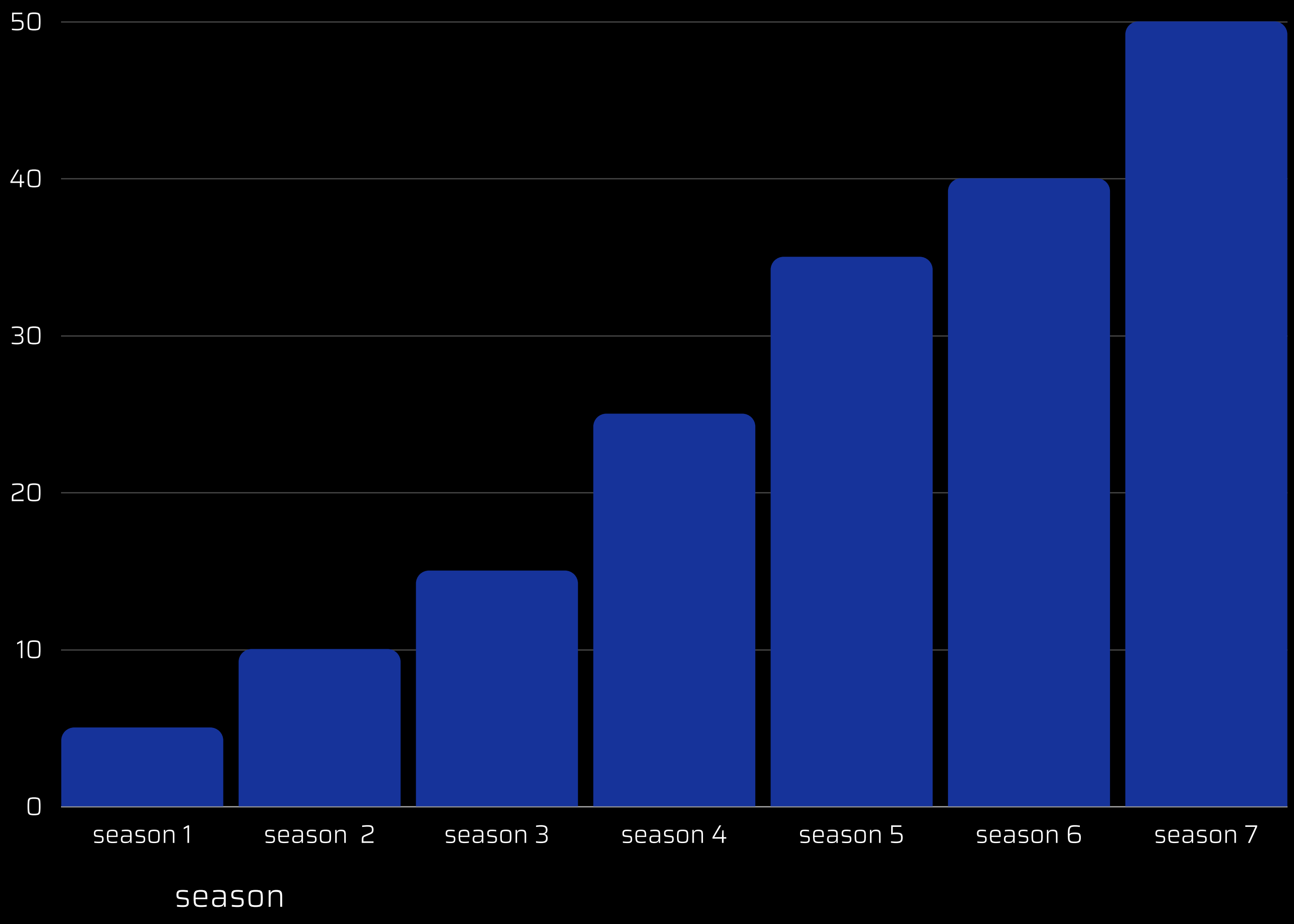
JULY 2022

Table of Contents

$22,212,820,000.00

Company valuation (as of 2022-27)

