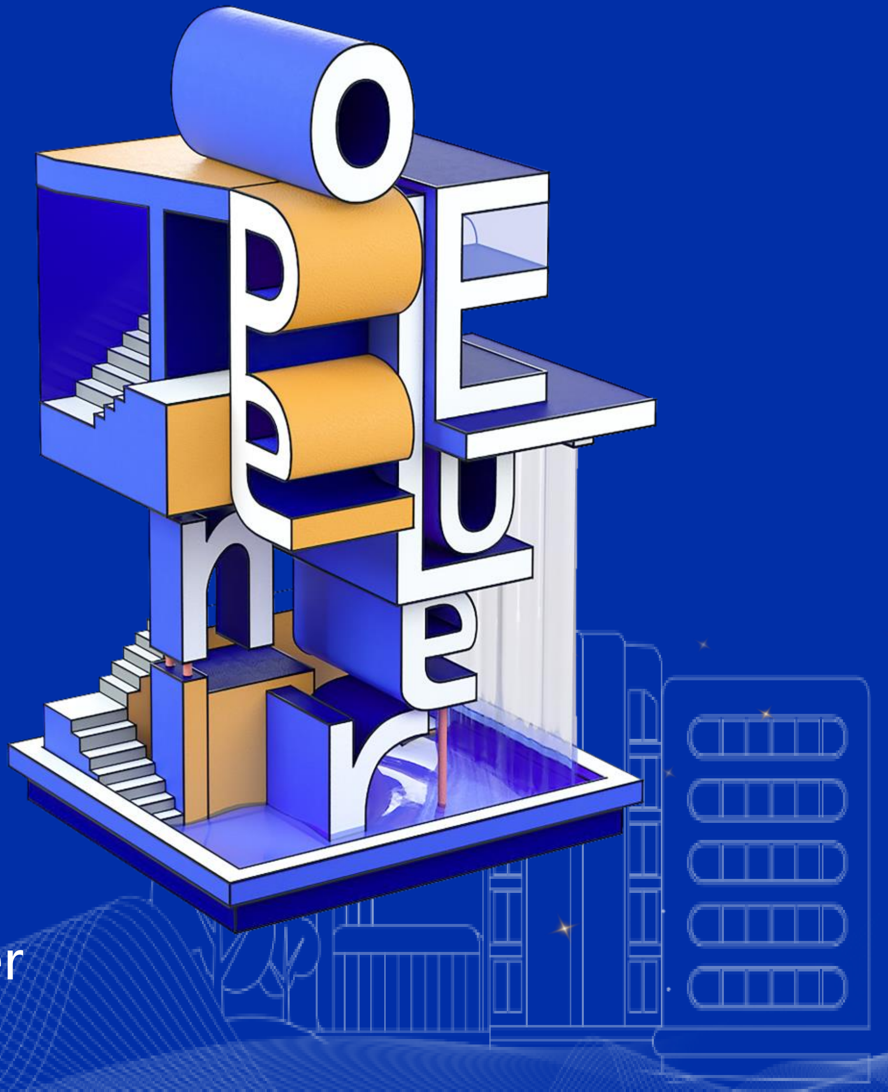
A-Tune: An AI-Based Automatic Parameter Tuning System

Donghui Chen

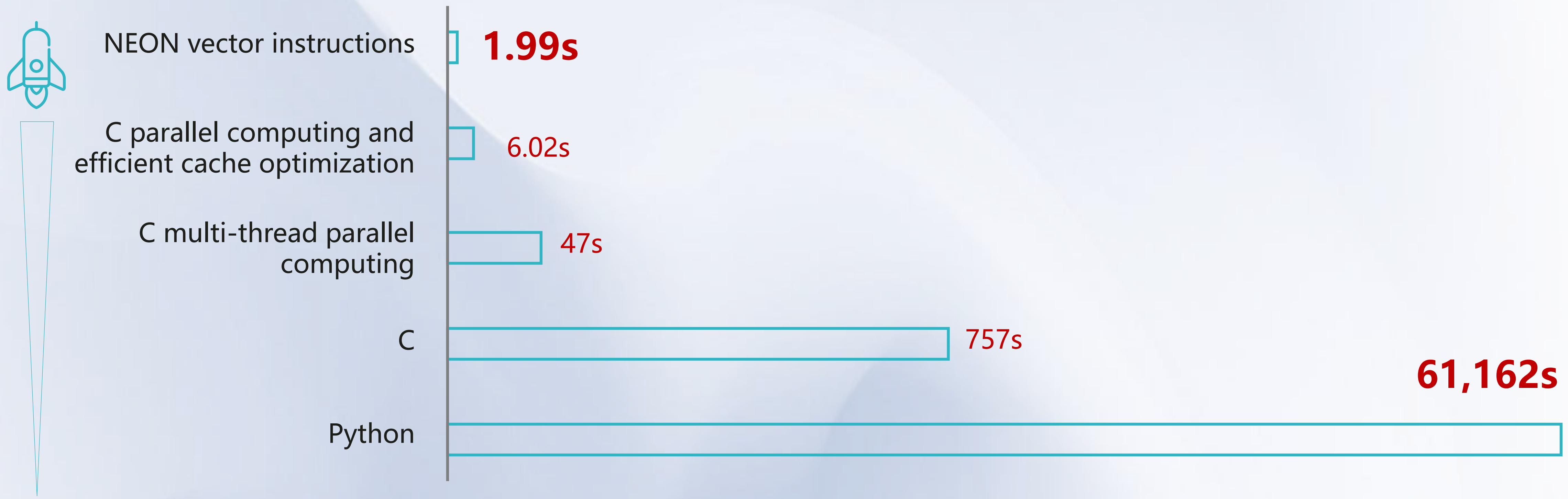
Senior software engineer, openEuler A-Tune SIG maintainer



Contents

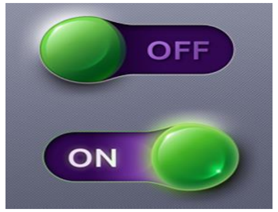
What Is Performance Tuning?

