云 :
: 云 .note
: http://note.youdao.com/noteshare?id=c10ecc5535e673b3f7fa396e57866569&sub=CA2841ACDA7C4A6B9C1D8EE4E5B4379E

云 享

介 下 作 与 , 作中主 作 , 主 ;
(你 信 二 上, 个主 为了)

介 下 , 亮 , 介 下 ; (主
做 事 。 , 做 事)

二 Java

, 为什么 ? ;

, 什么 候会 僵 ;

, 什么 , 何 ;

个 ? 何 ?

volatile ThreadLocal 使 ;

ThreadLocal什么 候会 OOM ? 为什么?

synchronized volatile synchronized 与 ;

三 JVM

JVM , GC ;

GC 两 , Minor GC Full GC 什么 ? 什么 候会 Full GC? 什么 ?

JVM classloader,为什么会 ?

什么 亲 ? 介 些 作 , 亲 ;

什么 下 们 亲 ;

JVM 优 些? 以 体 个 , 什么值?

JVM class 件 何 ;

Java

;

NIO 什么? 于何 ?

Java9 Java8 了什么;

HashMap 什么? 么 ? (会 伸
ConcurrentHashMap与HashMap HashTable , 了);

, 不 , 们 中 使 ;

义 ;

List Map , Arraylist 与 LinkedList , ArrayList 与 Vector ;

五 Spring

Spring AOP ?

Spring bean 作 ;

Spring Boot Spring做了 些 ? Spring 5 Spring4做了 些 ;

何 义 个Spring Boot Starter?

Spring IOC 什么? 优 什么?

SpringMVC 代 AOP 事 ;

中 件

Dubbo 介 ;

Dubbo ?

Dubbo Provider 供 上 , 体 么做?

Dubbo 候 ?

了 中 件产 ? 产 优 介 ;

中 件 何保 何 ?

Spring Cloud 介 ;

Spring Cloud 下Dubbo，什么 下 使 Spring Cloud?

七

介 : 他 享 ;

乐 业 ;

事 介 , 事 , 些，什么事两 交 三 交;

MySQL binlog 主 三 ? 优 什么?

MySQL , 乐 享 ;

事 2 交, ;

事 ，MySQL Spring 何 事 JDBC 何 事
事 事 ;

SQL 个 SQL ;

Redis

Redis为什么 么 ？redis 会 些 ?

Redis ;

Redis ;

Redis) Redis 何 ?

Redis 何使 Redis ?

Redis 作 ，Redis 何 ?

九 他

些 代 ? 会 你 些 ? (主
, 停 , 了 , 个 于 ,
你 , 上 会pass , 也 ！)