Tech+Ball Live Spectator Events



Summary of How Decentralization works as a Capital Market Application

Horizontal Partitioning and distributed query processing

Sharding (aka horizontal partitioning) and distributed query processing. Distributed databases are often implemented through "Sharding". The concept of database sharding has been gaining popularity over the past several years due to the enormous growth in transaction volume and size of business application databases for services such as:

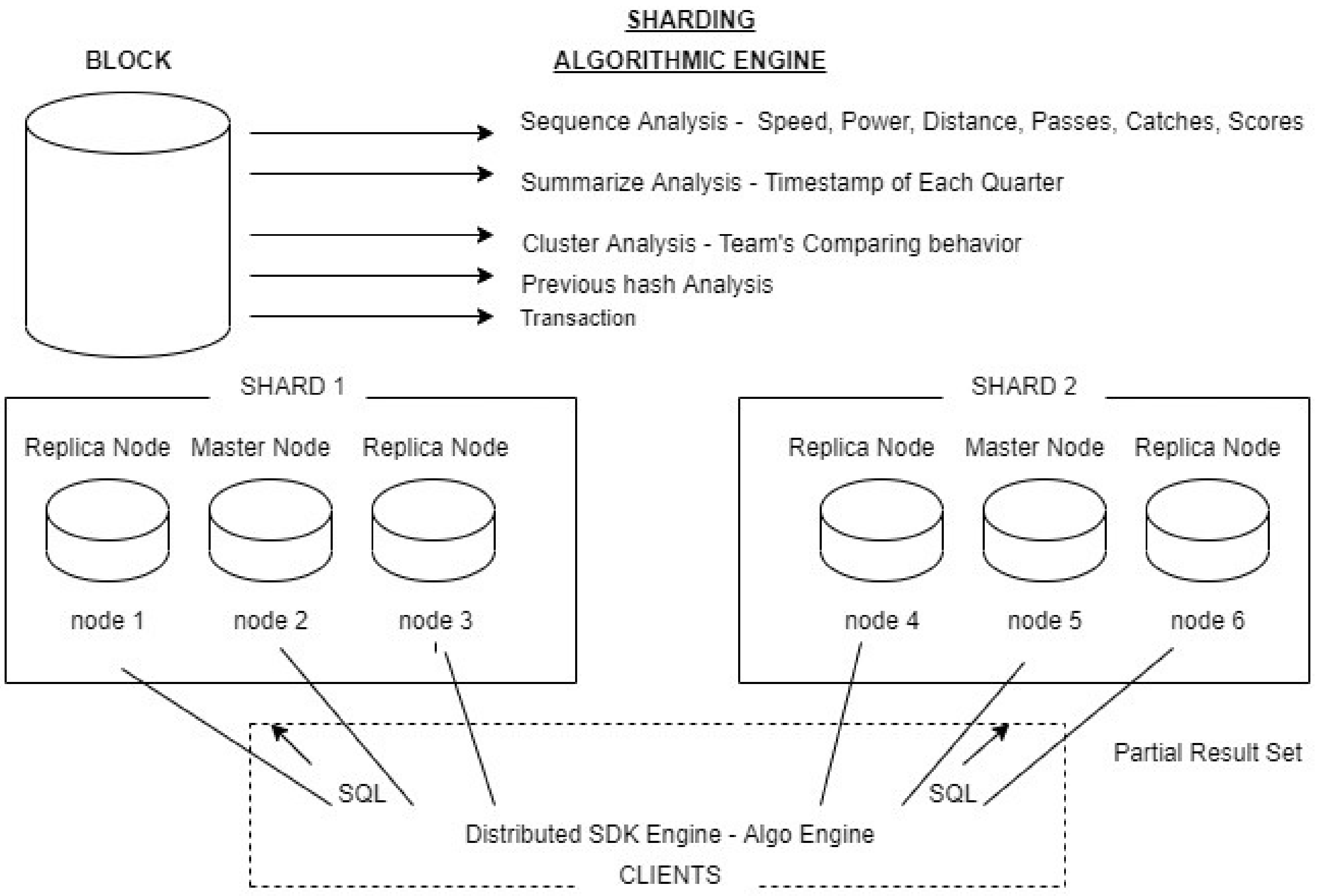
Online service providers Software as a Service (SaaS) companies
Social networking Web sites
Capital Market applications