4800 x 4800 Matrix Multiplication Acceleration Results



Introduction

- An operating system (OS) has a vast number of parameters with complicated correlations between them.
- These tunable parameters control all aspects of the OS.

			OpenEuler	GCC		App	
CPU • LLC cache prefetch • Memory interleaving • …	**ACC** • Encryption • Decryption • …	**NIC** • Interrupt affinity • …	**OS** • Scheduling affinity • …	**Compiler** • FDO • Parallelism • …	**Framework** • Cache size • memcpy size • …	**Application** • Workload • Threads • User definition • …	**Others** • 7,000+ tunable parameters

Parameter Name	Description	Default value
spark.driver.cores	Number of cores used by the Spark driver process	1
spark.driver.memory	Memory size for driver process	1 GB
spark.sql.shuffle.partitions	Number of tasks	200
spark.executor.cores	The number of cores for each executor	1
spark.files.maxPartitionBytes	Max number of bytes to group into one partition during file reading	128MB
spark.memory.fraction	Fraction for execution and storage memory. It may cause frequent spills or cached data eviction if given a low fraction	0.6

Spark parameter examples

Tuning

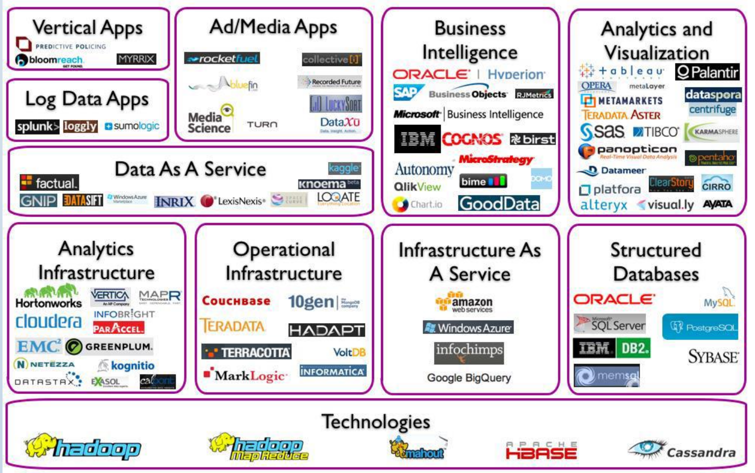
- ✓ Execution time: 0.1x ↓
- ✓ System resource utilization: 10x ↑
- ✓ Reduced errors: OOM, drive exhaustion, and timeout

Challenges and Objectives

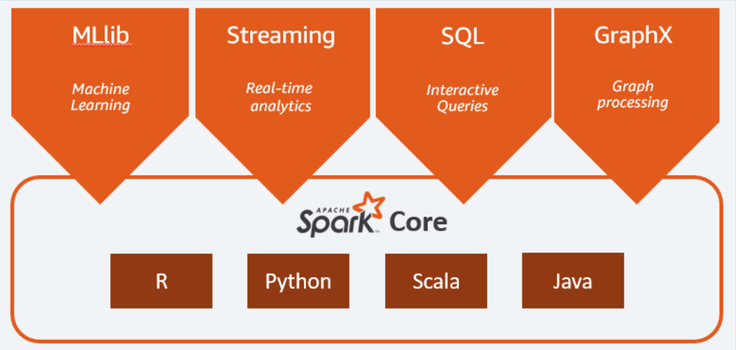
➢ Many parameters in a single system

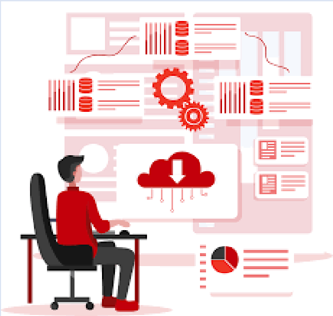
Symbol	Parameter	Groups
N_m	mapred.map.tasks	Map input split
N_r	mapred.reduce.tasks	Reduce output
s_{min}	mapred.min.split.size	Map input split
s_{max}	mapred.max.split.size	Map input split
B_m	io.sort.mb	Map output buffer
r_{rec}	io.sort.record.percent	Map output buffer
c	mapred.compress.map.output	Map output compr.
N_{copy}	mapred.reduce.parallel.copies	Reduce copy
N_{sf}	io.sort.factor	Reduce input
B_r	mapred.job.reduce.input.buffer.percent	Reduce input
S_{OB}	dfs.block.size	Reduce output
$N_{ms}(i)$	mapred.tasktracker.map.tasks.maximum	Set by HAC
$N_{rs}(i)$	mapred.tasktracker.reduce.tasks.maximum	Set by HAC

➢ A huge number of systems



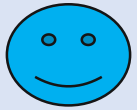
➢ Diverse workloads and system complexity





- ➢ Long tuning time
- ➢ Requires individual experience
- ➢ High labor costs
- ✓ Abundant tuning experiences

Manual tuning



- ✓ Short tuning time
- ✓ No need for individual experience
- ✓ Machine cost only
- ➢ Difficult to accumulate experience

Automatic tuning

Objectives: A-Tune uses AI to find the optimal parameters of a given workload and achieves the best application performance.