: https://shimo.im/docs/LUYuXUGSX8wTOzY7/ 云 -

Java多

1 原 ，为什么 创建 ？创建 式；

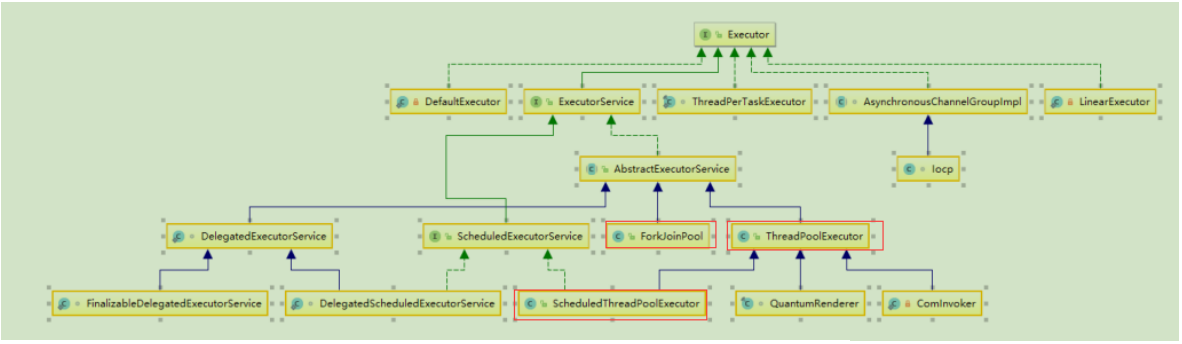
原 ：

JAVA

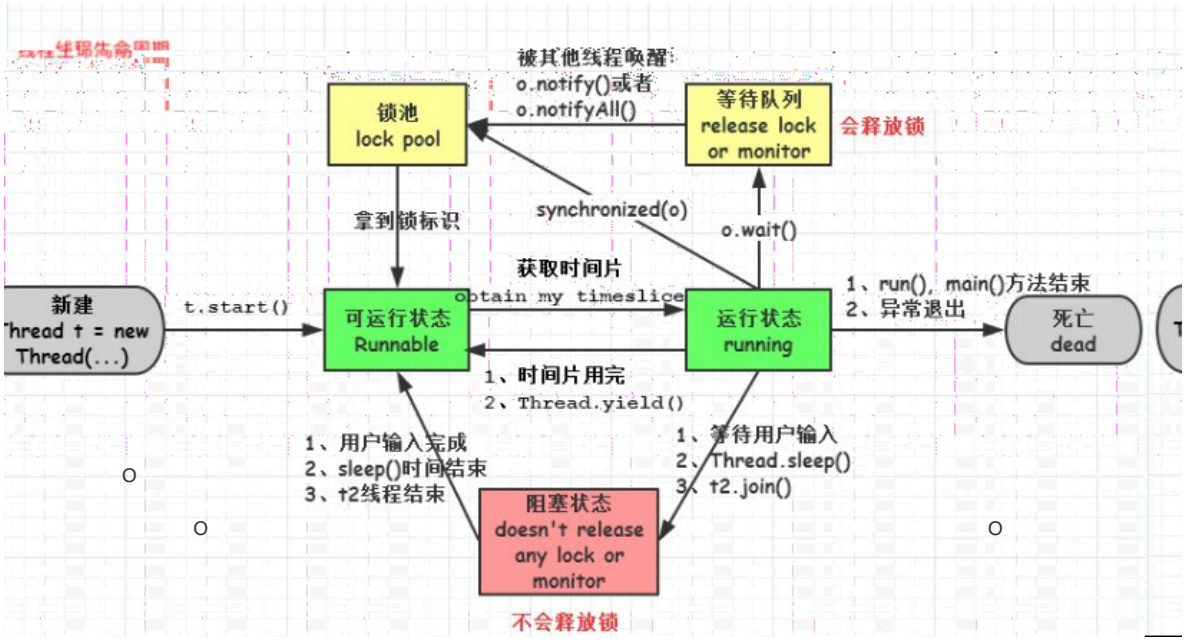
JAVA 二

创建 几 式：

ThreadPoolExecutor ThreadScheduledExecutor ForkJoinPool



2 命周 ，什么 候会出 僵 ；



僵 , SIGCHLD信 ,

停 僵 为 , 个 下 僵

3 安全 ，什么 安全，如何实 安全；

- 中不会产 享 ,

不 - 个 作主 中 , 不

三

1）互

临 ：syncronized ReentrantLock
信 semaphore
互 mutex
2)
CAS（Compare And Swap)
3)
代
使 Threadlocal 享 ，做 个 copy
储
：https://blog.csdn.net/jackieeecheng/article/details/69779824

4 创建 哪几个 心参 ？ 如何合 大小?

1)

```
public ThreadPoolExecutor(int corePoolSize, //
                          int maximumPoolSize, //
                          long keepAliveTime, //
                          TimeUnit unit,
                    //          任          //                    worker
                          BlockingQueue<Runnable> workQueue,
                          ThreadFactory threadFactory,
                          //        任              任
                          RejectedExecutionHandler handler)
```

2)
1 中 于 corePoolSize ，
2 中 于 于 corePoolSize ， workQueue 中,
中 们不 任 ， 中 ， 从 workQueue 中
任
3 workQueue ， 不下 任 ，
maximumPoolSize（ 值)
4 中 于 maximumPoolSize 使 RejectedExecutionHandler 任
：http://gudong.name/2017/05/03/thread-pool-intro.html