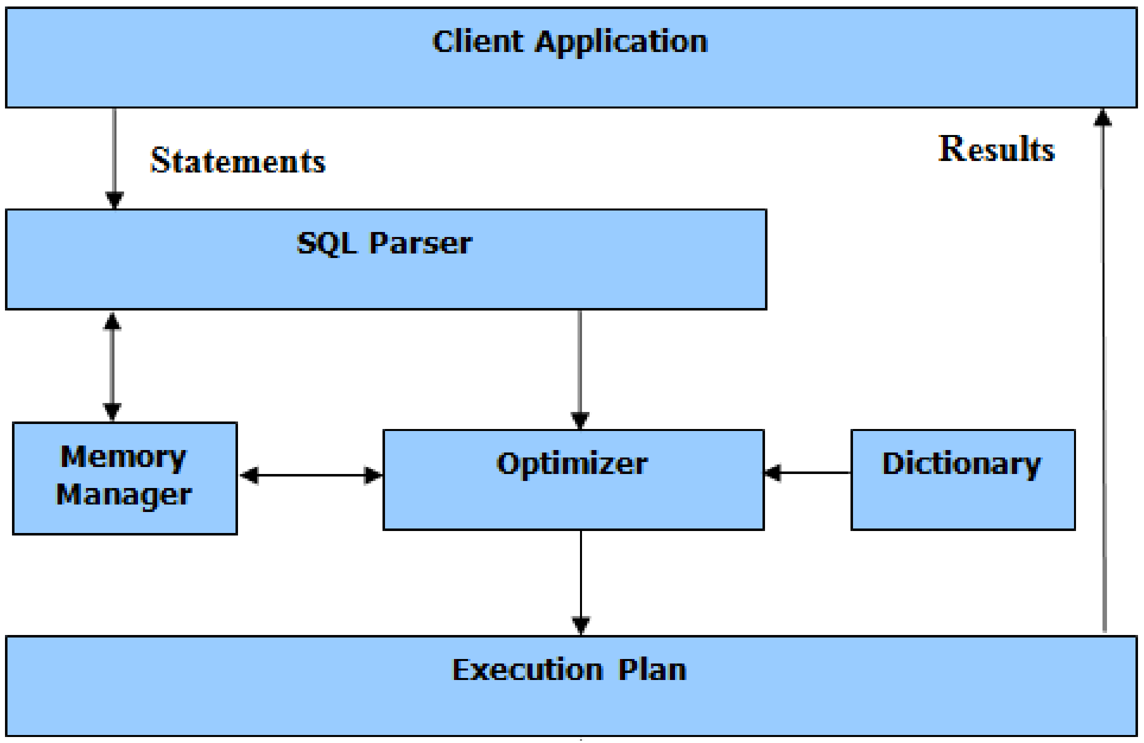
Sharding can be simply defined as "shared-nothing" horizontal partitioning of a database into a number of smaller database instances that collectively comprise one logical database. What drives the need for sharding is the fact that as the size and transaction volume of the database incurs linear growth, response times tend to grow logarithmically. Distributed queries allow far faster processing due to performing parallel execution of on each shard. It is well known that database performance drops in concert with an increase in database size. This is due in large part to the increasing size (depth) of index structures such as Merkle trees. Furthermore, instead of a single database server providing query execution for a single physical database, we can allow for server database node processes to provide the query execution. This is called distributed query processing and is transparent to a database client application. The client application merely opens the logical database, and the database configuration (described in a system architecture document) determines the physical makeup of the database (i.e. whether logical and physical mean the same thing, or whether the logical database consists of 2, 5 or 100 physical partitions, etc). Once connected, the client application begins/commits/aborts its transactions and queries in an algorithmic way. If the database is physically partitioned, the SDK query engine takes care of distributing the query on behalf of the client application and gathering (appending) the result data sets from each partition to present a single (logical) view to the client application. Sharding in the consensus layer can be combined. In this configuration, a single logical database is horizontally partitioned into two or more physical database instances. Each shard (physical database instance) then has a master server process, and a verification can be promoted to blockchain encrypted as a replication in generational memory the event of a real-time recorded descriptive data analysis of the original master. In this way, we can preserve the availability of the single logical database that would otherwise not be possible if any of the servers processing the shards were to fail.

