Atlas: the Spatial Consensus Blockchain

Project Hyperion
Decentralized Map Economy for 10B People
Prepared by: Team Hyperion (https://www.hyn.space)
25 August 2018

Atlas

Noun

1. Latin, from Greek Mythology *Atlas*
: a Titan, son of Iapetus and Oceanid Asia (or Clymene), condemned to hold up the sky for eternity

2. a bound collection of maps often including illustrations and informative tables, by Merriam-Webster dictionary

3. The Spatial Consensus Blockchain to support world-scale map services

ATLAS Objectives

Hyperion introduces a new global decentralized map economy for 10 billion people. Atlas is the enabling component of Hyperion to support world-scale map services on blockchain. We master peer-reviewed innovations proven in practice. Atlas focuses on the generality of a scalable blockchain for space and provides extensible smart contract support for specific features such as localization and location search. In addition to the inherent requirement of decentralization i.e., *censorship resistance, open participation and fault-tolerance*, Atlas also aims to meet the following design objectives:

- Scalable: Massive surface areas of earth with over 6 * 2^60 cells at the finest grain level. Over 800 billion localization request a day for 10 billion users estimated from the Tencent report. Nearly one third of searches of Google are location based queries. Billions of spatially distributed Internet of Things (IoT) devices sense and map the world at unprecedented scale.

- *Low-latency*: High-latency map services impair user experience and lead to real potential consequences. For example, additional traveling time, direct expenses and missed opportunities, commonly incurable during a high-speed setting such as driving. Latency also affects data freshness, which is crucial to applications like Self-Optimizing Network and Autonomous cars.

- *Low-cost*: Map services are mostly free for consumers with the freemium model but expensive for corporate use. The recent pricing scheme of Google Map is a multifold cost increase to many locked-in corporations. Lower cost of map services provided by Mapbox helped attract many big name adoptions e.g., Snapchat.

- *Privacy-preserved*: The Freemium model comes at the cost of user privacy trace in the form of location search priority ranking and location-based advertisement. Recently TechCrunch reports Google keeping user location history. In addition to user trace privacy, sensitive map data privacy should also be accounted for.

- *Extensible*: Natively provide high-performance location-based transactions (e.g. localization, nearby search) to location-based services via Smart Contracts. Support indexing of arbitrary location data, including static map data, time-series sensory data and location traces.

- *Interoperable*: Atlas, as a location oracle, requires cross-chain communication by bi-directional lightweight transaction verification and integration with relay-chain platform e.g., Polkadot.

Elastic Spatial Sharding

The Challenges

Elastic

Unprecedented and unpredictable skews in both space and time of global data, e.g., uncontrollable moving sensors in large-scale. It is necessary to handle the data skewness, load-balancing and index dynamic location data at massive scale. Besides, the Atlas network will gain increasing community support over time to support increasing network throughput.

Spatial

Location is the metadata of a wide range of data types, e.g., IoT data, which makes them inherently geospatial. Atlas needs to index arbitrary forms of location data, with an outward link to a fault-tolerant distributed database and storage.

Basic geospatial queries (e.g., polygon containment, intersection and spatial-join operation) require efficient and robust multidimensional indexing techniques. However, R-tree family index incurs large reconstruction cost with high frequency object insertion (normal in real world), making balancing tree structure difficult to achieve. Quad-tree family is difficult to handle efficient non-point data index. An efficient partition-friendly spatial index data structure is the key to ingest, index and query large-scale location in data for efficient performance.

Sharding

Sheer scale and fast speed of spatial data are challenges as mentioned. Existing spatial indexing techniques e.g., GiST, are not designed for massive parallelization and distributed environment with high data churn. It is both unrealistic and unnecessary to have all machines in the Atlas Blockchain network validating every transactions - sharding allows the network overall throughput to scale linearly/sub-linearly with increased number of validators.

The Solution

Atlas aims to design a sharding scheme for massive parallel operation (ingesting, indexing and querying spatial data) with desirable properties such as skewness-resistance, *lazy* node creation for fast data ingestion and locality-preserving. To that effect, we exploit opportunities of inherent location data characteristics, that are, locality-preserving and easy to parallelize. We leverage different characteristics of the Hilbert Curve, a fractal space-filling curve, for many design considerations as following:

- Space-filling: to address every location on earth with an unique identifier without singularity and discontinuity;
- Decompositional: to natively support a high-dimensionality tree data structure that uniformly partitions data workloads in lower-dimensionality for skewness resistance and elasticity of the distributed data structure;
- Deterministic: to enable *lazy* object assignments and avoid costly location node merge/split operations;
- Locality-preserving: to align the network-level and application locality, and to generate local processing unit in our spatial sharding scheme, respectively.