Related Work

➢ Tuning is a constant topic.

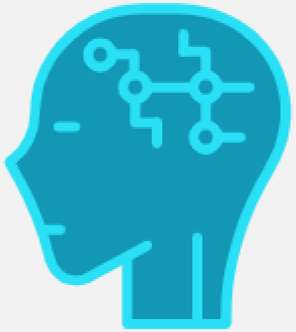
Method	Quality	Efficiency	Challenge 1	Challenge 2	Challenge 3
Manual tuning	Medium	Low	×	×	×
Statistical/Heuristic [iTuned SIGMOD '10, BestConfig SoCC '17]	Low	High	×	√	√
Traditional ML [OtterTune SIGMOD '17, RelM SIGMOD '20]	High	Medium	×	√√	√√
Deep learning/Reinforcement learning [iBTune VLDB '19, CDBTune SIGMOD '19, OtterTune VLDB '21]	High	Low	√	√√	√√

AI-Based Automatic Tuning

Full-stack coverage

Leverages a variety of technologies to simultaneously optimize several layers.

Intelligent exploration

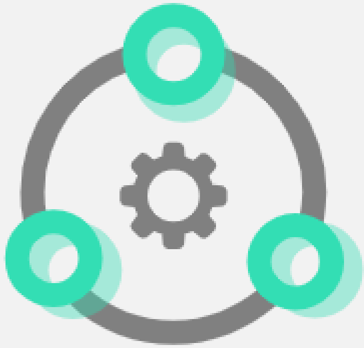


Explores tuning opportunities in an efficient and intelligent manner.

Target orientation

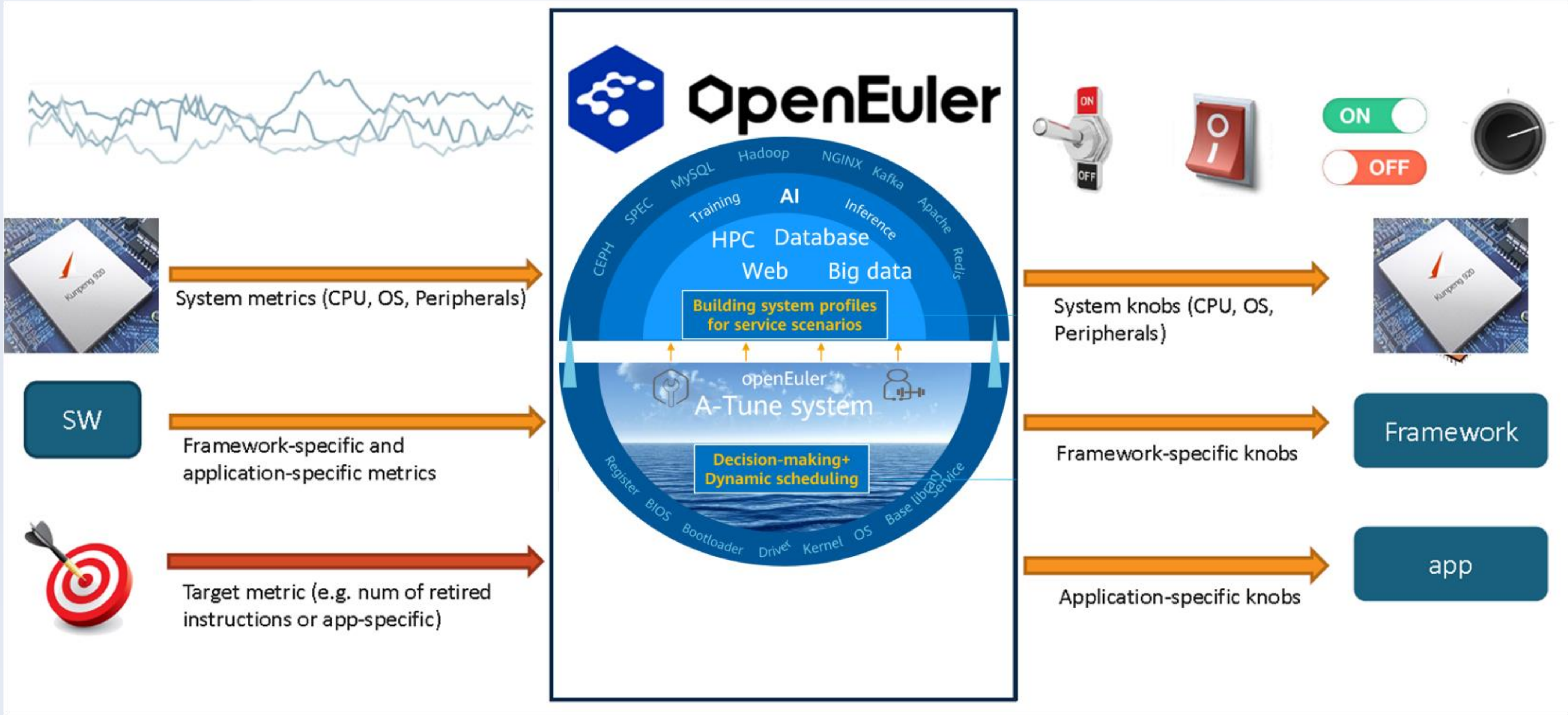
Tunes the system based on objectives and constraints.