3)
你 什么任 了，CPU IO ，任
不 ， 也不
任 为：CPU IO ， 于不 任 不

3.1）CPU

使 , Cpu +1

3.2）IO

： 以使 , CPU * 2

二：（ 与 CPU 之 + 1）* CPU

3.3）

以 任 为CPU IO , 使 不 ,

： https://www.cnblogs.com/cherish010/p/8334952.html

5 volatile ThreadLocal 使 场 和原 ；

volatile

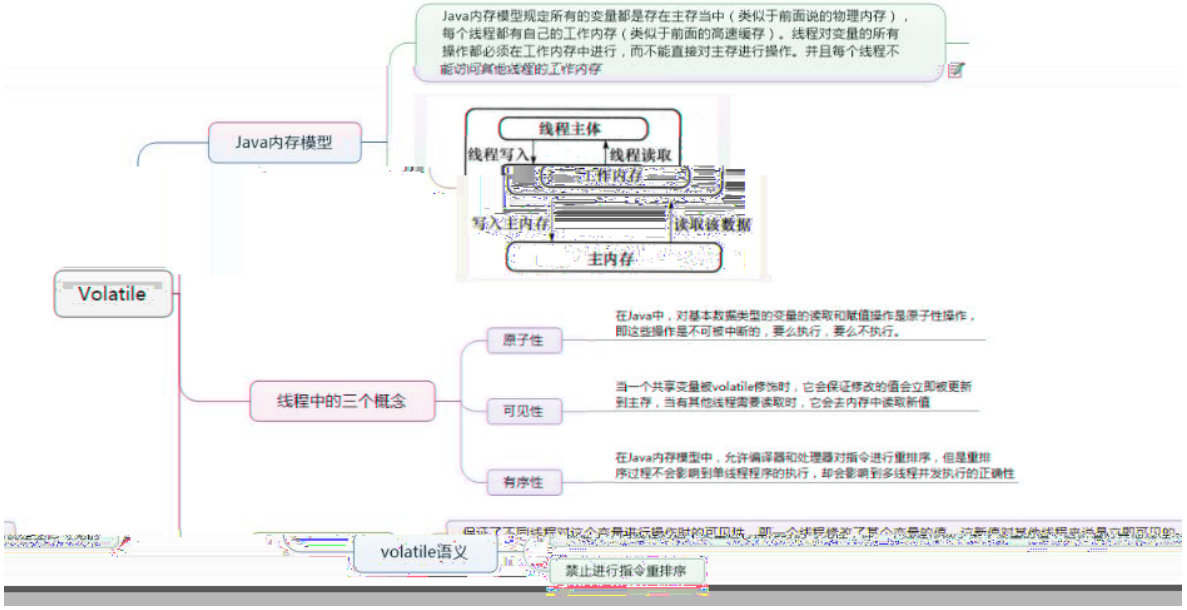
volatile 作 , JVM 会 Lock 令, 个

会

Lock 令 上 于 个 (也), 保 令 不会

令 之 位 , 也不会 令 ;

令 , 作



volatile

1) , : 停

2) , :

3) , : 个值

4) “volatile bean”

5) 低 “ - ” , :

： https://blog.csdn.net/hgc0907/article/details/79664102

： https://www.ibm.com/developerworks/cn/java/j-jtp06197.html

ThreadLocal

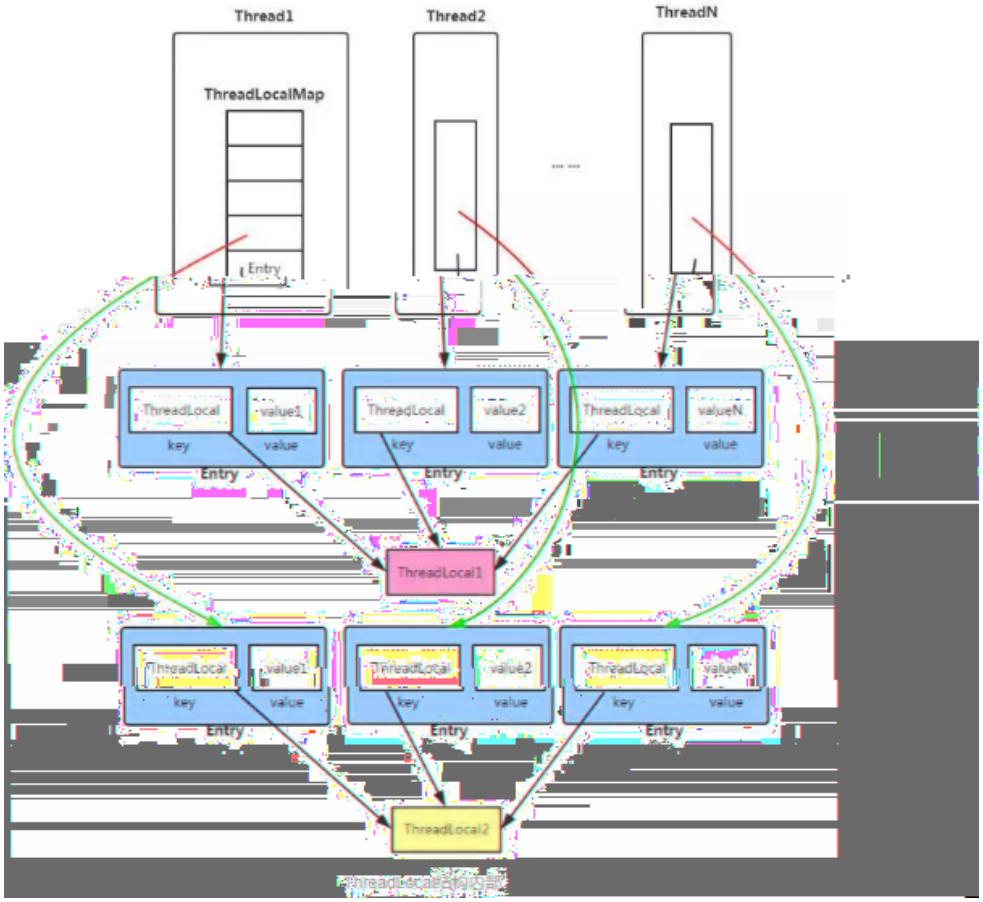
ThreadLocal ， 不 享 ThreadLocal

值 map中，以ThreadLocal 作为key， 之

值 享 ， 也 享 ，

： https://www.jianshu.com/p/ee8c9dccc953

： https://blog.csdn.net/xlgen157387/article/details/78297568



ThreadLocal

： Session

： https://www.jianshu.com/p/cadd53f063b9

6 ThreadLocal什么 候会出 OOM 况? 为什么?

ThreadLocal Thread ， 们 不 ， 会

，Thread 会 些 作， 中 ThreadLocalMap，

Thread **exit** 下：

```
/**
 * This method is called by the system to give a Thread
 * a chance to clean up before it actually exits.
 */
private void exit() {
    if (group != null) {
        group.threadTerminated(this);
        group = null;
    }
    /* Aggressively null out all reference fields: see bug 4006245 */
    target = null;
    /* Speed the release of some of these resources */
    threadLocals = null;
    inheritableThreadLocals = null;
    inheritedAccessControlContext = null;
    blocker = null;
    uncaughtExceptionHandler = null;
}
```