Trainings, Live sports games, Concerts, festivals, & Conference Mining Performance

Algo trading and Mining

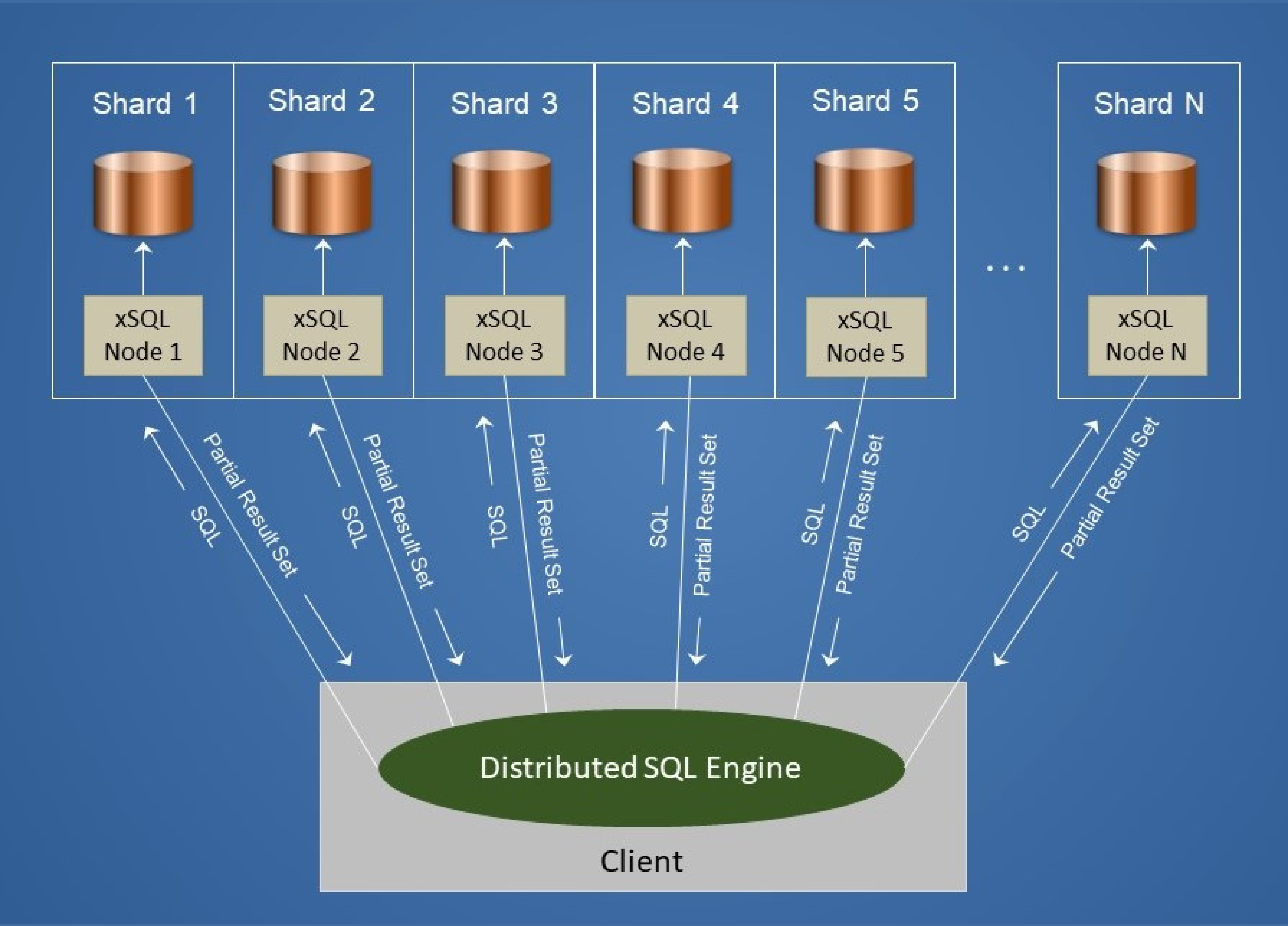
The current intent at Ethereum is to use a mining algorithm where miners are required to fetch random data from the state, compute some randomly selected transactions from the last N blocks in the blockchain and return the hash of the result. NIBLS Coin is fetching the same required randomly selected data at event performance's like concerts, festivals, & conferences. With trainings and live sports games (reference the diagram of sharding) instead of fetching data from the N Block we are using our algorithm to fetch the master, using query optimization we make instructions and rules that place automated trade decisions or we can have clients order trades to buy or sell real-time data from Official game ball data collection in the database network.


