Apusic 金蝶天燕

生成时间：2026-07-13

Product Technical White Paper

金蝶Apusic分布式缓存

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

Apusic是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

1 Introduction

1.1 Background

Driven by both the rapid development of the digital economy and the in-depth advancement of information technology innovation (Xinchuang) policies, enterprise IT architectures are experiencing a comprehensive wave of domestic upgrade. As a core middleware in IT architecture, distributed cache plays a critical role in improving system response speed, alleviating database pressure, and supporting high-concurrency scenarios. Currently, digital transformation across various industries has entered deep waters, with business scales in key areas such as government services, financial transactions, e-commerce retail, and energy dispatch continuing to expand. Data volumes are growing exponentially, making high concurrency, low latency, and high availability core requirements for enterprise IT systems, and the strategic value of distributed cache has become increasingly prominent.

Meanwhile, the information technology innovation industry, as the core support for ensuring national information security and breaking foreign technology monopolies, has moved from policy guidance to large-scale implementation. Key sectors including government, finance, energy, and healthcare have explicitly required IT infrastructure, middleware, and application software to achieve comprehensive autonomous controllability. As the core hub connecting the application layer and data layer, the autonomous controllability level of distributed cache directly relates to the security, stability, and supply chain security of the entire IT architecture, becoming an indispensable key component in the domestic upgrade process.

However, the current domestic distributed cache market is still dominated by foreign open-source products (such as Redis) and commercial cache middleware. These products have many shortcomings in adapting to domestic enterprise IT architectures and meeting information technology innovation compliance requirements, unable to fully match the domestic transformation needs of China's key industries.

Against this background, developing an autonomous, controllable, secure and compliant, high-performance, easy-to-operate, and fully adapted distributed cache middleware for domestic environments has become an urgent need to ensure enterprise digital transformation and information technology innovation implementation. Kingdee Tianyan, relying on years of enterprise middleware development experience, has independently developed the Kingdee Apusic Distributed Cache software product, filling the gap in the high-end domestic distributed cache market and helping enterprises build autonomous and controllable distributed application architectures.

1.2 The Role of Cache in Application Architecture

In enterprise-level distributed application architectures, application systems achieve high availability and high-concurrency support through cluster deployment. However, as business scales expand, data storage and access pressures continue to rise. Traditional databases (relational databases, distributed databases) are prone to response delays and performance bottlenecks in high-concurrency read/write scenarios.

Distributed cache, as a core means of architecture optimization, runs through the entire chain of distributed systems. Its core role is reflected in five aspects, becoming both the "performance engine" and "security shield" for stable operation of distributed architectures:

1.2.1 Improving System Response Speed and Optimizing User Experience

Distributed cache stores frequently accessed and hotspot data in memory. Applications can read data directly from cache without frequent access to the underlying database, reducing data access latency from milliseconds to microseconds and significantly improving system response efficiency.

1.2.2 Alleviating Database Pressure and Ensuring Data Layer Stability

In distributed architectures, the underlying database is often the performance bottleneck. High-frequency access requests can cause database connection exhaustion, excessive CPU load, and even crash risks. Distributed cache can intercept over 80% of high-frequency read requests, reducing database access pressure, preventing database overload from high-concurrency requests, and ensuring stable operation of the data layer. Additionally, cache can implement read-write separation adaptation, directing read requests to cache and synchronizing write requests to the database, further balancing database load and improving the throughput of the entire data layer.

1.2.3 Supporting High-Concurrency Scenarios and Enhancing Business Carrying Capacity

As enterprise digital transformation accelerates, high-concurrency scenarios are becoming normalized. In such scenarios, concurrent access volumes can be tens or even hundreds of times higher than daily levels, which traditional architectures cannot handle. Distributed cache, through its memory storage and cluster deployment characteristics, can easily support hundreds of thousands of concurrent accesses per second, meeting business access requirements in high-concurrency scenarios, ensuring continuous and stable business operation, and avoiding system crashes from concurrent overload.

1.2.4 Achieving Data Consistency Supplement and Optimizing Architecture Fault Tolerance

Distributed cache supports multiple caching strategies and can synchronize data with underlying databases, ensuring consistency between cached data and database data. Meanwhile, in distributed architectures, if a database node goes down, cache can temporarily replace the database to provide hotspot data access services, avoiding business interruption, improving architecture fault tolerance and availability, and buying time for database failure recovery.

1.2.5 Simplifying Architecture Complexity and Reducing Operation Costs

Distributed cache can achieve hotspot data aggregation, data preprocessing, and other functions, reducing interaction logic between applications and databases, simplifying the design and implementation complexity of distributed architectures. Additionally, commercial products of domestic distributed cache middleware have complete cluster management, monitoring alerts, and automated operation capabilities, reducing operation

complexity for operation personnel and lowering operation costs, helping enterprises achieve lightweight operation of distributed architectures.

In summary, distributed cache bears the core mission of "accelerating, decompressing, and stabilizing architecture" in distributed architectures. It is not only a key means to improve system performance but also the core support for ensuring high availability, high concurrency, and high stability of distributed architectures. Its performance, stability, and security directly determine the operation quality of the entire distributed application system.

1.3 Industry Demand Analysis

Currently, Redis open-source products, foreign commercial cache middleware, and in-memory grid middleware products used in domestic key sectors are increasingly exposing many pain points due to limitations in open-source licensing, technical architecture, operation capabilities, and domestic adaptation. There are clear gaps with the core needs of domestic enterprise digital transformation and information technology innovation implementation, which can be summarized in four aspects:

1.3.1 Autonomous Control and Compliance Requirements in Information Technology Innovation Context

As information technology innovation policies advance deeply across industries, key sectors such as government, finance, and energy have explicitly required IT infrastructure to achieve autonomous controllability. However, traditional Redis open-source products rely on foreign communities for core code. In recent years, the Redis open-source community has frequently modified its open-source licenses (RSALv2, SSPLv1, AGPLv3) for open-source commercialization and ecosystem interest games, creating compliance, security, and supply chain risks that cannot meet autonomous control and security compliance requirements.

1.3.2 Enterprise-Level Commercial Operation

Traditional Redis open-source products lack professional commercial operation support. Operation relies on command line with high complexity, making it difficult to support large-scale and standardized operation needs of large enterprises. Meanwhile, the lack of complete migration tools, monitoring alerts, and automated operation capabilities leads to high operation costs. There is an urgent need to enhance commercial operation features to improve operation efficiency.

1.3.3 Full-Stack Domestic Adaptation

Current enterprise IT architectures are gradually transforming toward domestic models. Domestic chips, operating systems, databases, and other infrastructure have been widely applied. However, traditional products have poor adaptability to domestic hardware and software, with performance bottlenecks and insufficient stability, unable to integrate into full-stack domestic IT architectures. There is an urgent need for a cache product that comprehensively adapts to domestic hardware and software.

1.3.4 Performance Improvement in High-Concurrency Scenarios

As enterprise business digital transformation accelerates, high-concurrency scenarios such as e-commerce promotions, government service peaks, and financial transactions are increasing. The Redis community open-source version has performance bottlenecks and data consistency issues in high-concurrency and large-scale deployment scenarios. It needs to combine business scenarios for performance optimization and high-availability design to support enterprise high-load business needs.

Against this background, Kingdee Tianyan has independently developed the Kingdee Apusic Distributed Cache software product, meeting business and technical needs for autonomous control, security compliance, high performance, easy operation, and ecosystem compatibility. It has gradually become a core component for government, enterprises, and public institutions to build enterprise-level autonomous and controllable distributed application architectures.

1.4 Target Audience

This white paper focuses on the technical architecture, core functions, technical characteristics, typical application scenarios, and commercial value of Kingdee Apusic Distributed Cache software product. For product installation, deployment, operation, development, and maintenance, please refer to the supporting product standard manual documentation.

Target readers of this white paper document:

- IT Information Leaders: Understand the value, application prospects, and product advantages of cache middleware products to support procurement decisions.
- Technical Selection Personnel: Master product technical architecture, core features, and performance indicators to evaluate product compatibility with enterprise IT architecture.
- Software Development Engineers: Familiarize with product functions, SDK support, and ecosystem compatibility features to support development integration and business migration.
- IT Operations Personnel: Understand product operation characteristics, management capabilities, and deployment adaptation to support product deployment and daily operations.

1.5 Terminology Definition

To ensure consistency and accuracy in document reading, the core terminology and definitions in this document are as follows:

- Distributed Cache: A cache technology that stores data across multiple nodes for shared access by multiple clients, enabling distributed data storage and high-concurrency access to improve system response speed.
- Domestic Adaptation: Product optimization and adjustment for domestic chips, operating systems, databases, and other IT infrastructure to ensure stable operation and performance standards in domestic

environments.

- National Cryptography Algorithm: Cryptographic algorithms designated by the State Cryptography Administration, including SM2 (signature and encryption), SM3 (hash digest), SM4 (block encryption), etc., used to ensure data security and compliance.
- SDK: Software Development Kit. Provides standardized interfaces and tools for developers, simplifying the development and integration process between products and business systems.
- Redis Ecosystem: Development system formed around Redis products, including client tools, development frameworks, script tools, monitoring tools, etc., supporting Redis development, deployment, and operations.

