Mikhail Surin

Curriculum Vitae

25/1-24 Oktyabrskaya str.
143980 Zheleznodorozhny
Russia
+7 (925) 705 32 46
surinmike@gmail.com
ssmike

Education

2013–2017 **Bachelor of Computer Science**, *Moscow Institute of Physics and Technology (State University)*, Russia, Dolgoprudny.
Thesis: Jepsen bindings for verification of serializable snapshots

2017–2020 **Master of Computer Science**, *Moscow Institute of Physics and Technology (State University)*, Russia, Dolgoprudny.
Thesis: Data structures for consensus algorithms on persistent memory

2016-2018 **Graduate**, *Yandex School of Data Analysis*, Russia, Moscow.

Experience

2022 – Now **Senior software engineer**, *Yandex*, Moscow.
YDB (is an open-source newsql dbms) query processor team.
Duties
- Development of the query compiler, optimizer and query execution runtime
- 2nd-line support of internal and external users

Achievements:
- Reworked the predicate pushdown logic to make use of calculated (non-literal) read ranges for table reads. As a result allowed users to efficiently use more complex filters in select statements including use of OR, tuple comparisons, type casts. To integrate in existing workloads designed and developed static trait inference for generated range expressions and implemented literal range preprocessor to optimize simple queries. Also performed various logic optimizations to better handle complex filters. Also made a point expression generation and static expression analyzer to use in join optimizer.
- Implemented automatic rule-based secondary inidices usage
- Reworked query execution model for OLTP pipeline moving the query logic execution from shards pipeline to a separate distributed service allowing users to stream table select results from shards removing the limit for the data read from shards in OLTP queries. Also the new execution model allowed the query processor to employ resource-based execution planning, improved TPC-C results by 40% and allowed the query processor to use the sequential shards scan in select statements with limit (before query was always executed on all affected shards).
- Added support for the ‘returning‘ keyword on the internal query language level forcing materialization of computed rows.
- Implemented cpu isolation for user queries which allowed users to configure database resource pools to prevent OLAP queries from affecting OLTP workloads and different OLAP workloads from affecting each other. Made use of lock-free data structures to avoid contention and designed event-based system to control user task weights. Experimented on different fair-share strategies.

Technologies: C++ 20, actor model, Python, Go

2019 – 2022 **Senior software engineer**, *Yandex*, Moscow.

yandex base search team.

Duties:

- Development and maintenance of lower levels of search runtime, a search index build pipeline and related infrastructure.
- Capacity planning of lower levels of search runtime.
- Support of ml-engineers from other search departments
- Led a team of 3 developers

Achievements:

- Reworked an inverted index build pipeline which enabled us to extract inverted index to a separate micro-service and save half of the compute resources.
- Designed and developed a low latency network storage for the search index with erasure coding support and strict latency requirements. Further optimized it using compile-time computations, optimized memory usage patterns making use of arenas, fixed-size ring buffers, zero-copy streams and for bigger throughput enabling the storage runtime to handle over 30k requests/disk reads per second on a single processor core. To improve throughput made use of asynchronous disk APIs, made experiments on software disk scheduling. Investigated and successfully mitigated SATA-related issues, linux scheduler latency issues, tuned the networking stack to optimize request latency.
- Implemented various optimizations in base ranking service using compile-time computations alongside with C++ concepts, fixed memory allocation patterns improving L2 cache metrics, reworked main ranking cycle to be event-based to efficiently pipeline storage requests.
- Put search document indices in the network storage which allowed us to to grow the search base by 100% and further reshard the search service enabling us to save 40% of compute resources
- Designed and developed a control plane for the storage with support for node/data evacuation and load balancing which enabled us to move the storage service to an internal cloud with automatic hosts maintenance and resource allocation. Further generalized it to use as a service in other departments. To improve the scalability of a planning controller designed and developed a reactive framework and to efficiently handle partial persistent state updates and a small ORM for YT objects in C++ to better employ devirtualization and copy elision. This framework improved 15 times reaction time on maintenance/host crashes and data delivery confirmations and allowed us to scale search base.
- Designed and led the generalization of the search storage for use in other departments as a service including a design of control plane. The list of successful integrations includes Yandex internal advertising platform.

Key technologies: C++ 17, MapReduce, Linux AIO, io_uring, Python, Go

2018 – 2019 **Software engineer**, *Yandex*, Moscow.

yandex base search team.

2017 – 2018 **Software engineer**, *Yandex*, Moscow.

news.yandex.com infrastructure team.

Duties

- Development and maintainance of a news scraping robot and document storage.
- Support of ml-engineers
- Second line support of partners (news agencies).

Achievements:

- Reworked the news document annotation service to by asynchronous which enabled the use of large batch neural models and optimized document annotation time 5 times. To compose computation-intencive and io-intensive (requests to external services) extensively used stackful coroutines alongside with concurrent programming techniques.
- Developed an external link feature which enabled the service to include links to social nets and comply with regional laws.
- Designed and developed a fault tolerant storage over multiple YT (in-house MapReduce system) clusters surviving drills and data-center outages. This enabled the document base to keep historical data, improved document delivery time by 30%. Also replaced distributed Redis caches, improving resilience.
- Designed and developed a cloud-native scheduler for checking news availability using the new document storage. Implemented reliable checker for historical data availability using OLAP and OLTP capabilities of an in-house MapReduce system (YT) and implemented rate-limiting system to avoid excessive load on partner sites.
- Designed a fault-tolerant stream clustering service instead of legacy non-incremental fresh news clusterization linked to a specific news indexer which enabled us to reduce amount of resources consumed by clustering by 5 times and created consistent clusterization of historical data. Also enabled the service to use persistent story IDs/urls which did not exist before the global clusterization.
- designed and developed a local falut-tolerant document storage for news indexer based on COW balanced trees optimized and write-ahead logging which enabled us to greatly simplify news ranking pipeline by using long-term data snapshots, build consistent data snapshots on historical data and created a possibility to reproduce exact inputs for ml-models.

Key technologies: C++ 14, Zookeeper, Redis, Lua, MapReduce, event-based architecture

2016 – 2017 **Junior software engineer**, *Yandex*, Moscow.

news.yandex.com infrastructure team.

2016 – 2016 **Software engineering intern**, *Yandex*, Moscow.

news.yandex.com infrastructure team.

Awards

2014 IMC – Second prize (107th place)

2014–2015 ACM ICPC Finals – 28th place

Languages

Russian Native speaker

English B2

Spanish B1