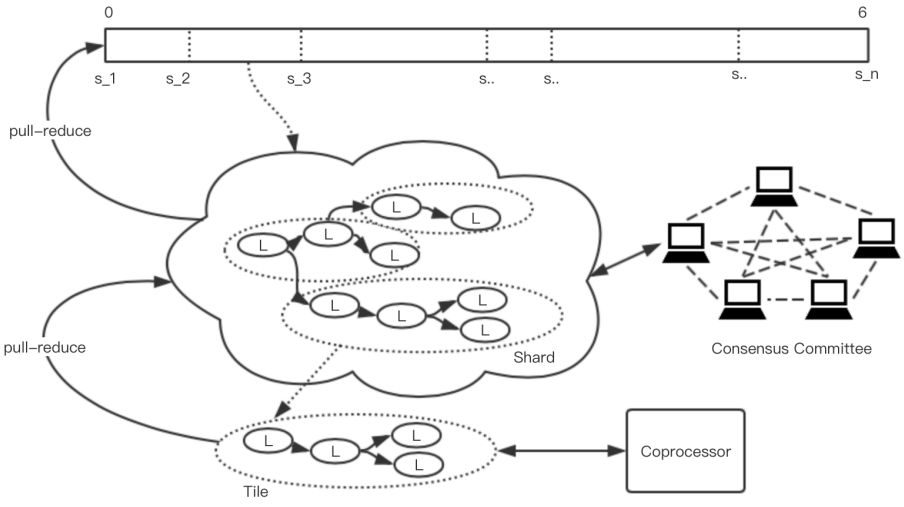

Figure 1: an overview of Elastic Space Sharding. s_i denotes the starting key of *shard i* in range partitioning. Each shard has its consensus committee to process location-based transactions. Within each shard, the multi-dimensional tree is further divided into tiles with locality preservation for coprocessing. The letter L in the tree nodes abbreviates *location*.

Atlas performs a two-stage elastic sharding to distribute the location-based transactions workload over shards and over cores of validators. Local node states are merged, firstly across coprocessors then across shards, via two-stage pull-reduce model, extended from the Gather-Apply-Scatter model in parallel processing.

Multi-level Location Addressing: Hilbert Curve allows Atlas to use 1-dimension key to *fully* and deterministically address high-dimension spaces (e.g., with block time as an additional dimension). Each location is a logical storage buffer to index arbitrary types of data with a key-value structure widely supported in distributed databases. Atlas employs a multi-dimensionality tree structure. Nodes on each level are addressed with different *decomposition* order of Hilbert Curve similar to the *S2 cell addressing scheme*. Each location, a cell, is indexed with a 64-bit cell_ID in range of 0 to 6.

Elastic Range-sharding: forming shards with consensus committee to validate transactions. High dimensionality tree structure allows us to more evenly partition *skewed* workloads on nodes with lower native dimensions. Since node address assigned by Hilbert Curve is locality-preserved by nature, sharding spaces is essentially r*ange-partitioning* of cell_ID in [0, 6). This scheme supports re-sharding elastically by simply adjusting the range partitioning intervals and moving only the objects in the changed regions. This deterministic addressing scheme enables *lazy* node creation with least node split/merge operation for efficient distributed spatial indexing, especially for high ingestion rate situations.

Atlas distributes the incoming data by shards at indexing time to reduce rebalancing overhead. Further, the node addressing strategy ensures a uniform partitioning of spatial objects effectively avoiding *hotspots* in a distributed setting*. Atlas integrates design of Locality-Preserving Distributed Systems*[1] to optimize locality preservation on both network and applications level, allowing users to interact with “nearby" shards to reduce latency.