Full automation



Conducts the entire tuning procedure automatically.

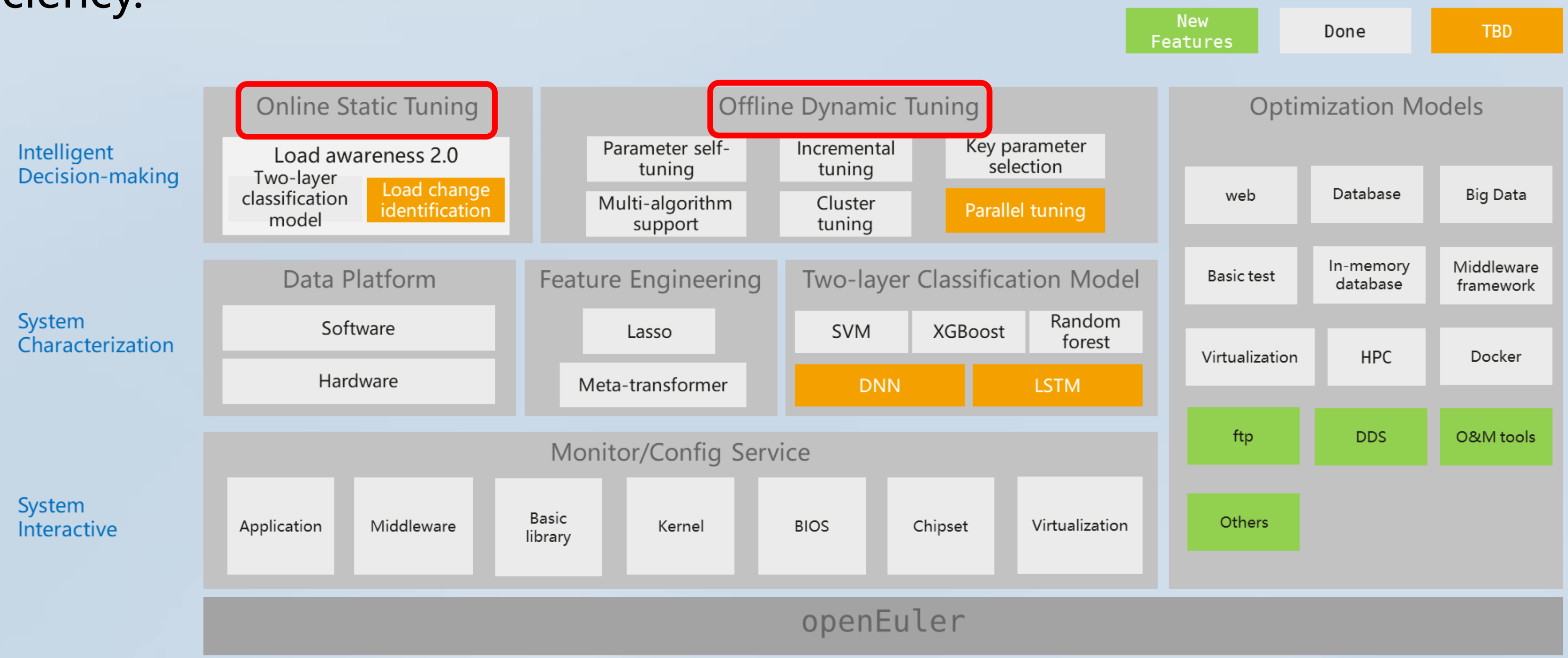
A-Tune: AI Enables the Optimal Performance of the System



A-Tune primarily focuses on automatic parameter tuning.

Architecture of A-Tune

A-Tune aims to analyze the resource usage (e.g., computing, storage, and network) of application workloads, support parameter tuning for mainstream applications in the industry, and improve tuning efficiency.