2 Product Overview

2.1 Product Introduction

Kingdee Apusic Distributed Cache Software (Apusic In-Memory Data Cache, abbreviated as AMDC) is an enterprise-level high-performance distributed cache middleware that is 100% compatible with Redis protocol and development ecosystem. Existing business applications can be seamlessly migrated without modification. AMDC features core capabilities of high-performance millisecond response, high-availability multi-replica fault tolerance, high-scalability elastic expansion, and refined web visual management functions. It provides secure and reliable high-frequency data storage and retrieval support for critical businesses in high-concurrency and distributed scenarios, reducing database pressure and improving overall system throughput.

AMDC focuses on key sectors including government, finance, energy, internet, and enterprise-level applications, providing high-availability, high-performance, high-security, easy-to-operate, and full-stack adapted cache solutions. It fully adapts to domestic chips such as Kunpeng and Feiteng, and domestic operating systems such as Kylin and OpenEuler, achieving end-to-end domestic deployment and helping enterprises quickly complete domestic replacement of the cache layer in IT architectures.

2.2 Product Architecture

The AMDC product architecture is shown in the figure below, consisting of three main parts: cache core, web console application, and operation management related command-line tools.

Product Modules and Components:

1. Cache Core: Responsible for processing core business logic of the cache. This layer includes core modules such as data structure processing, cache strategy, transaction management, cluster coordination, lock mechanism, publish-subscribe, and Lua script execution. Optimizing core algorithms and execution efficiency, reducing lock contention, and improving processing capability in high-concurrency scenarios; meanwhile integrating domestic security design, supporting national cryptography algorithm integration and security management, ensuring autonomous controllability of core functions.
2. Web Console: The core module supporting product commercial operation features, providing standardized and visual operation tools and capabilities for operation personnel. This layer includes sub-modules such as web management platform, migration tools, automated operations, monitoring alerts, and log management, responsible for the full lifecycle management of product deployment, monitoring, operations, and troubleshooting; providing standardized operation interfaces, supporting integration with existing enterprise operation systems, achieving standardization and automation of operation processes, and reducing operation complexity.
3. Command Line Tools: AMDC provides rich tools for development and operation engineers to use, manage, and operate AMDC clusters.

Table 1: AMDC Tool List

Tool	Chinese Name	Description
amdc-cli	AMDC Command Line Client	Used to connect to cache core service and perform data read/write and management operations.
amdc-check-rdb	RDB File Verification and Repair Tool	Used to check the integrity and validity of RDB snapshot files, troubleshoot file corruption issues, and repair partially corrupted RDB files.
amdc-check-aof	AOF File Verification and Repair Tool	Used to check the integrity and syntax correctness of AOF files, repair AOF file corruption caused by unexpected interruption, ensuring AMDC can normally load AOF files to recover data during startup.
amdc-conv-conf	Configuration File Conversion Tool	V2.0.4 and later versions have major upgrades in configuration file format and parsing. This tool can smoothly migrate configurations from old version configuration files, improving the efficiency and accuracy of version migration.
amdc-benchmark	AMDC Official Performance Testing Tool	Built-in basic performance testing tool that tests AMDC performance indicators in different scenarios, including QPS, response time, concurrent connections, etc., helping operation personnel and

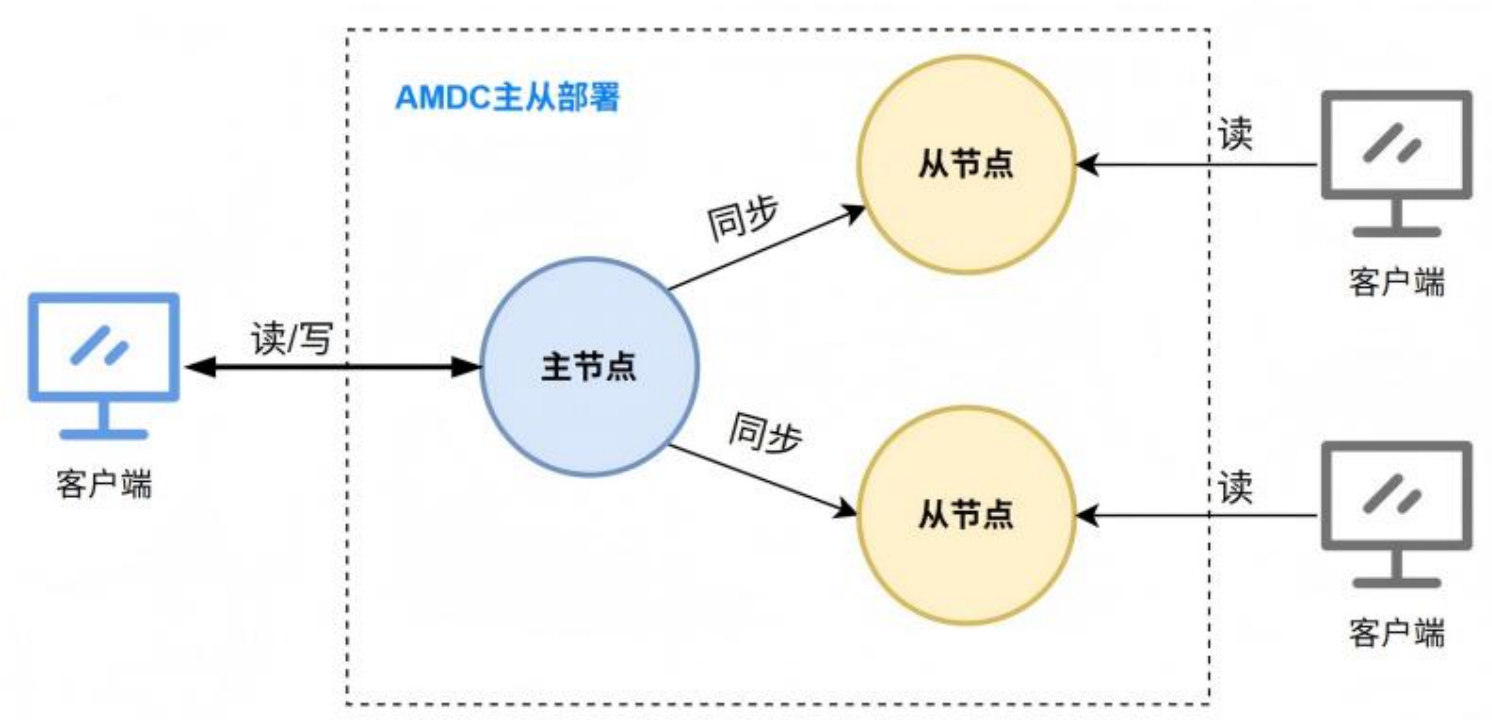
		developers evaluate AMDC service carrying capacity and optimize configuration parameters.

2.3 High Availability Architecture

AMDC provides a comprehensive high availability architecture for business continuity, supporting three high availability deployment modes: master-slave mode, sentinel mode, and cluster mode. It achieves data multi-replica storage, automatic failover, cluster scaling, and other functions, ensuring 7×24 hours uninterrupted operation of the product, supporting enterprise high-concurrency and high-stability business needs.

2.3.1 Master-Slave Mode

Master-slave replication mechanism with optimization upgrades, achieving real-time synchronization of master node data to slave nodes, building a one-master-multi-slave deployment architecture, improving system read performance and data availability.



The above figure shows a typical master-slave replication deployment mode of AMDC, used to achieve data redundancy, read-write separation, and high availability capabilities.

Core Node Description

- Master Node: Responsible for processing all write operations and some read operations, is the authoritative source of data.