SHARDING
ALGORITHMIC ENGINE
BLOCK
Sequence Analysis - Speed, Power, Distance, Passes, Catches, Scores
Summarize Analysis - Timestamp of Each Quarter
Cluster Analysis - Team's Comparing behavior
Previous hash Analysis
Transaction
SHARD 1
Replica Node
Master Node
Replica Node
node 1
node 2
node 3
SHARD 2
Replica Node
Master Node
Replica Node
node 4
node 5
node 6
Partial Result Set
SQL
SQL
Distributed SDK Engine - Algo Engine
CLIENTS


Client Application
Statements
Results
SQL Parser
Memory Manager
Optimizer
Dictionary
Execution Plan

SHARDING - EXAMPLE TWO

CONCERTS , FESTIVALS, CONFERENCES



NIBLS Coin Execution

for Market Data , Order Router, Algorithmic Engine.

This diagram depicts the major operational steps and elements of SDK consensus layers statements embedded within application code and submitted to cloud computing service host through Azure. The SDK statements are programed to verify their syntactical correctness. Then, if no errors are found, then is invoked. The algorithmic engine attempts to determine the most efficient means of processing the statement by interrogating the database dictionary to discover potential indexes and autoid/reference relationships between classes. This step results in an execution plan that identifies the procedural steps of sequence discovery analysis, clustering analysis, & summarization analysis that will take in producing the result set.

(Example., locate an object of class X by the index or ALT team token on field A; use the value of field B of the found object as a search value on indexed field D of class Y, and so on. After the statement has been parsed, then an execution plan formed, the database is updated or queried.

statements are embedded in C, C++, python or C# within a single-threaded application by instantiating an engine object and calling its execute method.

However, multiple processes can simultaneously access a database in shared memory, or two or more threads within a single process can simultaneous access an in-memory using the engine.

Specifically:

- dynamic memory allocation that was previously occurring during statement compilation and building the execution plan has been replaced with static memory “block” allocation
- “materialization” takes place only when necessary for sort and aggregation operations
- the SQL C++ and C SDK have been slightly modified to reduce the number of virtual and indirect calls, and use arrays instead of iterators where possible
- database metadata is shared between sessions to avoid caching
- a reference to the current memory allocator is used to eliminate the use of thread-specific memory.

Algo Engine Performance Continued

Query optimization

analyze SQL queries sent to the database and select the best search strategies for accessing the database.

Creating the optimal plan for execution of SQL statements:

There are two classes of SQL optimizers: cost-based instructions and rule-based.

With cost-based optimizers, query optimization greatly depends on data distribution. and collect statistical information themselves, to calculate the cost of candidate execution plans.

a rule-based optimizer that enables the developer to specify query execution plans within an application. For example, the optimizer never reorders tables in the query: the joins are performed in the sequence the tables were specified. Some of the other key rules that are used for query optimization include:

- If possible, an index is used.
- Each table is assigned an ordinal number representing its position in the From list.
- The search predicate is divided into the set of conjuncts and the conjuncts are sorted. Therefore, the expressions accessing the tables with smaller ordinal numbers are checked first.
- The execution of subqueries is optimized by checking the dependencies of the subquery expression. The results of the subquery are saved and recalculated only if the subquery expression refers to fields from the enclosing scope.