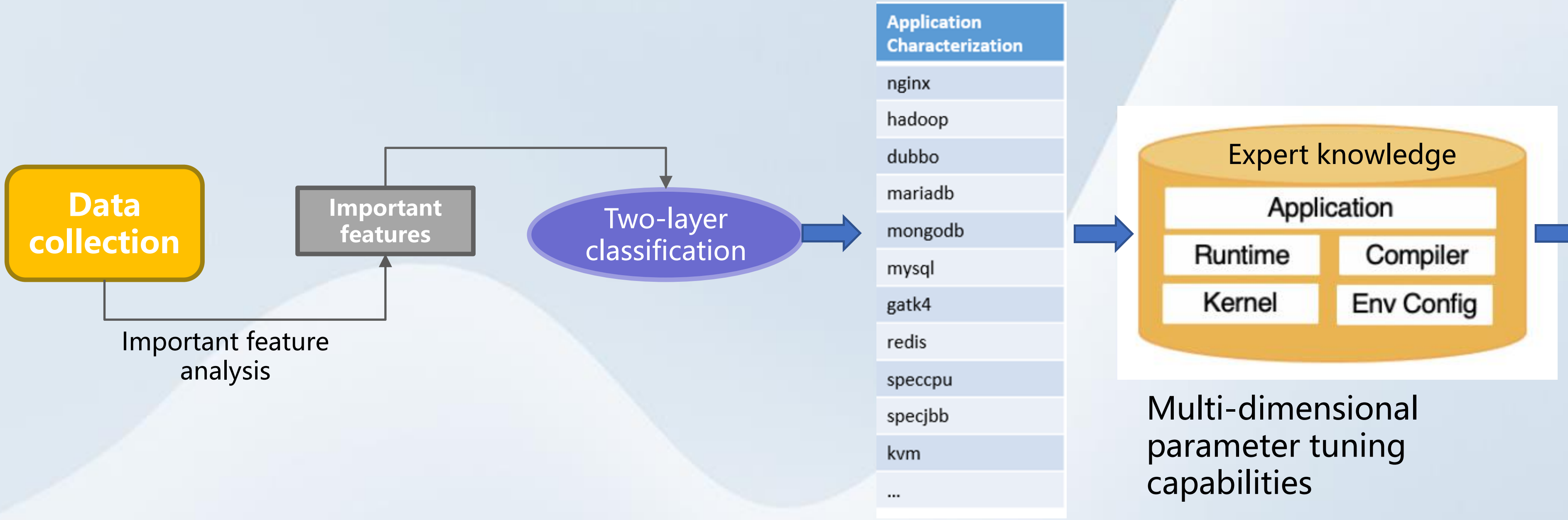


- Intelligent decision-making: online static tuning and offline dynamic tuning
- System characterization: data platform, feature engineering, training, and inference
- System interaction: system full-stack monitoring and configuration service

A-Tune: Online Static Tuning

Scenario: Common users and applications need to be online at all times.

Solution: Detects the current application workload, matches it to a known workload based on the classification model, and outputs empirical parameters.




```
[systemctl]
sysmonitor = stop
irqbalance = stop

[sysctl]
fs.file-max = 6553600
fs.suid_dumpable = 1
fs.aio-max-nr = 1048576
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.ip_local_port_range = 1024     65500
net.ipv4.tcp_max_tw_buckets = 5000
net.core.somaxconn = 65535
net.core.netdev_max_backlog = 262144
net.ipv4.tcp_max_orphans = 262144
net.ipv4.tcp_max_syn_backlog = 262144
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_synack_retries = 1
net.ipv4.tcp_syn_retries = 1
net.ipv4.tcp_fin_timeout = 1
net.ipv4.tcp_keepalive_time = 60
```

Empirical parameters

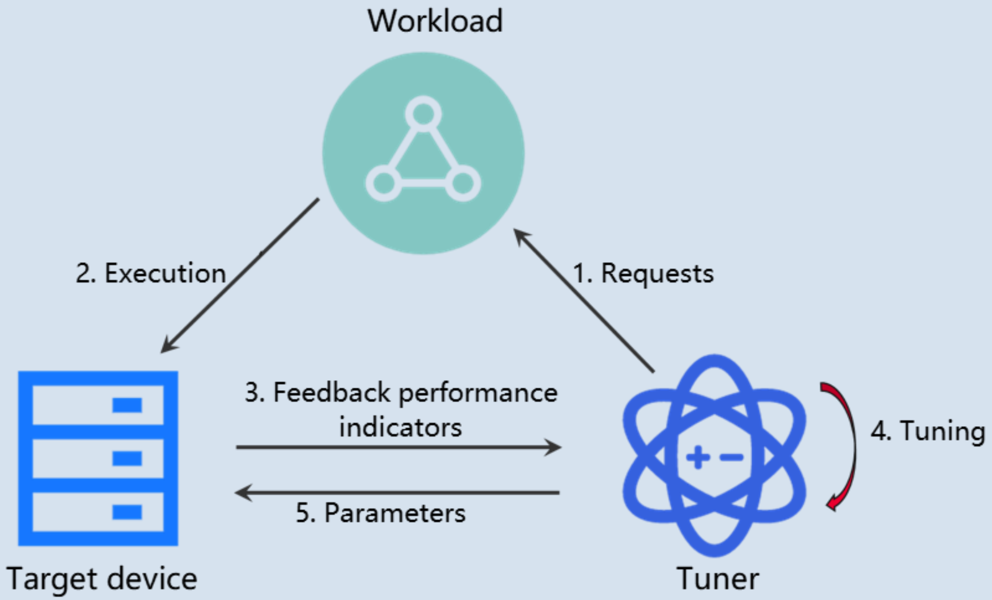
Key Technologies

- ✓ **Important feature analysis**: Automatically selects important features to characterize applications accurately.
- ✓ **Two-layer classification**: Accurately identifies the current workload.
- ✓ **Workload change detection**: Automatically identifies application workload changes and implements adaptive optimization.

A-Tune: Offline Dynamic Tuning

Scenario: Advanced users have high performance requirements.

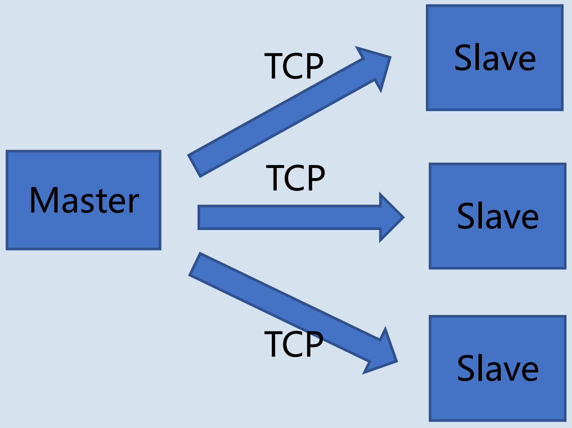
The tuner sets parameters for the target device, obtains the feedback performance indicators, and iterates continuously to obtain the optimal parameters.