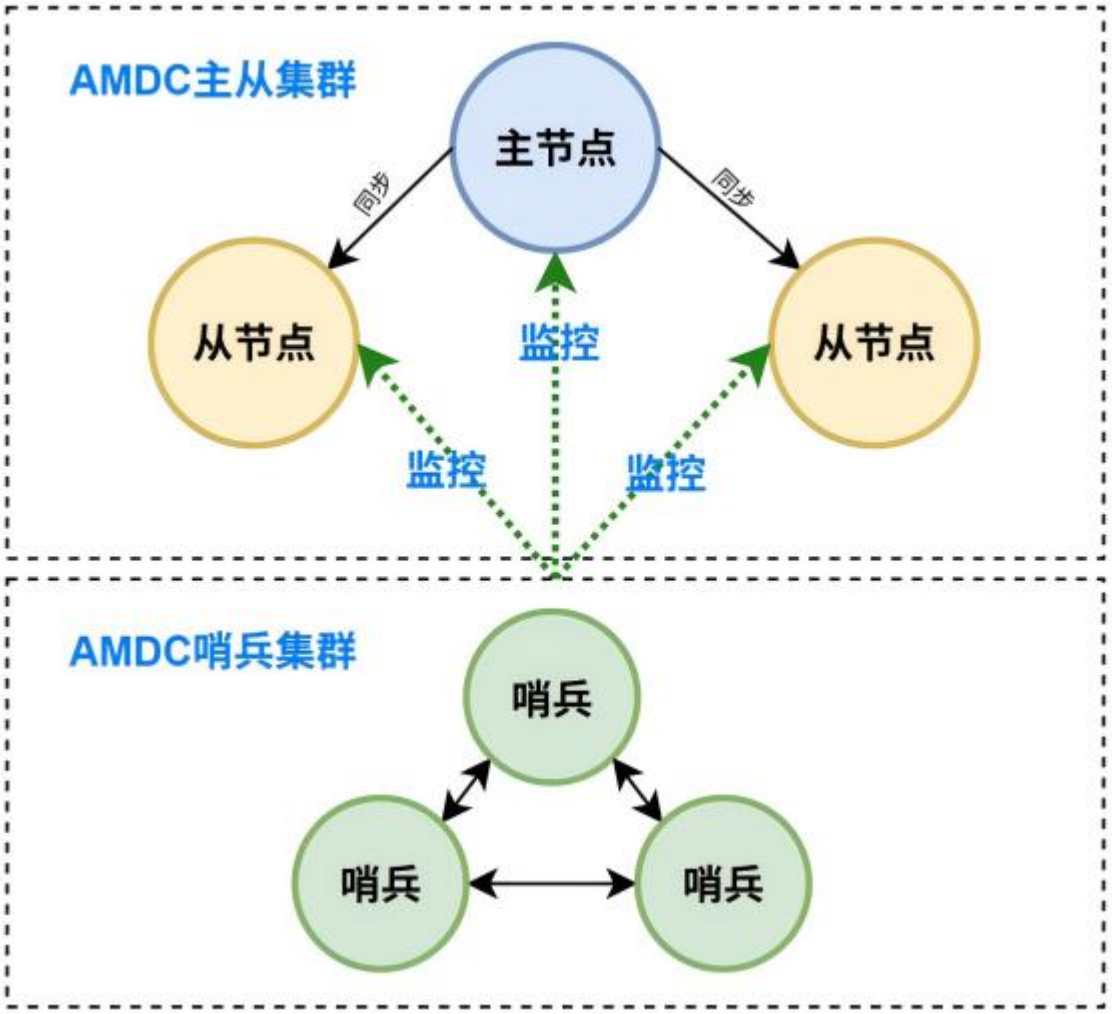
- Slave Nodes: Receive and replicate master node data changes in real time through the "synchronization" mechanism, maintaining data consistency. Slave nodes are mainly used to share read load and improve overall system throughput.
- Clients: Clients connect to the master node to execute "read/write" operations; clients connect to respective slave nodes to execute only "read" operations, reflecting read-write separation design.

Master-Slave Architecture Advantages

- Improved Readability: Read requests are distributed to multiple slave nodes, avoiding single point bottlenecks;
- Data Backup: Slave nodes serve as hot standby, can quickly take over when master node fails (requires sentinel or cluster mechanism coordination);
- Flexible Expansion: Slave nodes can be dynamically added according to business needs to cope with read traffic growth.

2.3.2 Sentinel Mode

The product has a built-in Sentinel module, supporting sentinel cluster deployment. The core function is to achieve automatic detection and automatic failover of master node faults without manual intervention, ensuring high availability of the system, solving the system unavailability problem caused by master node failure in master-slave replication architecture.



AMDC Sentinel Cluster Monitoring Mechanism

- Each sentinel instance performs heartbeat detection (marked as "monitoring") on the master node and all slave nodes through green dashed arrows, real-time sensing of their running status.
- If a node has no response exceeding the threshold, the sentinel will mark it as "subjectively down" and reach "objectively down" consensus through negotiation with other sentinels.

Automatic Failover

- When the master node is judged as unavailable, the sentinel cluster will automatically elect a new master node (usually selecting the most suitable one from slave nodes).
- The original slave nodes will be reconfigured as slaves of the new master node, and client connections will be updated accordingly (requires client support for sentinel protocol or use of proxy layer).

High Availability Guarantee

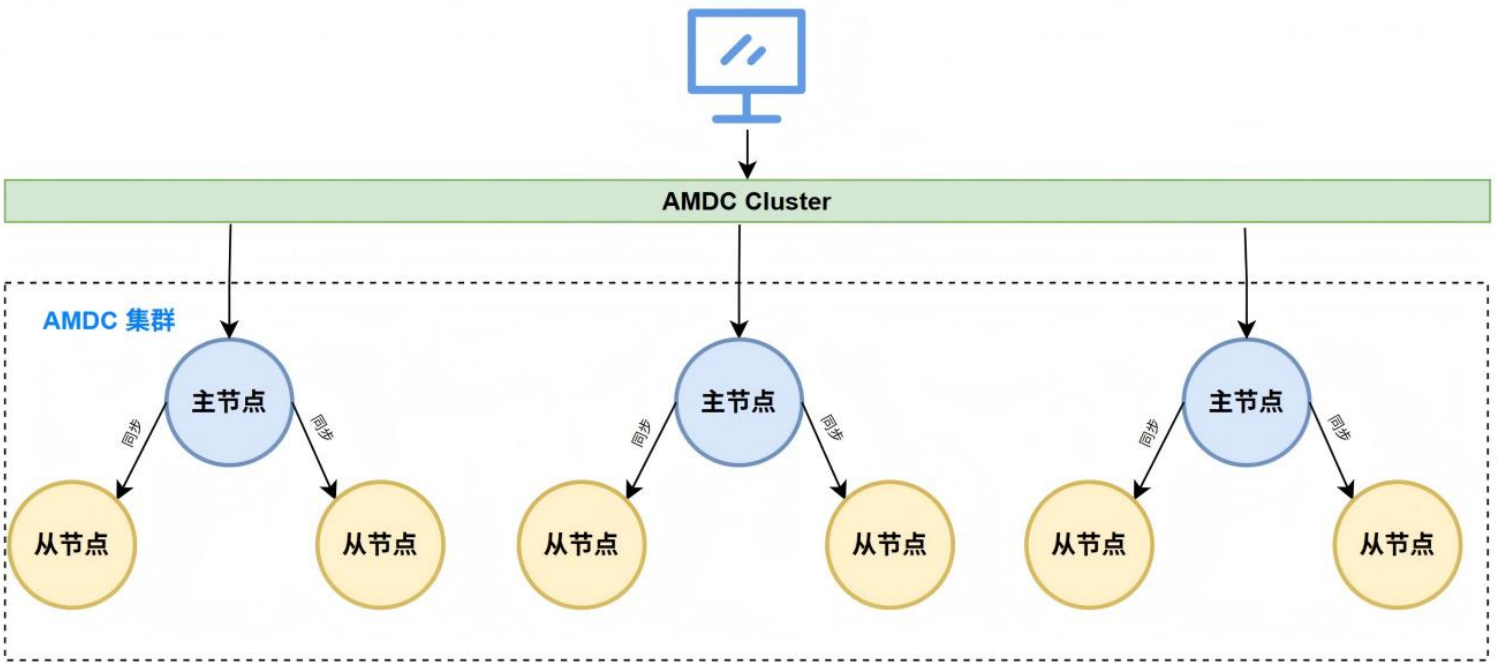
- The sentinel cluster itself has fault tolerance capability - even if individual sentinels go down, as long as a majority survives, fault discovery and failover can still be completed.
- It is recommended to deploy an odd number of sentinels (such as 3) to avoid split-brain problems.

Sentinel cluster deployment supports multiple sentinel node deployment, avoiding sentinel node single point of failure. The recommended number of sentinel nodes is odd (3 or more), improving the accuracy and reliability of fault judgment.

Supports dynamic scaling of sentinel nodes without restarting the system, adapting to system scale changes; log visualization supports sentinel log visual display, facilitating operation personnel to monitor sentinel running status and failover process, quickly troubleshooting problems.

2.3.3 Cluster Mode

The product supports Cluster mode, adopting Hash Slot sharding mechanism, achieving distributed data storage and load balancing, adapting to large-scale, high-concurrency cache scenarios, solving performance bottlenecks and capacity limitations of single-machine and master-slave architectures.



Data Sharding

- The client calculates the hash slot based on key, then the cluster routes to the corresponding master node.
- Achieves horizontal data splitting, breaking through single-machine memory and performance bottlenecks.

Replication

- Each master node asynchronously "synchronizes" its data to its two subordinate slave nodes.
- Slave nodes can be used for read request distribution (requires configuring cluster-slave-validity-factor and client support).

Automatic Failover

- If a master node goes down, any of its slave nodes will automatically be promoted to new master node (requires majority master node agreement).
- Original slave node relationships are rebuilt, and the cluster self-heals, ensuring service continuity.

Decentralized Architecture

- All master nodes collaborate equally, jointly maintaining cluster metadata (slot mapping table).
- No dependency on external coordination components (such as ZooKeeper), simplifying operation complexity.