ThreadLocal 使 下， 下不会 ，但 使 了 ， 依 于 ， 不 ， 么 会

： https://blog.csdn.net/xlgen157387/article/details/78297568

7 synchronized volatile区别

1) **volatile**主 个 例 ， 主 享 值从 使 个 以 值， 值 从主 中 ；**synchronized** ， 以 ， 他 住 ，**synchronized** 会 个 ， 令保 了 CPU 作 会 主 中（ ），从 保了 作 ， 也使 个 作

2) **volatile**仅 使 ；**synchronized** 以使 **volatile**不会 ；**synchronized** 会 ， 个 争 **synchronized** ，会

3) **volatile**仅 修 ，不 保 ；**synchronized** 以保 修 ， 为 临 ，从 保 临 中

4) **volatile** 不会 优 ， 以 令 ；**synchronized** 以 优

： https://blog.csdn.net/xiaoming100001/article/details/79781680

8 synchronized 度 场 ；

synchronized：，

：https://www.jianshu.com/p/cf57726e77f2

：

，：https://blog.csdn.net/u013925989/article/details/50208839

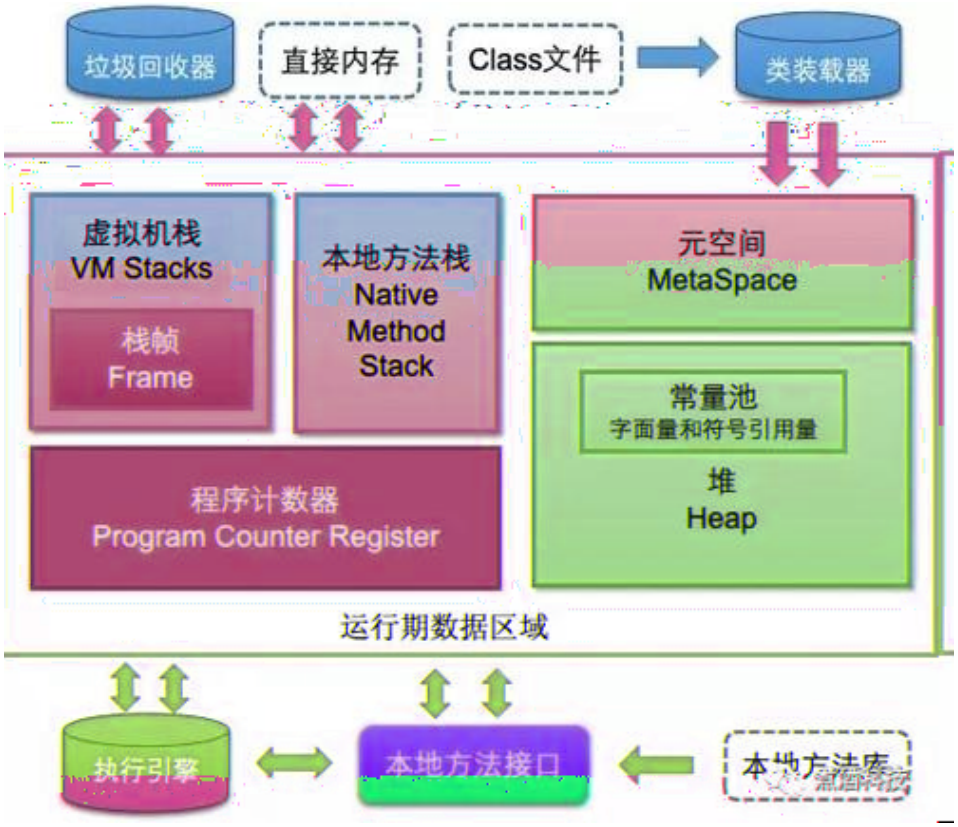
二 JVM 关

1 JVM内存 型，GC 制和原 ；

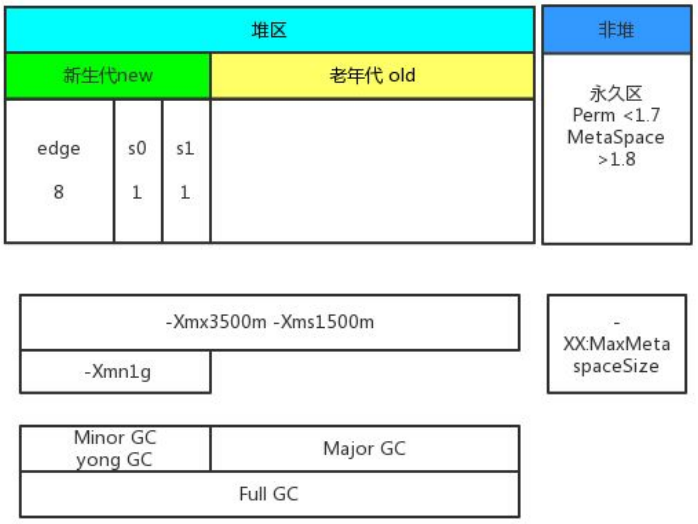
Jdk1.6 之 ： 久代，

Jdk1.7： 久代，但 “ 久代”，

Jdk1.8 之 ： 久代， 元



2 GC分哪两 ，Minor GC 和Full GC 什么区别？什么 候会 发Full GC？分别 什么 ？



从 代 , 们 之为 "minor GC"
从 代 , 们 之为 "major GC"