10+ tuning models are supported:

- Gaussian process regression (GPR)
- Random forest (RF)
- Extremely randomized trees (ET)
- Gradient boosting regression tree (GBRT)
- ...

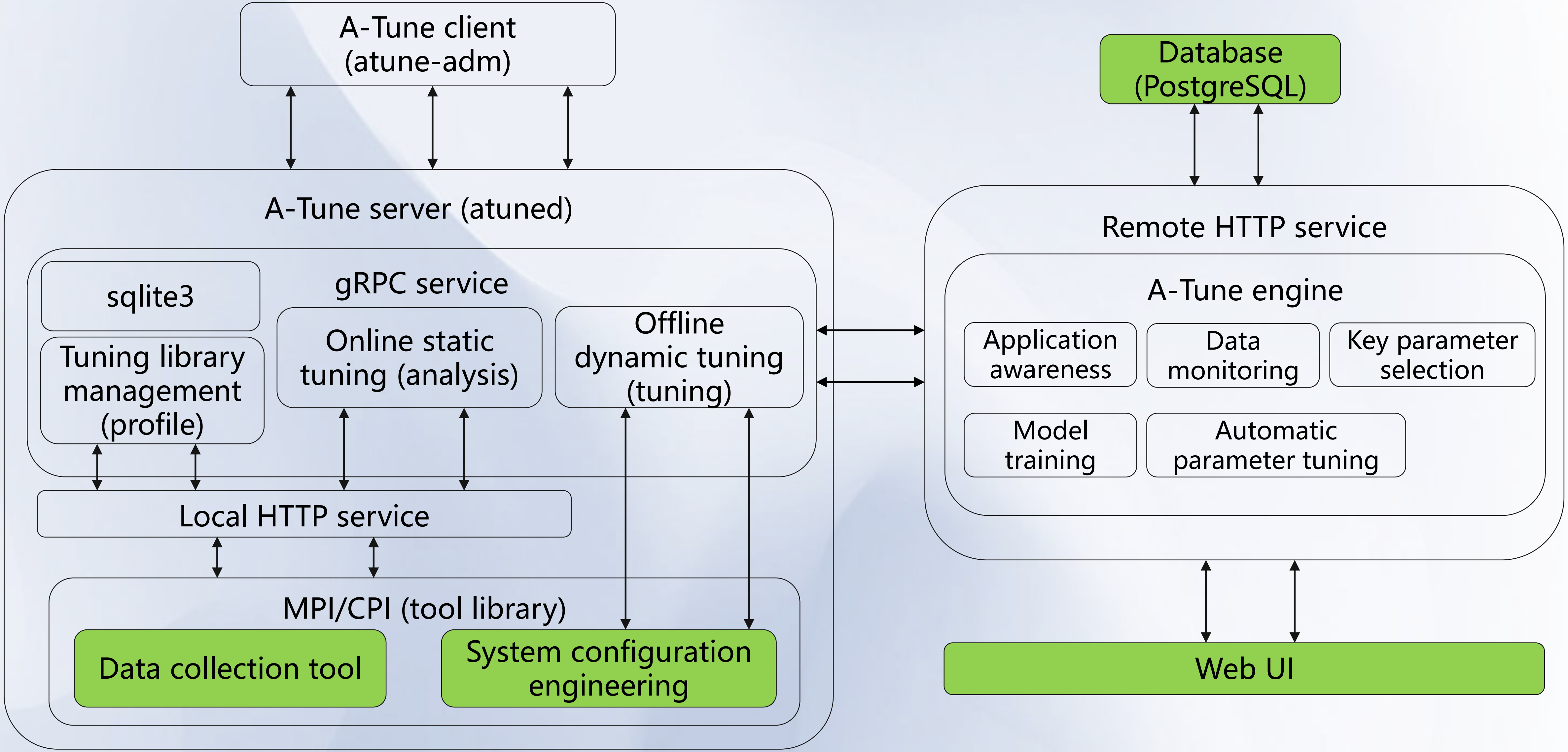
Cluster tuning support



Key Technologies

- ✓ **Important parameter selection**: Automatically selects important parameters to reduce search space and improve training efficiency.
- ✓ **Optimization algorithm construction**: Allows users to select the optimal algorithm based on application scenarios, parameter types, and performance requirements.
- ✓ **Knowledge base construction**: Adds the current workload characteristics and optimal parameters to the knowledge base to improve subsequent tuning efficiency.