Cluster Deployment Architecture Advantages

- Strong Horizontal Scaling Capability: Storage capacity and throughput can be linearly improved by increasing the number of master nodes
- High Availability: Master-slave redundancy + automatic failover, single point of failure does not affect overall service
- Read-Write Separation Potential: Slave nodes can bear part of read requests (requires client or proxy support)
- Native Support: Redis has built-in Cluster mode, no additional middleware required (optional proxy to optimize experience)
- Suitable for Large Data Volume Scenarios: Applicable to TB-level cache, session management, real-time counting, and other high-frequency access businesses

2.4 Product Operating Environment

2.4.1 Software and Hardware Environment

The product fully adapts to domestic software and hardware environments while being compatible with mainstream open-source software and hardware environments, ensuring stability and performance in different IT architectures. The specific adaptation environment list is as follows:

Table 2: AMDC Adaptation Environment

Category	Adaptation and Support Items
CPU Architecture	x86_64, ARM64 (Kunpeng/Feiteng)
Operating System	Galaxy Kylin, UOS, NeoKylin, CentOS 7+, Ubuntu 18.04+
Container Platform	Docker, Kubernetes
Installation Method	tar.gz / RPM / Docker Image

2.4.2 Deployment Resource Requirements

Table 3: AMDC Deployment Resource Requirements

Deployment Mode	Installation Content	Hardware Specifications (CPU/Memory/Disk)	Number of Servers
Standalone	AMDC Console, AMDC Service	8 cores/16G/100G	1
Master-Slave	AMDC Console, AMDC Service	8 cores/16G/100G	2+
Sentinel	AMDC Console, AMDC Service	8 cores/16G/100G	3+
Cluster	AMDC Console, AMDC Service	8 cores/16G/100G	3+

3 Functions and Technical Features

The core functions of Kingdee Apusic Distributed Cache product cover the full lifecycle management of cache data, high availability assurance, security compliance, operation management, development integration, and other full scenarios, adapting to the diverse needs of government, finance, energy, internet, and other industries.

3.1 Core Functions

3.1.1 Data Types

The product supports rich data types, including basic core types and extended types, adapting to diverse data storage needs of business systems. Meanwhile, performance optimization is performed for each data structure to improve read/write efficiency in high-concurrency scenarios.

Table 4: Data Types Supported by AMDC

Data Type	Chinese Name	Type Description
String	String	Can store strings, numbers, binary data (such as images), maximum 512MB
List	List	Doubly linked list structure, supports addition/deletion at both ends, index-based access, can achieve ordered and duplicate collections
Hash	Hash (Hash)	Key-value pair collection, suitable for storing objects (such as user information), can operate fields individually, saving memory
Set	Set	Unordered, unique string collection, supports intersection, union, difference operations
Sorted Set	Sorted Set	Ordered, unique collection with score, sorted by score, supports range queries
Bitmap	Bitmap	Implemented based on String, operates by bit, uses bit to represent state (0/1), extremely memory-saving
HyperLogLog	Cardinality Statistics	Extremely small memory (about 12KB) to count the number of unique elements in massive data
Geo	Geospatial	Stores latitude and longitude, supports distance calculation, range query, nearby people, places, etc.
Stream	Stream	Supports message publishing, subscription, consumption confirmation, message backtracking, etc., adapting to lightweight message passing

		scenarios
Bitfield	Bitfield	Implemented based on String, reads and writes multiple bit fields, supports integers (8/16/32 bit)

All AMDC data types are stored as key-value pairs, where the key is always a string, and the value corresponds to the above different types, all supporting atomic operations.

3.1.2 Operation Commands

AMDC provides full instructions for cache data CRUD, batch operations, atomic operations, transaction control, publish-subscribe, etc. Supports operations corresponding to String, Hash, List and other type data. Developers can continue using original Redis command usage habits without additional learning, achieving zero-cost onboarding. It provides efficient and convenient basic support for cache data management and business logic implementation in various industry scenarios, while ensuring atomicity and high response speed of command execution, adapting to high-concurrency business needs.

Specific operation commands include:

1. Data Operations: Supports SET, GET, DEL, INCR, DECR and other instructions, achieving basic operations such as add, query, delete, increment, and decrement of data.
2. Batch Operations: Supports MSET, MGET, DEL and other batch instructions, reducing network interaction times, improving batch data processing efficiency.
3. Atomic Operations: Supports INCRBY, DECRBY, SETNX and other atomic instructions, avoiding data inconsistency in concurrent scenarios.
4. Cache Expiration Management: Supports EXPIRE, PEXPIRE and other instructions, customizing data expiration time, supporting automatic cleanup of expired data, optimizing memory utilization.

> *Note: For more detailed introduction of operation commands, please refer to the relevant chapters of "Kingdee Apusic Distributed Cache Development Manual".*

3.1.3 Data Persistence

Persistence is one of the core basic capabilities of AMDC product, supporting RDB data snapshot and AOF log two core mechanisms with performance optimization, effectively ensuring the integrity and recoverability of cache data after node failure or restart. The RDB mechanism can generate memory data snapshots on demand, adapting to large-scale data backup scenarios. The AOF mechanism records every write operation instruction in real time, achieving precise data recovery.

Users can flexibly choose single mechanism or hybrid mode, balancing data security and operation flexibility, fully adapting to enterprise-level data persistence needs.

3.1.4 Data Eviction Strategy

AMDC supports cache data eviction. When cache data memory reaches the preset upper limit, it automatically filters and deletes invalid or low-value data, releasing memory space, avoiding memory overflow, ensuring stable operation of cache service, while improving effective data cache hit rate and optimizing system performance.

AMDC supports multiple flexible eviction strategies, including LRU, LFU, volatile-lru, allkeys-lru, volatile-ttl, volatile-random, allkeys-random, noeviction. The default eviction strategy is noeviction, which rejects all write operations when memory is insufficient and only allows read operations, adapting to most enterprise basic cache scenarios.

Table 5: Cache Data Eviction Strategies

Strategy	Chinese Name	Strategy Description
LRU	Least Recently Used	Prioritizes eviction of data with the lowest usage frequency in the recent period, suitable for most conventional cache scenarios
LFU	Least Frequently Used	Prioritizes eviction of data with the fewest access times in a period, more precisely screening low-value cache data
volatile-lru	Expired Key LRU	Executes LRU eviction strategy only for keys with expiration time set, keys without expiration time do not participate in eviction
allkeys-lru	All Keys LRU	Executes LRU eviction strategy uniformly for all keys in cache, regardless of whether expiration time is set
volatile-ttl	Expired Key Shortest TTL	Prioritizes eviction of data with shortest remaining expiration time for keys with expiration time set
volatile-random	Expired Key Random	Randomly selects data to evict from keys with expiration time set, efficient operation, no need to filter
allkeys-random	All Keys Random	Randomly selects data to evict from all keys in cache
noeviction	No Eviction	Rejects all write operations when cache memory is insufficient, only allows read operations, avoiding accidental deletion of valid data. Note: This strategy is the default.

3.1.5 Transaction Support

Supports Multi/Exec/Discard/Watch and other transaction instructions, achieving atomic execution of multiple cache operations, either all succeed or all fail, ensuring data consistency. Optimizes transaction execution efficiency, reduces transaction blocking time, adapting to transaction consistency needs in high-concurrency scenarios.

3.1.6 Publish-Subscribe

Supports PUBLISH, SUBSCRIBE, PSUBSCRIBE and other instructions, achieving message publishing and subscription, supporting channel subscription and pattern subscription, adapting to scenarios such as message notification and real-time push. Optimizes message distribution efficiency, reducing message latency.

3.1.7 Lua Script Support

Supports EVAL, EVALSHA and other instructions, allowing developers to customize Lua scripts to implement complex cache operation logic. Script execution is atomic, reducing network interaction times and improving business processing efficiency; optimizes Lua script execution engine, improves script execution speed, supports script caching to avoid recompilation.

3.1.8 Pipeline

Supports batch sending of multiple requests to the server, reducing network round-trip times and improving batch operation throughput.

3.2 Security

The product builds a full-process, multi-level security protection system. On the basis of national cryptography algorithm support, it enhances identity authentication, permission control, data security, vulnerability protection and other functions, ensuring product operation security and data security, meeting enterprise-level security compliance needs.

3.2.1 Authentication and Authorization

The product provides strict identity authentication and permission control mechanisms to prevent illegal access and unauthorized operations, ensuring product operation and data security. Specific functions include:

1. Identity Authentication: Supports password authentication and key authentication methods. Password authentication supports password complexity control (such as password length, character type, periodic change), passwords are encrypted for storage to prevent password leakage; key authentication supports SM2 key authentication, improving identity authentication security, adapting to high security level requirements.
2. Permission Control: Adopts fine-grained permission management model, distinguishing different roles such as administrators, operation personnel, and ordinary users, assigning clear operation permissions to each role, limiting operation scope for different roles.
3. Operation Audit: Records all user operation and access records, including operator, operation time, operation content, operation result, etc., facilitating security traceability and fault troubleshooting. Audit log retention time can be customized.