Minor GC
个YouGen , eden , S0\S1 会 于MinorGC Allocation Failure(YoungGen 不), minorGC

Major GC
OldGen 不 , Major GC

Full GC
Full GC 个 — 代 久代

Full GC
1）System.gc
2）promotion failed (代 , eden S 不下, Old 不下, 么Promotion Failed,会 FullGC)
3）CMS Concurrent-Mode-Failure
于CMS 中主 为 : 1.CMS initial mark 2.CMS Concurrent mark 3.CMS remark 4.CMS Concurrent sweep 2中gc 与 , 么 依 产 , 个 会产 CMS-Mode-Failure, 为SerialOld 做mark-sweep-compact
4） 代 于 代 余 (为了 代 代)
使 G1,CMS , FullGC 候 Serial+SerialOld
使 ParalOld , FullGC 候 ParallNew +ParallOld.

3 JVM5 几 classloader，为什么会 多 ？

: JRE , jre 下 rt.jar,charsets.jar

: JRE ext中JAR

: ClassPath 下

义 : 义 下

为什么会 :

1) ,

2) 为了

4 什么 双亲委 制? 介 些 作 ，双亲委 型 好处;

个 了 ， 不会 ， 个

， ， 上 ，依 ，

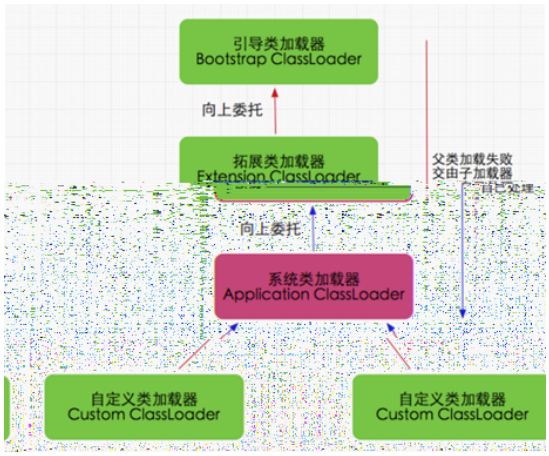
， 以 任 ， ，倘

任 ， 会 ， 亲 ， 个儿 不

， 丢 亲 ， 亲 件事 也 不了 ，儿

， 不 传 中 亲

作



: String.class 不会 ， 便 以 API

: 亲 了 ， ClassLoader

5 什么 况下我们 坏双亲委 型;

<待 充>

6 常 JVM 优 哪些? 可以具体到 哪个参 ， 成什么值?

优

console，jProfile，VisualVM

Dump 信 :

信 :

信 :

信 :

: 个 什么 下

CPU : 些 CPU

: 些 中 ( )

7 JVM 内存划分 加 器 垃圾 垃圾 器 class 件 如何 ;

JVM ( )

( )