Continued...

Using A rule-based query scanning to supports goals of

1) Predictable and fast execution of queries and tune the queries and specify execution plan

2) By arrange tables and filters according to simple rules using indexes avoid sequential table scans whenever possible.

3) Base rules mostly using Indexes and avoiding sequential table scans whenever possible.

4) Combination of ascending or descending indexes order by clauses, comparison operates (<=>, ...) in search constrains, etc.

5) Some operations are handled without Indexes: example between expressions is not processed through index but instead the engines do a lookup based on one if the boundaries of the range, and simply checks the validity of the other boundry.

6) Smart algorithm is used to choose the right index for a query. There are various criteria: the engine attempts to pick the index that supports the required order and replaces the maximum Number of predicates in the query while keeping the index length a minimum.

Scalability

Algorithm to identify Arbitrage & Rules Implemented

We start by building an algorithm to identify arbitrage opportunities. (example)

Requirements:

Computer program that can read current market price metrics with rule-based query scanning to support goals
(Using Descriptive Data analysis)

Value 1: Distance   (measured or tracked long distance travels will be half a token
Value 2: Passes (Successful connected passes to teammates will be more than half of a token)
Value 3: Catches   (Successful connections to players will represent half tokens)
Value 4: Scores   (Successful 6-point scores recorded will represent full tokens)
Value 5: Power   (Tracking of powerful ball travels will result in half tokens)
Value 6: Speed   (recorded speeds of how fast the ball travels will be half a token)
Value 7:   Steals   (Successful steals representdecrease on blockchain for offensive alt team token and the defending team recorded steal gives that ALT team a increase of tokens in realtime)

continued...

Can we explore the possibility of arbitrage trading on Index symbols (ALT TEAM TOKENS) listed for international marketplace?

Read the incoming price feed of each analysis from both indexes and exchanges.

Price feeds from both matches.

Order-placing capability that can route the order to the correct exchange.

walkforward testing capability on real-time price feeds.

The computer program should perform the following:

Read the incoming price feed of each analysis from all Indexes and exchanges.

using the available foreign exchange rates, convert the price of one currency to the other.

If there is a large enough price discrepancy (discount the brokerage costs) leading to a profitable opportunity.

Technical Requirements for NIBLS Coin Algo Trading

Token Exchange Service

Implementing the algorithm using a program language is the final component accompanied by walk forward testing (Trying out the algorithm with biometric balls to see if using it is profitable).

The challenge is to transform the identified strategy into an integrated computerized process that has access to a trading account for placing orders.

The following are the requirements for algorithmic trading:

Computer - programming knowledge to program the required trading strategy, hired programmers.

Manu-tech of technology ball to record real-time data.

Sports athletes & teams to schedule daily active sports challenges for testing algorithm to place orders.

The ability and infrastructure to walk forward test the system before it goes live on real markets

walk forward testing depending on the complexity of the rules implemented in algorithm.

Timeline & Referenced further Reads

Manufacture & Code	Test week 1	Test week 2	Build up data	M&A Exit
Manufacture Hardware & Develop computer program	Test computer program	walkforward test proof of stake	Play local seasons & Playoffs and collect data	Merge or become a acquired for an exit

Notes:

Name of Athletes or Teams reference

https://ethereum.org/en/whitepaper/# identity-and-reputation-systems

Proof of stake

https://ethereum.org/en/whitepaper/# mining

Financial derivatives and stable value currencies

https://ethereum.org/en/whitepaper/# financial-derivatives-and-stable-value-currencies

Fees

https://ethereum.org/en/whitepaper/# fees

NIBLS HARDWARE
USA TECH + BALL

Questions? Contact us.

www.Linkedin.com/in/drb-family-business-prvte

bledsoedashawn130@gmail.com

+1 347 754 6933