3.2.2 Data Security Assurance

The product provides comprehensive data security assurance around the entire lifecycle of data transmission, storage, and use, preventing data leakage, tampering, and loss. Specific functions include:

- Data Transmission Encryption: Supports TLS/SSL (version 1.2 and above) encryption and national cryptography algorithm encryption. Data transmission between client and server, master and slave nodes, and cluster nodes all use encryption methods, preventing data from being stolen or tampered with during transmission.
- Data Storage Encryption: Supports SM4 algorithm for encrypted storage of persistent data, backup data also uses encrypted storage, preventing data leakage.
- Data Masking: Supports masking processing for sensitive data.

3.2.3 National Cryptography Algorithm Support

The product has built-in national cryptography algorithms designated by the State Cryptography Administration, achieving encryption protection for data transmission, storage, identity authentication and other links, complying with national cryptography management standards, supporting Level 3 protection, meeting security compliance needs of government, finance and other key sectors, ensuring data security and business compliance.

Fully supports three core national cryptography algorithms SM2, SM3, SM4, seamlessly integrated into various core modules of the product, including data transmission encryption between client and server, master-slave replication link encryption, data persistence encryption, identity authentication encryption, etc., achieving full-process data security protection.

3.3 Management and Operations

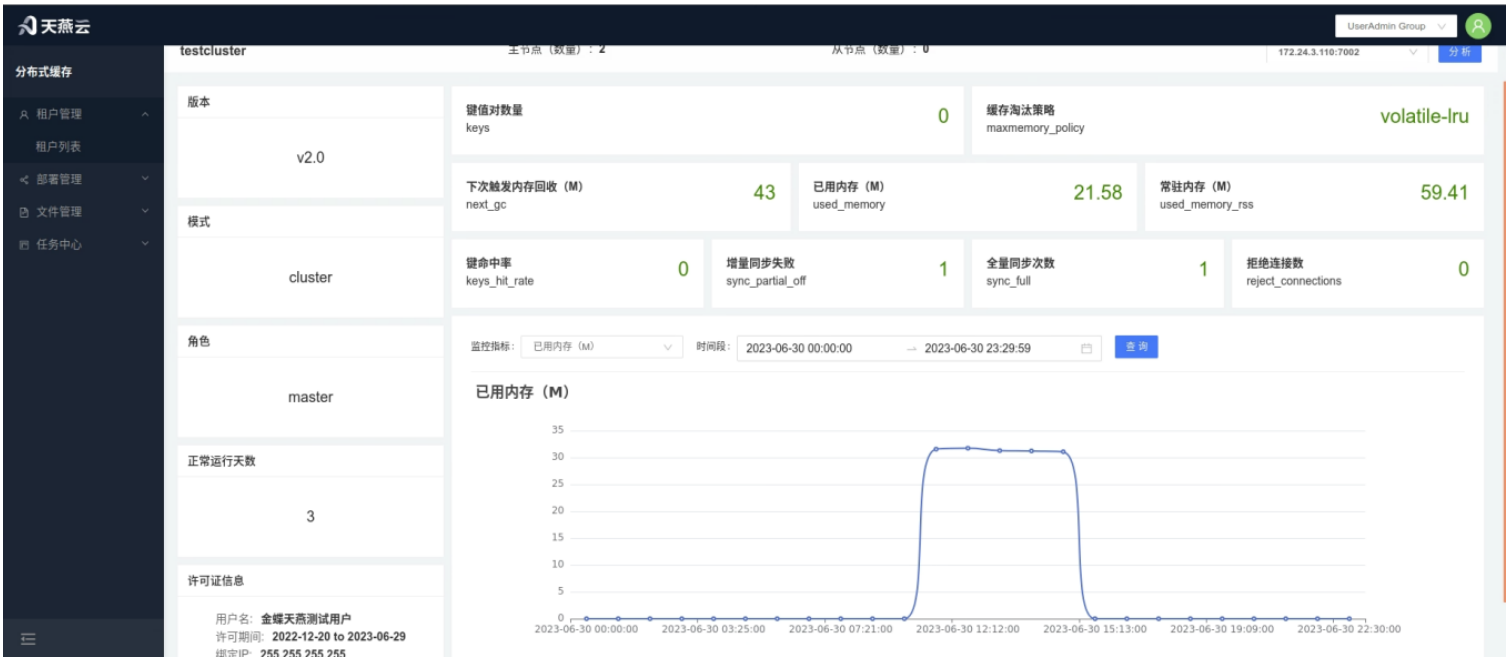
Kingdee Apusic Distributed Cache provides web visual management, automated operations, convenient migration and other functions, simplifying operation processes, reducing operation complexity and costs, adapting to enterprise-level scaled and standardized operation needs.

3.3.1 Visual Web Console

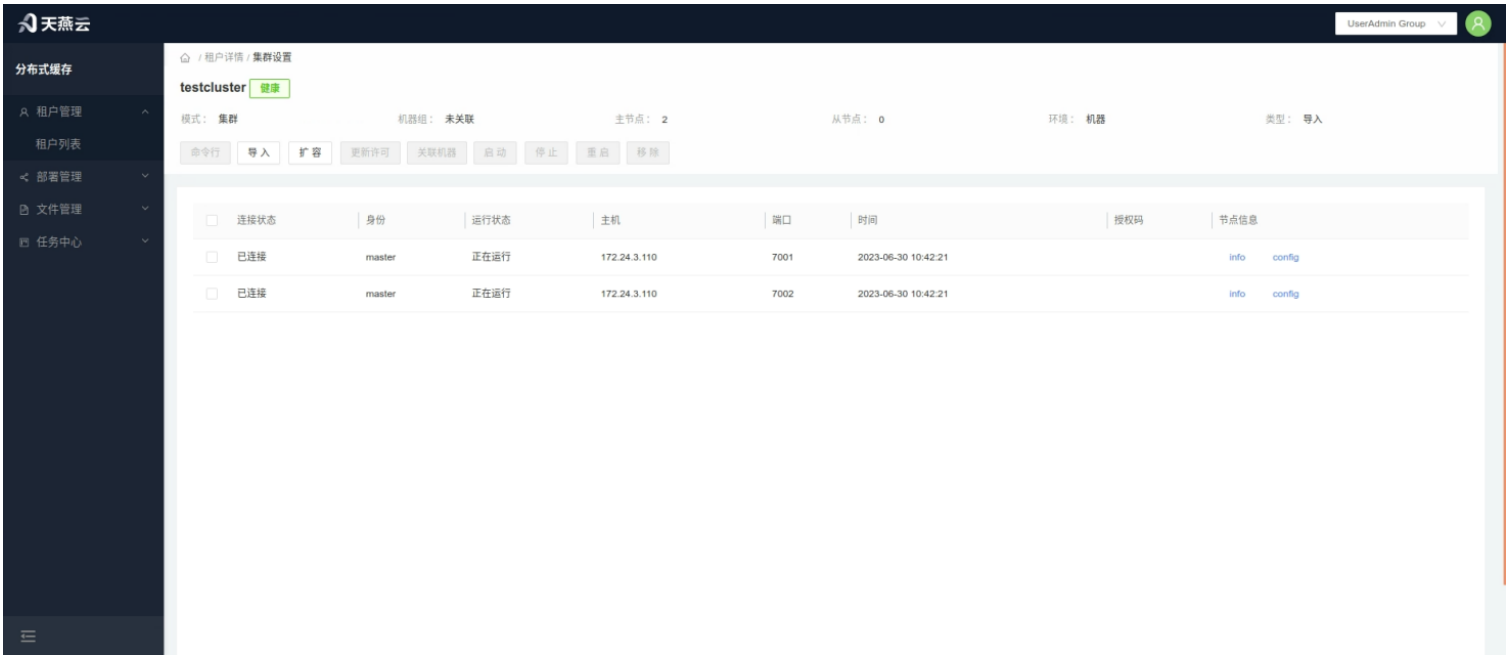
AMDC provides a self-developed web visual management platform, replacing traditional command-line operation methods, achieving full-process visual operations for product deployment, monitoring, configuration, and operations. The interface is simple and intuitive, operations are convenient. Operation personnel can complete daily operation work without mastering complex command-line instructions, lowering operation barriers and improving operation efficiency, adapting to standardized management needs of enterprise-level operation teams.

