

Cache 方案比較

Samuel

Why cache?

- 時常取得db的某些資料
- 把這些資料放在**RAM**以加快取得速度
- 空間換取時間

分類

- **Local Cache**
- **Remote Cache**

Local Cache (static variables in JVM)

```
@Slf4j
public class CachePool {

    public static final String ACTION_CONFIG = "ACTION_CONFIG";

    private static HashMap<String, Object> CACHE_MAP = new HashMap<>();

    public static void putCache(String sKey, Object oObj){
        CACHE_MAP.put(sKey, oObj);
    }

    public static Object getCache(String cacheName){
        return CACHE_MAP.get(cacheName);
    }

    public static void cleanCache(String cacheName){
        CACHE_MAP.remove(cacheName);
    }

    public static void cleanAllCache(){
        CACHE_MAP.clear();
    }
}
```

Clear cache by execute .java (舊)

```
public class ClearCacheJob implements StatefulJob, InterruptableJob{
    static Logger logger = Logger.getLogger(ClearCacheJob.class);

    private final String jobName = this.getClass().getName();
    private volatile boolean bInterrupt = false;
    private volatile Thread thread;
    private QjobService qjobService;
    private CachePool cp;

    public QjobService getQjobService() {
        if ( qjobService ==null){
            qjobService = new QjobServiceImpl();
            qjobService.setQjobDao(new QjobDaoImpl());
        }
        return qjobService; }

    public void setQjobService(QjobService qjobService) { this.qjobService = qjobService; }

    @Override
    public void execute(JobExecutionContext context) throws JobExecutionException{
        long lStart = LogUtil.logFuncStart(logger);
        thread = Thread.currentThread();
        Properties pParm = JobUtil.transferQJobDataMap(context.getJobDetail().getJobDataMap());
        LogUtil.LogDebug(logger, sMsg: "runJob " + jobName + "(" + thread.getName() + ")" + " Start...");
        DataProxy dataProxy = new DataProxy();
        Connection connEC = null;
        try{
            connEC = dataProxy.getConnection();
            pParm.setProperty("JOB_NAME", pParm.getProperty( key: "JOB_OID",jobName));
            String sJobStartDate = DateUtil.getSysDatetime();
            String sJobStatId = JobUtil.logJobStart(dataProxy, connEC, sJobType: "Core", pParm.getProperty("JOB_NAME"),sJobStartDate);
            this.getQjobService();
            //Clean Lookup Cache
            ConfUtil.resetLookupGroup();

            //Clean Cache
            CachePool.cleanAllCache();
        }
```

Clear cache by API (新)

```
@GetMapping(path = "/clearJVMCache")
public ResponseEntity<String> getBlogDetailNumber(@RequestParam(required = false, value = "cacheName") String cacheName,
HttpServletRequest request){

    if (null == cacheName) {
        CachePool.cleanAllCache();
        return ResponseEntity.ok()
            .body("Clean all cache success");
    } else {
        CachePool.cleanCache(cacheName);
        return ResponseEntity.ok()
            .body("Clean cache: " + cacheName + " success");
    }
}
```

Guava Cache 套件

- 1.何時過期
- 2.如何淘汰
- 3.重新整理內容

Remote Cache

```
redis:6379> SET tomcat-cache-get-randomBanner '{"SQL_STMT":  
  "SELECT ..."}'
```

```
get tomcat-cache-get-randomBanner
```

```
} finally {  
    if (null != dataProxy) {  
        dataProxy.release();  
    }  
    redisCache.release();  
}
```




Memcached

僅能儲存key-value類型

multi thread i/o

redis 6.0

Local vs Remote

Local	Remote
無法共享	多個server共享
本地較快	遠端較慢
擴充包括後端伺服器&cache	擴充僅限cache



Thank you

產業首選 · 通路標竿