



APUSIC
固若长城
睿比世界

FAQ)

Kingdee Apusic In-Memory Data Cache V2.0.1

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

版权声明

本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。

本档如有更新，不另行通知。对本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

 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 Troubleshooting Common Issues
 - 1.1 Issue 1: Port Occupation and Resolution
 - 1.2 Issue 2: Modifying the AMDC Cache Service Port
 - 1.3 Issue 3: Modifying the AMDC Console Port
 - 1.4 Issue 4: Loading the ACL File
 - 1.5 Issue 5: Resolving AMDC Master-Slave Failover
 - 1.6 Issue 6: Request Latency Issues
 - 1.7 Issue 7: Monitoring Executed Commands
 - 1.8 Issue 8: Process Running But Unable to Connect
 - 1.9 Issue 9: System CPU Continuously Occupied at High Levels

1 Troubleshooting Common Issues

1.1 Issue 1: Port Occupation and Resolution

Installing AMDC occupies certain ports, and the client connection relies on internal DNS services. If you cannot guarantee that the ports are unused, or if the installation environment lacks effective DNS, startup may fail. Default ports occupied by the product:

- AMDC Cache Service default port:6359
- AMDC Sentinel Service default port: 26359
- AMDC Console default port: 9001

If the aforementioned ports are occupied before startup, it will result in a failure. You can switch the ports or stop the programs currently using them.

1.2 Issue 2: Modifying the AMDC Cache Service Port

Edit the configuration file: `vim conf.yaml`, `port` field

1.3 Issue 3: Modifying the AMDC Console Port

Edit the configuration file: `vim amdc-console/config.yaml`, modify the `HTTP_PORT` field.

For more configuration files, refer to Chapter III of Section I.

1.4 Issue 4: Loading the ACL File

Place the ACL file in the directory specified by the AMDC cache configuration item "ACLFile", and execute ACL load in the AMDC cache command line to reload the user configurations in the ACL file.

Note:

1. Reloading the ACL file will replace existing ACL rules in the cache.
2. If there is any invalid data in one or more entries in the ACL file, the entire file will fail to load, and the existing rules in the cache system will continue to be used.

1.5 Issue 5: Resolving AMDC Master-Slave Failover

In the master-slave mode of AMDC, when a slave process is started, it will send a SYNC Command to the master to request a synchronous connection. Whether it is the first connection or a reconnection, the master will start a background process to save the data snapshot to the data file. At the same time, the master will record all

commands that modify the data and cache them in the data file. After the background process completes the caching operation, the master will send the data file to the slave. The slave side saves the data file on the hard disk, then loads it into the memory, and then the master will send all the operations that modify the data to the slave side.

1. In the cache service cluster, if a slave fails, the system will reconnect automatically. Should a Slave fail and go offline, it will automatically reconnect upon recovery. Upon receiving the Slave's connection, the Master sends its complete data file to the Slave. If the Master receives synchronization requests from multiple Slaves simultaneously, it will only start one background process to save the data file, then send it to all Slaves to ensure the Slave's normal operation.
2. In the cache service cluster, if the master fails, manually restarting the main service enables read-write operations by the client. When the master cache service fails, it still supports read operations by the client on the slave service. You need to manually restart the main service. Once the main service is restarted and the system has recovered, it will again support read-write operations by the client on the main service.

1.6 Issue 6: Request Latency Issues

If you encounter noticeable latency in AMDC requests, you can identify the specific problem with the following methods: 1. `slowlog` command: This command can query commands with longer execution times (the length of time can be configured via the `slowlog` parameter in the `conf.yaml` file), allowing better tracking of the code execution process. 2. `-bigkeys`: Use the `amdc-cli` client, connect and add the `-bigkeys` parameter to query the largest values, thus determining whether the delay is due to excessively large values. 3. `-memkeys`: Similarly, this parameter queries the total memory usage of key-value pairs to determine whether the delay is due to overly large key-value pairs.

1.7 Issue 7: Monitoring Executed Commands

Use `amdc-cli` to connect, then use the `monitor` command to monitor the commands continuously received by AMDC. Alternatively, use the following command to record them in the background to the `monitor.log` file:

```
amdc-cli -h [ip] -p [port] [-a password] monitor >monitor.log
2>&1 &
```

To stop recording to `monitor.log`, use the following command:

```
ps -ef | grep amdc-cli | grep -v grep | awk '{print $2}' | xargs
kill -9
```

1.8 Issue 8: Process Running But Unable to Connect

When the client exceeds Linux limits or uses more memory than the license limit, it may appear that the process is running but connection timeouts or connection failures occur.

1. Client exceeding Linux limits: Some Linux systems have a default limit of 1024 for client connections. When there are too many business applications producing over 1024 client connections, the excess will be restricted by Linux and unable to connect further.

Solution: Permanently set the user-level file descriptor limit by adding the following lines to `/etc/security/limits.conf`:

```
* soft nofile 65535
* hard nofile 65535
```

Note: Different Linux systems may require different methods for modification. If the above does not work, please search for the corresponding system's TCP connection count.

2. Exceeding the license memory limit: The default memory limit for licenses is generally 6GB (for most cases). When the business system writes data exceeding 6GB and continues to write, it causes frequent triggering of the memory eviction algorithm in AMDC, leading to a decrease in AMDC's performance. Under continuous high concurrency, some requests may timeout, causing errors in the business segment.

Solution: Expanding capacity resolves the issue. There are two methods: one is to apply for a license with greater memory, directly increasing the usable memory per node (not recommended to exceed 16GB, as larger sizes can lead to reduced query efficiency); the second is to implement a distributed cluster mode (corresponding to Redis' cluster mode).

1.9 Issue 9: System CPU Continuously Occupied at High Levels

Similar to issue 8, part 2, frequent triggering of the eviction algorithm significantly increases CPU utilization and maintains high levels over long periods. The solution is the same as for issue 8, part 2.

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