Web Console Core Functions:

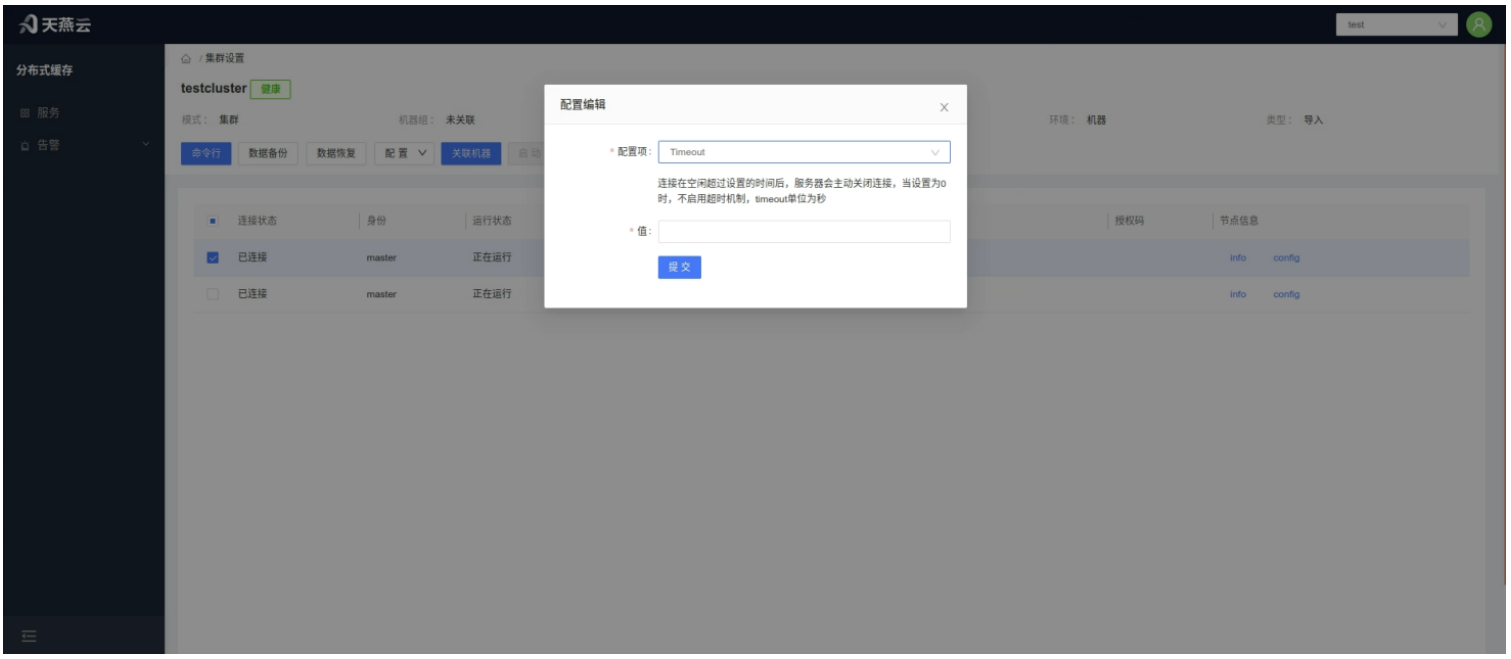
1. Visual Monitoring: Real-time monitoring of cluster node status, CPU usage, memory occupation, QPS, response time, cache hit rate and other core indicators, supporting indicator curve display and abnormal warnings, facilitating operation personnel to grasp product running status in real time;



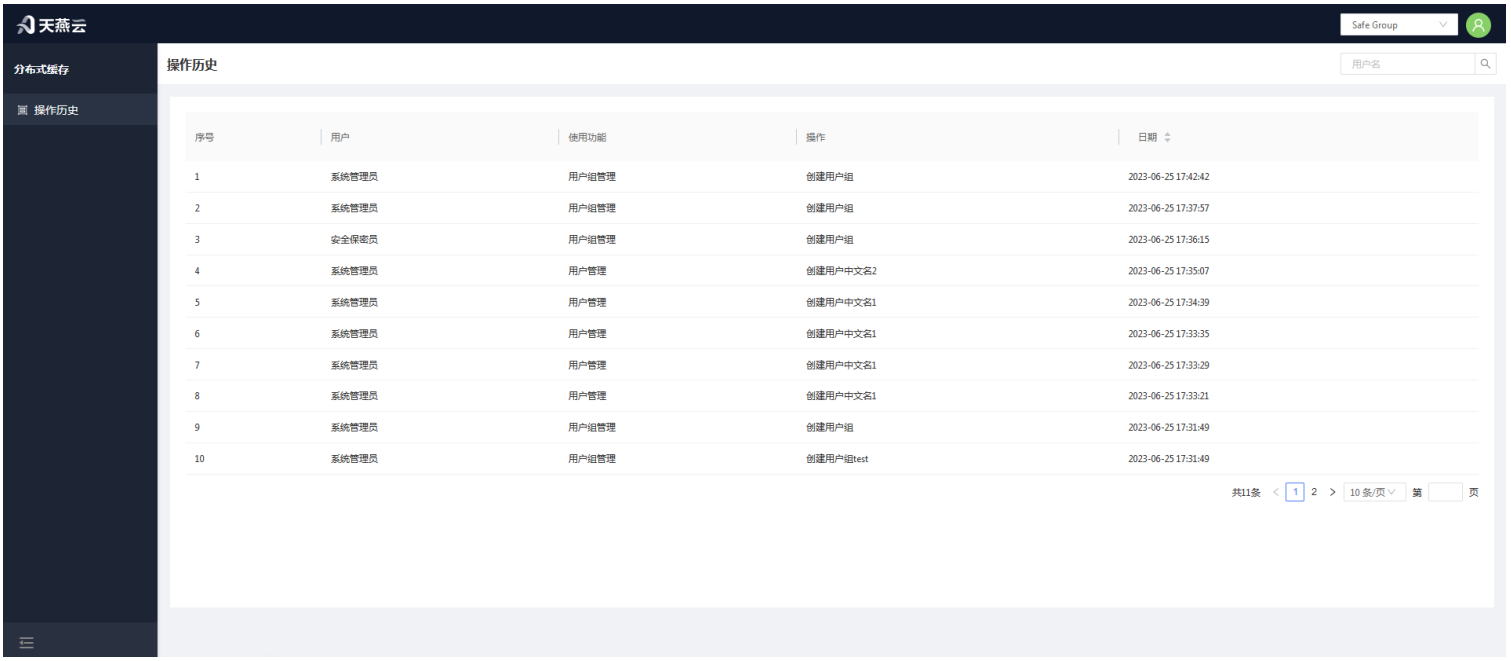
2. Node Management: Supports add, delete, restart, offline and other operations for standalone, master-slave, and cluster nodes, visually displaying node configuration and running status, simple and convenient operation;
    - Cluster Management: Supports cluster scaling, hash slot allocation, data migration and other operations, visually displaying cluster topology and hash slot distribution, simplifying cluster management processes;



3. Configuration Management: Supports visual configuration of product core parameters, cache strategies, encryption configuration, monitoring alert thresholds and other parameters, configuration changes take effect in real time without restarting nodes;

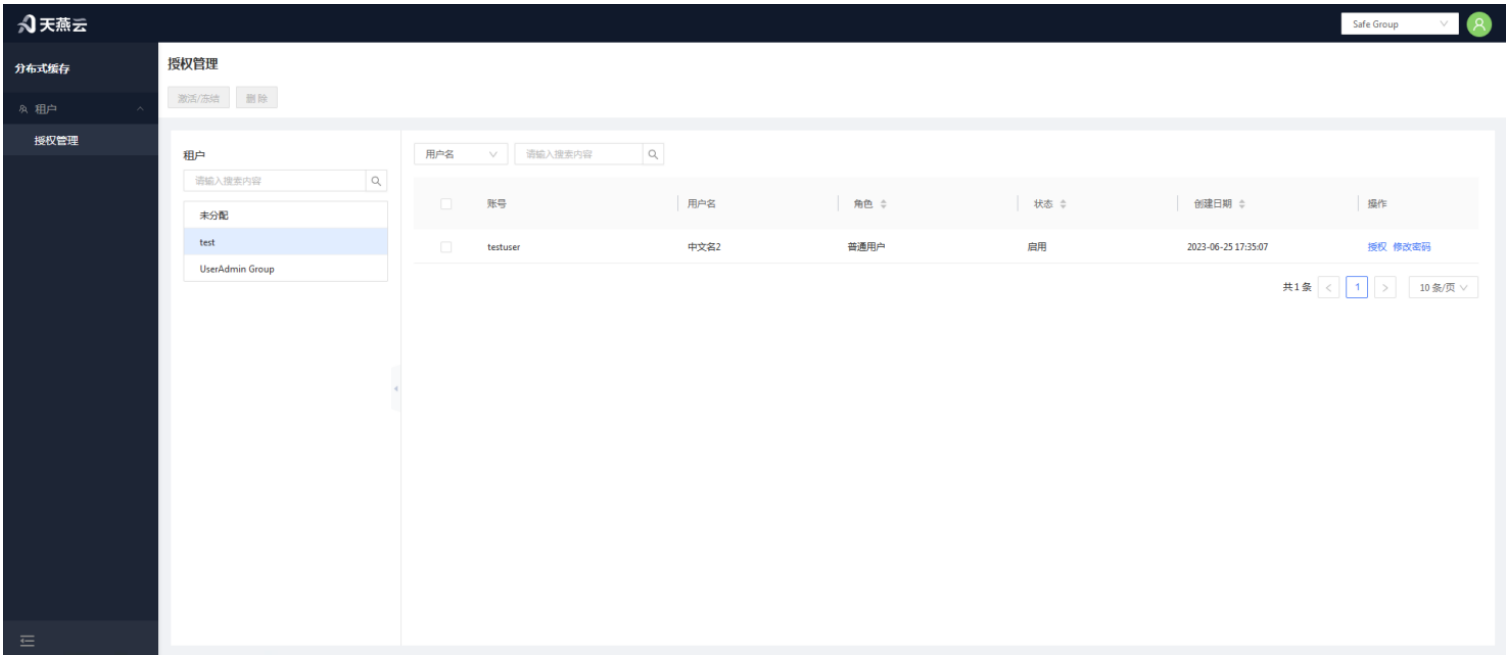


4. Log Management: Supports collection, storage, query and export of operation logs, exception logs, audit logs, sentinel logs and other types of logs. Log retention time can be customized, facilitating operation personnel to troubleshoot faults and trace operations;



5. Permission Management: Supports fine-grained user permission assignment, distinguishing roles such as administrators, operation personnel, ordinary users, limiting operation scope for different roles, ensuring

operation security. Supports user password encrypted storage and password complexity control.