: - - 代

: Serial ParNew **Parallel** Scavenge Serial **Old** **Parallel Old** CMS G1 Z

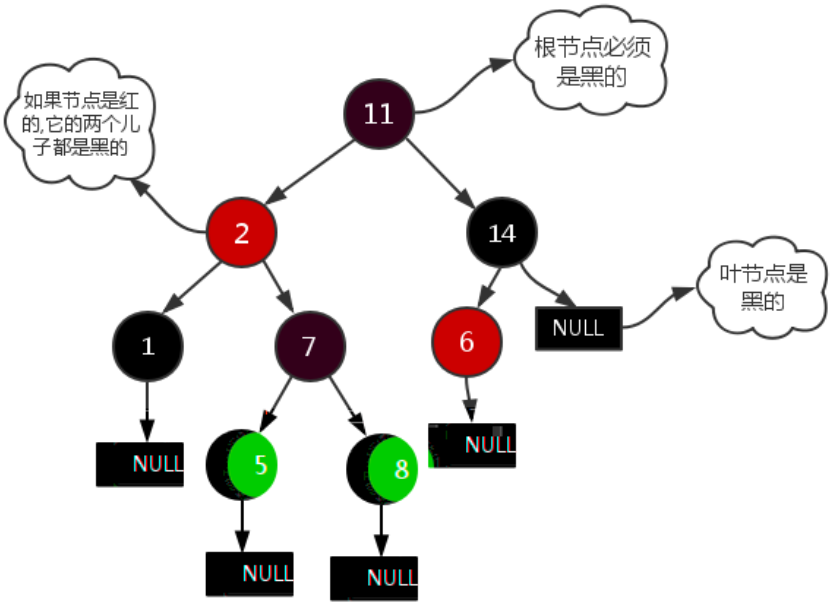
class 件 何

: https://blog.csdn.net/sinat_38259539/article/details/78248454

```
ClassFile {
    u4 magic;                   魔数
    u2 minor_version;           副版本号
    u2 major_version;           主版本号
    u2 constant_pool_count;     常量池计数器
    cp_info constant_pool[constant_pool_count-1];   → 常量池
    u2 access_flags;            访问标志
    u2 this_class;              类索引
    u2 super_class;             父类索引
    u2 interfaces_count;        接口计数器
    u2 interfaces[interfaces_count];   → 接口表
    u2 fields_count;            字段计数器
    field_info fields[fields_count];   → 字段表
    u2 methods_count;           方法计数器
    method_info methods[methods_count];   → 方法表
    u2 attributes_count;        属性计数器
    attribute_info attributes[attributes_count];   → 属性表
}
```

三　Java扩展

1　　实　原　和应　场　；



(　　二　)五　　：

1) 个　　么　　，　么

2)

3) 个　　，

4)　　个　　，　么　　个儿

5)　个　，从　　上

1）　　于C++　STL中,map　set　　.

2）　　linux　　Completely Fair Scheduler,　　,

　　储　　上,个　　个　,

　　储　,　　.

3）IO　　epoll　　sockfd，以　　.

4）ngnix中,　　**timer**,　为　　,　以　　.

5）java中　TreeSet,TreeMap

2　NIO　什么?　　于何　场　?

（New IO）为　　（boolean　　）　供　　，使　　以　供　　伸

　　：**I/O**　　+　　**I/O**

NIO

以使　　令　　-Xlog　　JVM 上　　件　　以
修

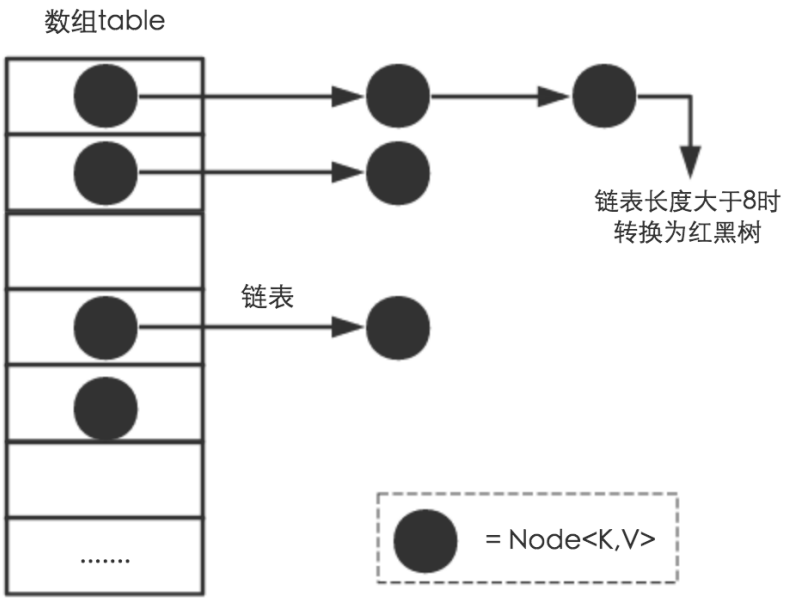
8)
Java 9　　了　Java 8 中　　，　　G1　为
．为　　于Parallel　　，G1会　　上做　　事　，　Parallel　乎
上做任何事　，　　上　　依　GC　　G1　会为
作，从

9）I/O
java.io.**InputStream** 中　　了　　**InputStream** 中
readAllBytes：　　**InputStream** 中　　余
readNBytes：从 **InputStream** 中　　中
transferTo：　　**InputStream** 中　　**OutputStream** 中
　　：
java8
java9

4　HashMap内　　什么？底层　　么实　　？。

HashMap
jdk8以　：　　+
jdk8以　：　　+　　（　　8　，　　为　　）

，　　，　　会产　　，　　get　　候，会　　，
CPU　100%　　，　以　　下使　**HashMap**



5　延伸　察ConcurrentHashMap与HashMap　HashTable　，　察对　　3 入了　　度；

HashMap HashTable ConcurrentHashMap 与
，HashMap
ConcurrentHashMap jdk1.8中
ConcurrentHashMap1.7 1.8 不
ConcurrentHashMap1.8
ConcurrentHashMap1.8
ConcurrentHashMap

6 反射 及实 ，反射 不 很慢，我们在 中 否 免使 反射；

于 些 修 为 中

二

```
Foo foo = new Foo();
: Object getClass
Class cla = foo.getClass();
二 : 例
Class cla = foo.class;
三 : Class.forName
Class cla = Class.forName("xx.xx.Foo");
```

三

1)

了 些 ，以 JVM 些代 优 ，作
些 作低 们 代
中使

2)