Local Tile-sharding: forming tiles as local processing units for massive parallelization. Micro-sharding is inspired from *Mosaic: processing trillion edge graph on a single machine*[2]. Mosaic devises an effective scheme based on Hilbert Curve to generate locality-preserving *Hilbert-order Tile.* Each tile is a local processing unit of batch edges (ie., local tree) for multi-core processing (up to 244 cores in Mosaic) with high locality and compression rate. This design has a simple load-balancing scheme solving the inherent problem of handling large dataset in single node (low-locality and load-balancing issues due to skewed data structures within the shard).

---

[1] Basescu, C., Nowlan, M.F., Nikitin, K., Faleiro, J.M., & Ford, B. (2014). Crux: Locality-Preserving Distributed Services.

[2] Maass, S., Min, C., Kashyap, S., Kang, W., Kumar, M., & Kim, T. (2017). Mosaic: Processing a Trillion-Edge Graph on a Single Machine. *EuroSys*.

Block Production Protocol

Key inspiration are from peer-reviewed works of OmniLedger[3] and RapidChain[4]. Atlas combines the innovation of both works with other proven works.

Security and Consistency

Random Membership

Public randomness based on public cryptographic sortition protocols for consensus group *bootstrapping*[5] and *reconfiguration* (joining and re-organizing) over epochs to resist long-term security compromise. Provable robust reconfiguration[6] to prevent join-leave attack with Cuckoo Rule with Distributed Hash Table.

Atomic Cross-shard

Each transaction is either committed or eventually aborted with simple lock-unlock model with O(1)-size coordination.

Robust Transition

Gradual transition to ensure BTF consensus with at least 2/3 honest majorities and minimize the chance of temporary loss of liveness. Fast view-change to defend leader failure and single-shard DoS attack.

---

[3] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018). OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding. *2018 IEEE Symposium on Security and Privacy (SP)*, 583-598.

[4] Zamani, S.M., Movahedi, M., & Raykova, M. (2018). RapidChain : A Fast Blockchain Protocol via Full.

[5] Syta, E., Jovanovic, P., Kokoris-Kogias, E., Gailly, N., Gasser, L., Khoffi, I., Fischer, M.J., & Ford, B. (2016). Scalable Bias-Resistant Distributed Randomness. *2017 IEEE Symposium on Security and Privacy (SP)*, 444-460.

[6] Awerbuch, B., & Scheideler, C. (2006). Towards a Scalable and Robust DHT. *Theory of Computing Systems, 45*, 234-260.

Scalable Consensus Coordination

Scalable Collective Signing

PBFT prepare and commit phases with collective signing (CoSi) rounds[7]. Reduce per round communication complexity with tree structure from O(n^2) to O(n) and signature verification from O(n) to O(1).

Checkpoint Pruning

Stable checkpoint technique in PBFT[8] and State blocks include multi-hop back pointers to headers of intermediate blocks[9].

Effective Information Dispersion[10]

Erasure coding mechanism to gossip large blogs over network and the protocol[11] for honest nodes agree on the Merkle tree root to ensure consistency.

Trust-but-verify

Fast validation of small-value transaction with optimistic confirms (within a few seconds) and elects another group of slow-but-secure validators to verify transactions (within a few minutes). Penalties are applied to dishonest validators.

---

[7] Kokoris-Kogias, E., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., & Ford, B. (2016). Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. *USENIX Security Symposium*.

[8] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In 3rd USENIX Symposium on Operating Systems Design and Implementation (OSDI), Feb. 1999.

[9] K. Nikitin, E. Kokoris-Kogias, P. Jovanovic, N. Gailly, L. Gasser, I. Khoffi, J. Cappos, and B. Ford. CHAINIAC: Proactive Software- Update Transparency via Collectively Signed Skipchains and Verified Builds. In 26th USENIX Security Symposium (USENIX Security 17), pages 1271–1287, Vancouver, BC, 2017. USENIX Association.

[10] Noga Alon, Haim Kaplan, Michael Krivelevich, Dahlia Malkhi, and JP Stern. Addendum to scalable secure storage when half the system is faulty. Information and Computation, 2004

[11] Ling Ren, Kartik Nayak, Ittai Abraham, and Srinivas Devadas. Practical synchronous byzantine consensus. CoRR, abs/1704.02397, 2017.

Privacy and Secret Sharing

Atlas will process various *map data*, *location traces* and *user profiles* that may contain sensitive information. It is desirable to have native support for privacy preserving of these data.