3.3.2 Automated Operations

The product provides complete automated operation capabilities, replacing traditional manual operation, reducing operation personnel workload, improving operation efficiency, lowering operation error rates, adapting to enterprise-level scaled operation needs.

Supports automatic backup and recovery, with configurable backup cycle, mode, and storage path. Backup data is encrypted. Exception alerts can be sent via SMS, email, etc., with customizable thresholds and levels, response time ≤10s. Provides automated deployment scripts and deployment through web console, supporting standalone, master-slave, and cluster mode deployment and node scaling. Deployment time ≤30 minutes. AMDC cluster configuration automatic synchronization, supporting integration of custom operation scripts into the web platform, improving operation efficiency.

3.4 Compatibility

3.4.1 Redis Compatibility

The product is fully compatible with Redis native RESP2/RESP3 protocols and functions, command sets, data structures, and advanced features. Existing business code developed based on Redis can be migrated to AMDC product without modification. Compatible with third-party client tools such as Redis Desktop Manager and Redis Insight. Development and operation personnel can continue using existing development tools and habits. Compatible with mainstream development frameworks such as Spring Data Redis, MyBatis Cache, Laravel Redis, and Redission, supporting framework native Redis integration methods without modifying framework configurations to complete product and framework integration.

The product is compatible with Redis native monitoring indicators and log formats, supporting integration with monitoring tools such as Prometheus and Grafana without restructuring existing monitoring operation systems.

3.4.2 Information Technology Innovation Environment Compatibility

- Domestic Chip Adaptation: For mainstream domestic chips such as Kunpeng 920/930 and Feiteng 2000+/6000 architecture characteristics, optimizes instruction sets and running logic, adjusts memory management and process scheduling mechanisms, improving product running efficiency and stability on domestic chips. Ensures product performance is consistent with open-source environments with performance loss ≤10%.
- Domestic Operating System Adaptation: Fully adapts to mainstream domestic operating systems such as Kylin, UOS, and OpenEuler, optimizes system calls, resource occupation, file operations and other logic, solving compatibility issues under domestic operating systems, ensuring stable product operation.

The product has undergone strict compatibility testing and performance testing in various domestic software and hardware environments, providing complete adaptation test reports to ensure long-term stable operation in domestic environments, meeting application needs of key sectors.

3.5 Client Application Development

AMDC is compatible with the Redis development ecosystem and can directly use Redis client SDKs for mainstream development languages such as Java, Python, Go, C++, C#, and PHP. Developers can quickly complete product and business system development integration without learning new interfaces and development methods.

Provides complete SDK development documentation, sample code, debugging guides, and other supporting materials, detailing SDK installation, configuration, and interface usage methods, helping developers quickly get started. Reduces development and migration costs, supporting smooth transition of enterprise existing development systems.

4 Product Advantages

Compared with traditional open-source Redis products, foreign commercial cache middleware, and other domestic cache products, Kingdee Apusic Distributed Cache Software has significant differentiated competitiveness, forming core advantages of "domestic leadership, excellent performance, security compliance, convenient operation, ecosystem compatibility, and controllable cost". It can fully meet enterprise IT architecture domestic replacement, high-concurrency business support, security compliance, and scaled operation needs, providing enterprises with more cost-effective and reliable cache solutions.

4.1 Autonomous and Controllable

Domestic advantage is the core competitiveness of this product and the key differentiator from open-source Redis and foreign commercial cache products. AMDC product deeply aligns with information technology innovation policy requirements, achieving autonomous and controllable core code and full-stack domestic adaptation while ensuring functional completeness and performance stability. It fully meets domestic replacement needs of government, finance, energy, and other key sectors. Specific advantages are as follows:

1. Core Code Autonomous and Controllable: Removes all foreign dependent components and restricted technologies. Core code has independent intellectual property rights with code autonomy rate exceeding 98%, can provide complete intellectual property certification documents. Breaks free from technology dependence on foreign open-source communities and vendors, fundamentally avoiding security risks such as technology backdoors and data leakage, meeting domestic procurement requirements of key sectors.

2. Full-Stack Domestic Adaptation, Connecting Domestic Links: The product fully adapts to mainstream domestic chips such as Kunpeng, Feiteng, and Loongson, and mainstream domestic operating systems such as Kylin and OpenEuler, adapting to full-stack domestic IT architectures. The product has established deep cooperation with mainstream domestic software and hardware vendors, government informatization vendors, and financial technology vendors, completing full-stack domestic adaptation and compatibility testing, forming a complete domestic ecosystem collaboration system. Adapts to domestic IT architecture specifications in government, finance, and other key sectors, supports integration with domestic operation platforms, security platforms, and identity authentication platforms, building full-stack domestic operation and security systems. Meanwhile provides customized development services to adapt to personalized domestic needs of different industries.

3. Domestic Service Support, Timely and Efficient Response: Relying on domestic professional technical teams, provides full-process domestic service support, including domestic deployment, domestic adaptation debugging, customized development, technical training, troubleshooting and other services. Supports 7×24 hours Chinese technical support with response time ≤1 hour, troubleshooting resolution

time ≤4 hours. Compared with service delays and language barriers of foreign commercial cache products, it better meets service needs of domestic enterprises, ensuring stable product operation.

4.2 High Performance

The product optimizes core algorithms, memory management, I/O operations, and other aspects for high-concurrency and large-scale deployment scenarios, achieving significant performance improvement. Compared with Redis 6.x and 7.x, performance improvement exceeds 2 times in typical scenarios. Cluster performance can scale linearly, fully adapting to enterprise high-concurrency business needs.

Specific descriptions are as follows:

1. Enhanced Concurrent Processing Capability: Optimizes core service layer execution logic, reduces lock contention, improves concurrent request processing efficiency. Single machine QPS ≥100,000, cluster QPS can scale linearly with node count. Supports concurrent connections ≥100,000, response time fluctuation ≤0.5ms in high-concurrency scenarios. Can easily support cache needs in high-concurrency scenarios such as e-commerce promotions, government service peaks, and financial transactions. Compared with open-source Redis 6.2, throughput increases by more than 20% in high-concurrency scenarios.
2. Low Response Latency, High Stability: Optimizes memory management mechanism, adopting memory defragmentation, memory pre-loading and other technologies to improve memory utilization, reduce memory leaks, and lower data access latency. Optimizes I/O operations, reduces disk I/O overhead, especially in persistence scenarios. Through I/O multiplexing, asynchronous I/O, and other technologies, avoids impact of persistence process on product performance, ensuring performance stability in high-concurrency scenarios.
3. More Reasonable System Resource Occupation: Compared with Redis community open-source version, product CPU occupation rate reduces by more than 15% in multi-threaded high-concurrency scenarios, adapting to enterprise server resource optimization needs, reducing hardware investment costs.
4. Excellent Cluster Scaling Performance: Supports dynamic scaling of cluster nodes. Cluster node count can scale to 100+. Scaling process does not affect normal business operation. Sharded data automatically migrates, achieving load balancing. Optimizes cluster coordination mechanism, reduces communication overhead between cluster nodes, improves overall cluster performance. During scaling process, performance can grow linearly with node count without obvious performance bottlenecks, adapting to enterprise cache needs as business scales continue to expand.

4.3 High Security

The product builds a full-process, multi-level security protection system. On the basis of national cryptography algorithm integration, it strengthens identity authentication, permission control, data security, vulnerability

protection, and other functions. It has passed multiple security compliance certifications, fully meeting enterprise-level security needs, especially adapting to government, finance, and other key sectors with extremely high security compliance requirements.

4.4 Improved Operation Efficiency

The product provides web visual management, automated operations, convenient migration and other functions, significantly simplifying operation processes, reducing operation complexity and costs, adapting to enterprise-level scaled and standardized operation needs.