使 个 中

3)

于 允 代 些 下不 允 作（ ），以
使 会 之 作 - - 代 上 ，低 代
了 ， 候，代 为 也

7 定义 场 及实 ；

义 ， SpringAOP 以

（API ）

1） 义 个 ， 上

2） SringAOP

3）

4） 值

8 List 和 Map 区别

List 储 ，Map 储 值 ，

List中 储 ， 且允 ，值允 个**null**；

Map中 储 ， 不 ，值 以 ，key 个**null**

二

List

1） 以允

2） 以 个**null**元

3） 个 ，保 了个元 ，

4） ArrayList LinkedList Vector ArrayList 为 ， 供了使 ， LinkedList 于 从 List 中 元 为

Map

1）Map不 collection Map 个

2）Map 个 Entry 两个 ，也 个 个值，Map 会 值 但

3）TreeMap 也 Comparator Comparable 了 个

4）Map 你 以 个 **null** 值但 个 **null**

5）Map 个 HashMap LinkedHashMap Hashtable TreeMap （HashMap TreeMap ）

Set（ ）

1）不允

2） ，你 保 个元 储 ，TreeSet Comparator Comparable 了 个

3） 允 个 **null** 元

4）Set 个 HashSet LinkedHashSet 以 TreeSet 于 HashMap HashSet；TreeSet 了 SortedSet ， TreeSet 个 compare() compareTo() 义

三 （ ）

1) 你 会使 中 元 , 么 List 你 你
了 , 么 List ArrayList 以 供 ,
元 , 么 LinkedList
2) 你 中 元 们 储, 么 List, 为 List
个 , 储
3) 你 保 元 ,也 你不 值 , 么 以 个 Set
, HashSet LinkedHashSet TreeSet Set 了
, 且 供了 TreeSet 个 SortedSet, 储于 TreeSet
中 元 以使 Java Comparator Comparable LinkedHashSet 也
元 们 储
4) 你以 值 储 么 Map 你 你 以 你
从 Hashtable HashMap TreeMap 中

: List Set Map

9 Arraylist 与 LinkedList 区别，ArrayList 与 Vector 区别;

1)
Vector ArrayList 使 , LinkedList 使 , :
LinkedList 位 作，不 ;
ArrayList Vector ，不 位 作
但 ArrayList 元 , LinkedList

2)
Vector ，ArrayList LinkedList 不

3)
ArrayList 元 会 50%, Vector 100%, ArrayList

: ArrayList LinkedList 使

四 Spring 关

1 Spring AOP 实 原 和场 ;

AOP（Aspect Orient Programming），作为 , 于
些

事

二
AOP（ AOP , 不 Spring AOP）主 主
Spring AOP AspectJ

1）AspectJ

AspectJ 代 ， AspectJ ， 个 令 ， 个 代 ， 代 了业 ， ， 于下 ，

2）Spring AOP

Spring AOP 代 ， 业 ， 以不会 ， 于 代 ，Spring AOP 供了 JDK 代 以 CGLib JDK 代 为 代 例， 不 代 信 （ Java ），个 了代 代 （ ），代 ， 代 例 ， 体 invokeHandler

CGLib 代 依 **asm** ， 代 **class** 件 ，修

但 Spring AOP 于 下， 依 于AspectJ

2 Spring bean 作 域和 命周 ；

作

类别	说明
singleton	在Spring IoC容器中仅存在一个Bean实例，Bean以单例方式存在，默认值
prototype	每次从容器中调用Bean时，都返回一个新的实例，即每次调用getBean()时，相当于执行new XxxBean()
request	每次HTTP请求都会创建一个新的Bean，该作用域仅适用于WebApplicationContext环境
session	同一个HTTP Session共享一个Bean，不同Session使用不同Bean，仅适用于WebApplicationContext环境
globalSession	一般用于Portlet应用环境，该作用域仅适用于WebApplicationContext环境