On-chain Secret

Users may want to have private map data on-chain with whitelisted access. We implement a system inspired by *CALYPSO: Auditable Sharing of Private Data over Blockchain*[12] and *TDH2 protocol*[13], a decentralized threshold cryptosystem with auditable access-control. The system is O(1) in write time and O(n) in read time with the number of Secret Control Trustees (SCTs).

The system is bootstrapped using Distributed Key Generation[14], in which *n* SCTs jointly generate a public/private key pair where the private key is retrievable via a t-out-of-n secret sharing scheme with verifiable validity by Non-interactive Zero Knowledge Proof. The same set of SCTS provide access control and audit record of read/write operations.

Zero Knowledge Location Proof

The user location trace is extremely sensitive. Atlas is designed to support zero knowledge proof of location traces satisfying predefined functions, e.g., geo-fencing check. In reference to the *Pepper Project: towards practical verifiable computation*[15] and zk-SNARK[16], we can attain O(n) setup time with arithmetic circuit size, O(n log n) prove time with security parameter and O(n) verification time with instance size (and security parameter).

---

[12] Kokoris-Kogias, E., Alp, E.C., Siby, S.D., Gailly, N., Syta E., Jovanovic, P., Gasser, L., & Ford, B. (2018). CALYPSO: Auditable Sharing of Private Data over Blockchain. *IACR Cryptology ePrint Archive, 2018*, 209.

[13] Shoup, V., & Gennaro, R. (1998). Securing Threshold Cryptosystems against Chosen Ciphertext Attack. *Journal of Cryptology, 15*, 75-96.

[14] Gennaro, R., Jarecki, S., Krawczyk, H., & Rabin, T. (1999). Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. *Journal of Cryptology, 20*, 51-83.

[15] Pepper: Toward practical verifiable computation. (n.d.). Retrieved from https://www.pepper-project.org/#publications

[16] Ben-Sasson, E., Chiesa, A., Tromer, E., & Virza, M. (2014). Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture. *USENIX Security Symposium*.

For each location proof contract, it must be bootstrapped with configurations (e.g. geo-fencing area). The compiler takes a program and transforms it into a quadratic arithmetic program (QAP)[17]. The QAP along with a security parameter will be used to generate the respective zk-SNARK proving and verification keys. User as the prover provides location and the respective proving key to generates a location proof. The proof is then recorded on Atlas and used by the verifier along with corresponding policy verification key to verify the designated proof.

Anonymous Profile

Atlas needs to keep user profile data anonymous on 3rd party access. Such profile may include information such as restaurant visits, dollar spending and reputation (combat paid reviews, inappropriate language etc.). We refer to *AnonRep: Towards Tracking-Resistant Anonymous Reputation*[18] in part of our implementation. The novelty is to store data without enabling long-term tracking by using mix-network (*mix-net*)[19] with verifiable shuffle and one-time pseudonym. Furthermore, defend against duplicated feedback with linkable ring signature. The system scales in O(n) with number of server participants.

For a new user, his/her profile data is sequentially encrypted by mix-net servers to generate a record <*user public key (private key sk), encrypted data*>, which is then broadcasted to all servers. In rounds, the system, with verifiable shuffle, transforms records on all servers into <*one-time pseudonym, plain-text data*>, where the *one-time pseudonym* is a one-time public key for a corresponding *sk*. Allowing users to anonymously interact with the system and third-parties to access profile data without revealing their actual owner. At the end of a round, mix-net obtains the updated records by reverse transformation.

All feedback (e.g. comment on a message) interactions with the system will be committed through linkable ring signatures[20]. Such that, all participants know the feedback came from a registered user without knowing the exact identity. Any dishonest client attempting to submit *duplicated* feedback on the same message will also be immediately detected.

---

[17] Gennaro, R., Gentry, C., Parno, B., & Raykova, M. (2012). Quadratic Span Programs and Succinct NIZKs without PCPs. *IACR Cryptology ePrint Archive*.

[18] Zhai, E., Wolinsky, D., Chen, R., Syta, E., Teng, C., & Ford, B. (2016). AnonRep: Towards Tracking-Resistant Anonymous Reputation. *NSDI*.

[19] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communica- tions of the ACM*, 24(2):84–88, 1981.

[20] Liu, J.K., Wei, V.K., & Wong, D.S. (2004). Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. *IACR Cryptology ePrint Archive, 2004*, 27.