1. Web Visual Management, Lowering Operation Barriers: Provides self-developed web visual management platform, replacing traditional command-line operation methods. Achieves full-process visual operations for node management, cluster management, configuration management, monitoring alerts, log management, etc. Interface is simple and intuitive, operations are convenient. Operation personnel can complete daily operation work without mastering complex command-line instructions, lowering operation barriers and improving operation efficiency.

2. Automated Operations, Reducing Manual Input: Provides complete automated operation capabilities, including automatic backup and recovery, exception alerts, automated deployment, configuration automatic synchronization, and other functions, significantly reducing manual operation workload and lowering operation error rates.

3. Convenient Data Migration, Ensuring Smooth Business Transition: Built-in Redis migration tool supports Redis 5.0 and above version migration, compatible with open-source and other commercial Redis products. Adopts full migration plus incremental synchronization scheme. Migration process does not affect business. Data consistency reaches 99.999% or above. Provides visual monitoring, resumable upload, and migration reports, reducing migration costs and interruption risks, ensuring smooth business transition.

4. Complete Monitoring and Fault Troubleshooting Capabilities: Supports real-time monitoring of cluster node status, performance indicators, cache hit rate, hotspot data, log information, etc. Visual display of monitoring data, supports monitoring indicator curve display and abnormal warnings. Provides comprehensive log management functions, collecting various operation logs, exception logs, and audit logs. Supports log query, export, and analysis, facilitating operation personnel to quickly troubleshoot faults. Supports automatic detection and automatic recovery of node failures with failover time ≤30s, reducing fault handling time and ensuring system stability.

5. Operation Ecosystem Integration, Adapting to Enterprise Existing Systems: Provides standardized operation interfaces, supporting integration with existing enterprise operation platforms such as

Prometheus and security platforms, achieving standardization and automation of operation processes without restructuring enterprise existing monitoring and operation systems.

5 Typical Application Scenarios

Kingdee Apusic Distributed Cache Software can be widely applied in government, finance, energy, internet, enterprise-level applications, and many other industries. It adapts to various core scenarios such as high concurrency, high security, domestic replacement, and scaled operations, providing efficient, secure, and stable cache support for enterprise business systems. It helps enterprises improve business response speed, alleviate database pressure, ensure system stability, and reduce IT costs, promoting enterprise digital transformation and domestic upgrade.

5.1 Technical Usage Scenarios

The product focuses on supporting core technology scenarios for enterprises, specifically supporting typical technical scenarios such as hotspot data caching, application server session centralized storage, and distributed locking, adapting to diverse business needs of enterprises and improving business system performance and stability.

5.1.1 High-Frequency Hotspot Data Caching

Addressing the problem of frequent hotspot data access and high database pressure in enterprise business systems, AMDC product optimizes hotspot data caching function. Adopts memory pre-loading mechanism, loading hotspot data into memory in advance, reducing data access latency, achieving efficient caching and dynamic update of hotspot data. Significantly improves hotspot data access speed, alleviates database load, adapting to high-concurrency hotspot scenarios such as e-commerce promotions, government service peaks, and financial transaction queries.

AMDC supports rich hotspot data eviction strategies, avoiding invalid hotspot data occupying memory. Meanwhile supports automatic cleanup and manual cleanup of expired data, optimizing memory utilization.

5.1.2 Application Server Session Centralized Storage

Addressing the session sharing problem in enterprise application server distributed deployment scenarios, AMDC supports providing session centralized storage solution for application servers. Achieves centralized management and unified storage of sessions, ensuring user session continuity, supporting horizontal scaling of application servers, and improving system availability.

5.1.3 Distributed Lock

Addressing problems such as data inconsistency and resource competition conflicts caused by multi-node concurrent operations in enterprise distributed systems, AMDC product provides high-performance and highly reliable distributed lock function. Achieves unified resource control in distributed environments, ensuring atomicity

and data consistency of concurrent operations, adapting to scenarios such as distributed task scheduling, distributed transactions, and multi-node data synchronization.

AMDC supports native instructions (SETNX, EXPIRE, etc.) to implement distributed locks, providing basic functions such as lock acquisition, release, reentry, and timeout automatic release, ensuring lock atomicity and reliability.

5.1.4 Distributed Counter and Rate Limiting

AMDC has built-in distributed counter and distributed rate limiting capabilities. Relying on deep optimization of underlying atomic operations, it can provide high-precision, low-latency, and strongly consistent counting and traffic control capabilities in distributed deployment and multi-instance concurrent scenarios. It is widely applicable to enterprise-level scenarios such as interface rate limiting, API call statistics, hotspot resource access counting, task concurrency control, traffic peak shaving, inventory and quota overselling prevention, effectively ensuring stability and reliability of distributed systems under high concurrency and large traffic.

5.1.5 Lightweight Message Queue

For enterprise lightweight message passing scenarios, based on Stream data structure, optimizes lightweight message queue function. Achieves full-process management of message publishing, subscription, consumption, confirmation, and backtracking. No need to deploy independent message queue middleware additionally, reducing system complexity and deployment costs. Adapts to lightweight message scenarios such as log collection, message notification, data synchronization, and asynchronous task scheduling.

5.2 Business Application Scenarios

5.2.1 Government Application Scenarios

Adapts to full-stack domestic IT architecture in government sector. Applied in government services, data sharing, office and security protection scenarios. Caches hotspot government data, implements session sharing. Ensures data security through national cryptography algorithms. Supports standardized operations, meeting autonomous control and Level 3 protection and above compliance requirements. Improves government system response efficiency and stability.

5.2.2 Financial Industry Application Scenarios

Meets high security and high concurrency needs in financial sector. Applied in transaction, query, risk control systems, and domestic replacement scenarios. Caches sensitive data such as transactions, market data, and risk control. Meets high performance and low latency needs of financial business systems.

5.2.3 Energy Industry Application Scenarios

Adapts to scaled and distributed IT systems in energy sector. Applied in monitoring, energy management, and equipment management scenarios. Caches high-frequency data such as equipment operation and energy consumption. Supports high-concurrency read/write and sharded storage. Ensures uninterrupted system operation through high-availability clusters. Assists cache layer domestic replacement and operation optimization in energy sector.

5.2.4 Internet Application Scenarios

Fits internet high-concurrency and low-cost needs. Applied in e-commerce, social media, short video, and other scenarios. Caches hotspot data such as products, users, and bullet comments. Prevents cache breakdown and avalanche. Supports session sharing and cluster dynamic scaling. Compatible with Redis ecosystem, reducing deployment and operation costs. Ensures system stability in high-traffic scenarios.

5.2.5 General Application Scenarios

Adapts to domestic upgrade needs of large and medium-sized enterprises. Applied in ERP, CRM, supply chain management, and other systems. Caches basic and high-frequency business data. Ensures data consistency. Supports distributed locks and sharded storage. Simplifies operation processes. Reduces development and migration costs. Helps enterprises improve business processing efficiency and complete cache layer domestic replacement.

6 Conclusion

Kingdee Apusic Distributed Cache Software focuses on the core positioning of "domestic replacement, commercial enhancement, full-stack adaptation, and ecosystem compatibility". It has built a complete functional system and technical architecture, possessing six core advantages: domestic leadership, excellent performance, security compliance, convenient operation, ecosystem compatibility, and controllable cost. It fully meets enterprise IT architecture domestic replacement, high-concurrency business support, security compliance, and scaled operation needs.

Through autonomous and controllable core code, full-stack domestic adaptation, and deep integration of national cryptography algorithms, the product solves pain points of traditional open-source Redis products and foreign commercial cache middleware in autonomous control, security compliance, and domestic adaptation. It can comprehensively replace open-source Redis and foreign commercial cache middleware, especially adapting to domestic procurement needs of government, finance, energy, and other key sectors. Through performance optimization, commercial operation feature enhancement, and ecosystem compatibility optimization, it significantly improves product practicality and cost-effectiveness, reducing enterprise procurement costs, migration costs, and operation costs. It helps enterprises improve business efficiency, ensure system stability, and promote digital transformation and domestic upgrade.

After years of large-scale application practice and refinement in industry customer markets, as well as testing and certification by authoritative third-party evaluation institutions, Kingdee Apusic Distributed Cache Software V2.0 product's core technical indicators meet standards, with complete functions, stable performance, and security compliance. It can be widely applied in diverse scenarios across government, finance, energy, internet, enterprise-level applications, and many other industries. It provides efficient, secure, and stable cache solutions for enterprises and is the preferred product for cache middleware in the process of enterprise IT architecture domestic replacement.

全国统一服务热线

4008-555-800

金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年，前身为“金蝶中间件公司”，是金蝶集团旗下新一代软件基础云平台服务商，云计算国家标准制定企业，国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业，也是“两网一站四库十二金”国家重点工程的基础平台提供商，产品广泛应用于政府、军工、金融、能源等关键行业，累计服务客户总数超过10万家。

Apusic 金蝶天燕

云计算国家标准制定企业

金蝶集团旗下基础软件企业

信息技术应用创新核心企业

官网：www.apusic.com