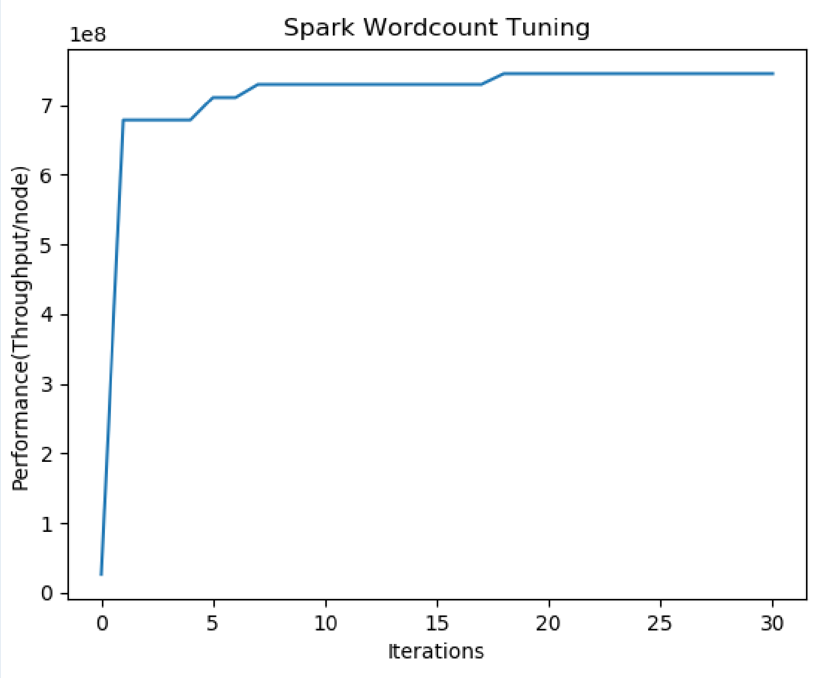
A-Tune Implementation Framework



Tuning Results

Parameter	Description	Default Value
spark.driver.cores	Number of cores to use for the driver process.	1
spark.driver.memory	Amount of memory to use for the driver process.	1g
spark.sql.shuffle.partitions	Number of tasks.	200
spark.executor.cores	Number of cores to use on each executor.	1
spark.files.maxPartitionBytes	Maximum number of bytes of a single partition when reading files.	128 MiB
spark.memory.fraction	Fraction of memory used for execution and storage.	0.5

Spark parameters



Big Data Scenario	Tuning Duration	Tuning Effect (Throughput/Node)
Tuning by professional engineers	15 days	1367254463
A-Tune offline dynamic tuning	3 days	1435561695

In microservice and big data scenarios, performance is improved by 30% over the default system configuration and 5% over the configuration optimized by professional engineers. The tuning efficiency is five times that of manual tuning.

Tuning Results

We optimized etcd with A-Tune, increasing performance by 10%+. The results were displayed in the openEuler application porting competition.

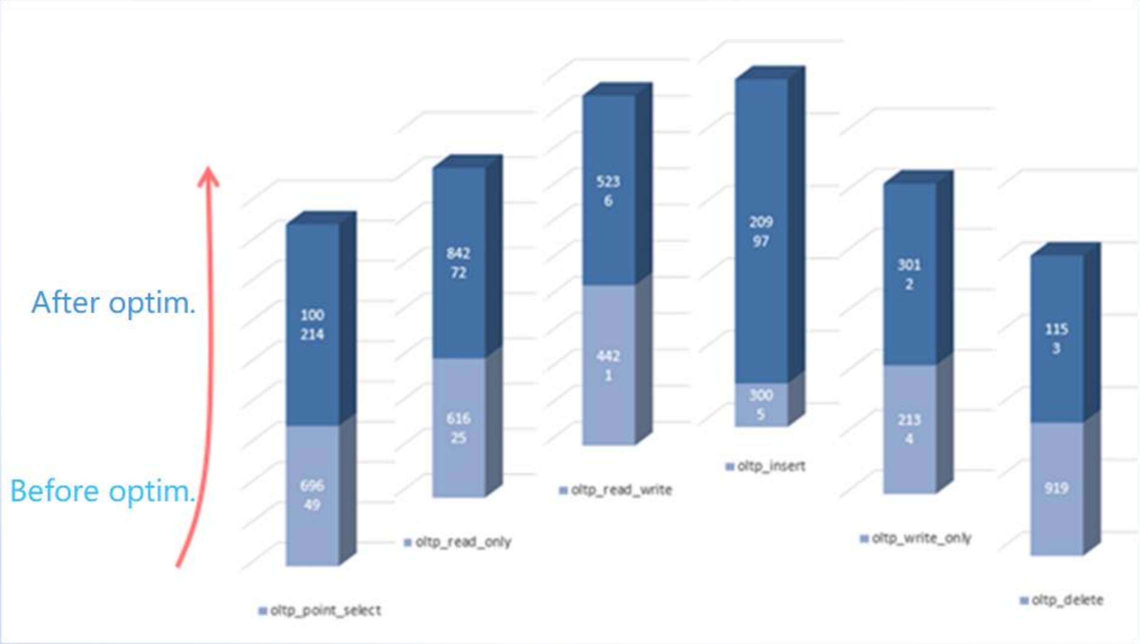
Write performance test:

Keys	Key Size	Value Size	Connections	Clients	Target etcd Server	Total (Before Optim.)	Total (After Optim.)	Performance Improvement
10,000	8	256	1	1	Leader	9.4018s	8.6203s	0.7815s
100,000	8	256	100	1,000	Leader	5.5742s	4.5548s	1.0194s
10,000	8	256	1	1	All members	9.9732s	8.5401s	1.4331s
100,000	8	256	100	1,000	All members	4.9520s	4.2021s	0.7499s

Read performance test:

Keys	Key Size	Value Size	Connections	Clients	Consistency (Linearization/ Serialization)	Total (Before Optim.)	Total (After Optim.)	Performance Improvement
10,000	8	256	1	1	Linearizable	5.9422s	5.6072s	0.335s
10,000	8	256	1	1	Serializable	2.7337s	2.6914s	0.0423s
100,000	8	256	100	1,000	Linearizable	2.2282s	2.0714s	0.1568s
100,000	8	256	100	1,000	Serializable	1.6616s	1.5518s	0.1098s

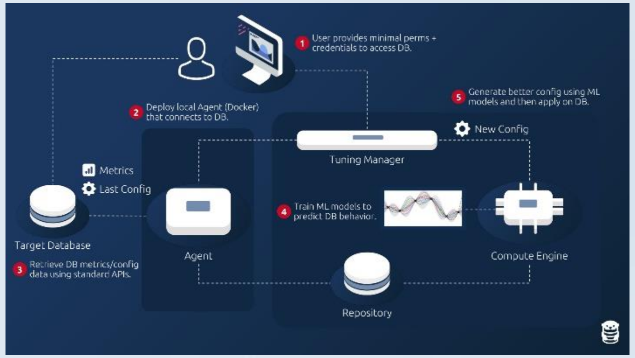
MySQL 8.0.25 is optimized in 6 different scenarios on openEuler that operates on VMs with 4 cores and 16 GB memory, as well as 32 cores and 64 GB memory. In the oltp_insert scenario, the throughput is improved by five times.



Prospect: A-Tune Architecture 3.0 Breaks Performance Bottlenecks

From offline to online

Industry trend: continuous performance tuning capabilities for online applications to deliver performance-as-a-service



OtterTune provides online parameter tuning for cloud databases.

Granulate focuses on resource queue tuning for online applications.

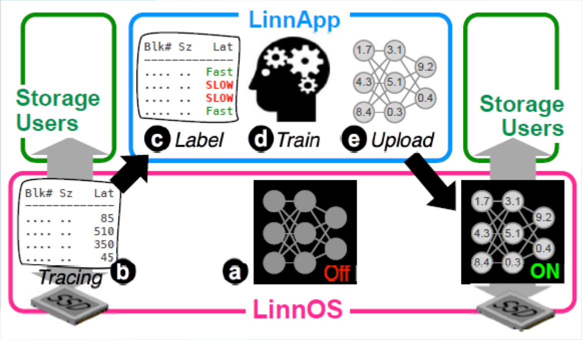
[1] Towards Dynamic and Safe Configuration Tuning for Cloud Databases. SIGMOD2022.
[2] HUNTER: An Online Cloud Database Hybrid Tuning System for Personalized Requirements. SIGMOD2022.
[3] An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems. VLDB2021.

Challenges:

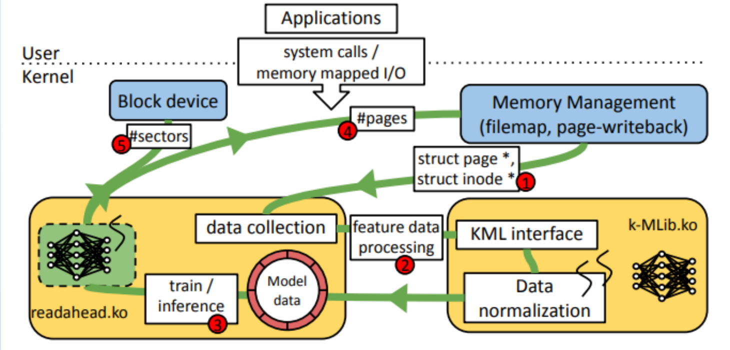
- Cold start: How do we deliver optimal performance in the absence of historical data when a new application goes online?
- Adaptability: How can the cost of online training be reduced when a model is trained against online application load that changes dynamically?
- Security: How do we keep the performance of the intermediate tuning result above the preset threshold to ensure high system availability during tuning?

From user mode to kernel mode

Industry trend: utilizing AI technologies to shift from conventional system configuration optimization to system design optimization



LinnOS uses AI to reconstruct the kernel block layer and predict I/O latency, boosting SSD array performance.



KML builds an AI model in the kernel space to implement load-aware data prefetching policies.

[1] LinnOS: Predictability on Unpredictable Flash Storage with a Light Neural Network. OSDI2020.
[2] KML: Using Machine Learning to Improve Storage Systems. arXiv2021.
[3] LiteFlow: towards high-performance adaptive neural networks for kernel datapath. SIGCOMM2022.

Challenges:

- Computing overhead: How can model complexity and performance be balanced with the limited computing resources available in the kernel space?
- Timeliness: How do we infer and execute the highly dynamic optimization policies in milliseconds?
- Universality: How do we construct a universal framework for different optimization types that involve data collecting and optimization algorithms for various kernel modules?

- A-Tune repository: https://gitee.com/openeuler/A-Tune

Thank you.

How to Engage

@openEuler
https://twitter.com/openEuler

r/openEuler
https://new.reddit.com/r/openEuler/

openEuler
https://www.youtube.com/@openeuler/

openEuler
https://www.linkedin.com/company/86315548/

Official website

LinkedIn newsletter

Join SIGs

Download

OpenEuler

Inclusive Architecture | Rapid Evolution | Reliable Supply Chain

Future of A-Tune

- Online tuning: real-time workload awareness + online transfer learning and incremental learning
- Tuning service: collaborates with cloud-native technologies to provide services for cloud scenarios
- Accelerated optimization based on expert experience
- Automatic parameter space generation