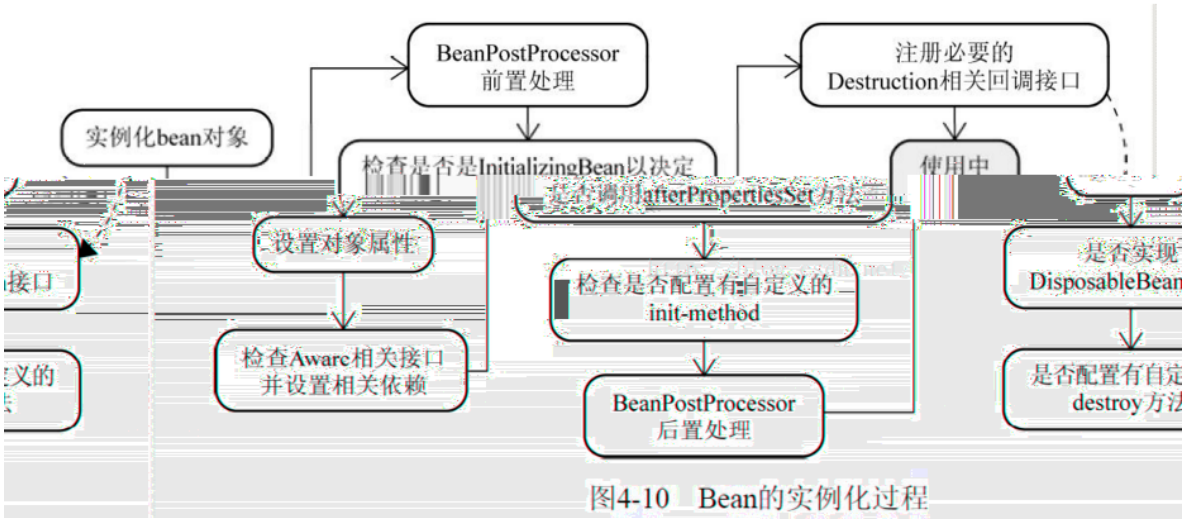


图4-10 Bean的实例化过程

3 Spring Boot Spring做了哪些 ?

1）Spring Boot 以 Spring ；

2） 了 Tomcat，Jetty Undertow ，也 以 ， 不 做 作了；

3） 像Spring xml 件 ；

4） 以 Spring SpringBoot XML 为Java ， bean 为使 (@Autowire)， 个xml properties 个appliaction.yml 件中

5） 供了 些 ， ， 以 些 些 三 ；

6） 依 （ ，例 spring-webmvc jackson-json validation-api tomcat ）， 供 POM 以 Maven 们 依 ，SpringBoot会 他依

Spring 5 Spring4做了哪些 ；

Spring 4.x

1. 依

2.

3. web

4. Bean Validation 1.1（JSR-349） SpringMVC

5. Groovy Bean 义DSL

6. Java 作API

7. JSR310 API

8. 任 MVC 他

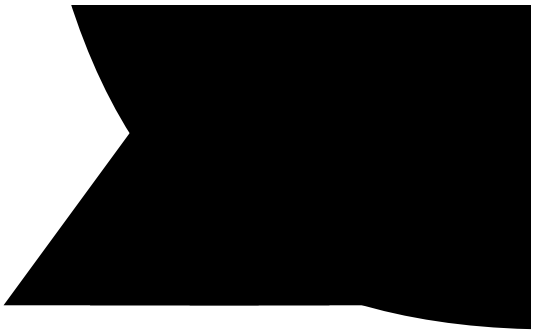
Spring 5.x

1. JDK8

2.

3. SpringWebFlux

4.

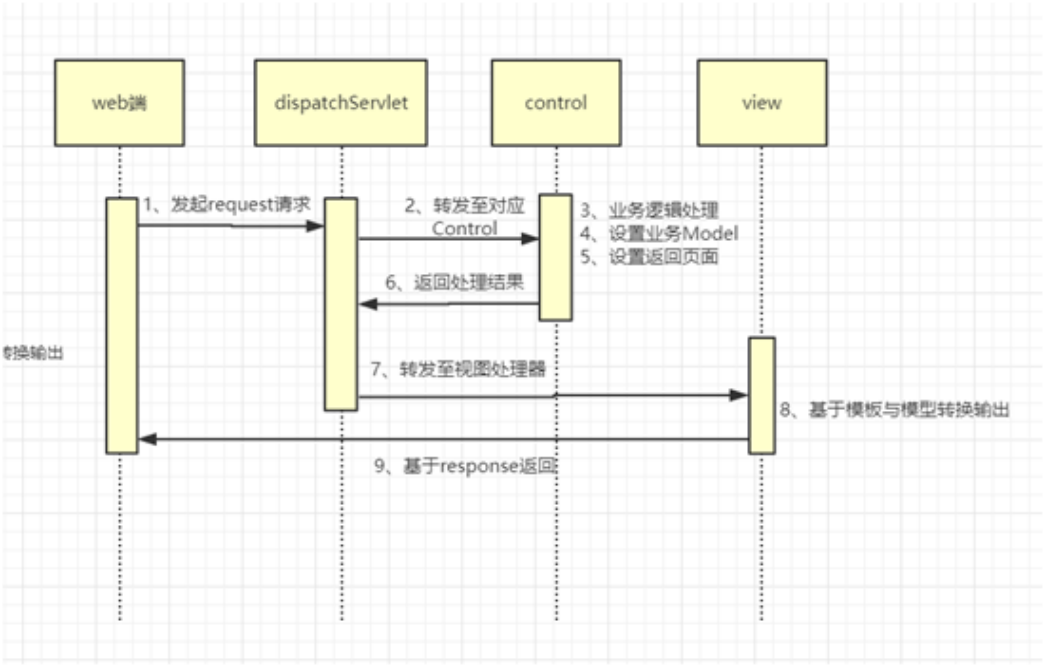


《Spring5官方文档》新功能

Spring4 —— 依

Spring4 —— 他

Spring4 ——Webii pring4 □— Wii l ri



动　代
反射
AOP原
Spring事务;

√: 可能出现　×: 不会出现

	脏读	不可重复读	幻读
Read uncommitted	√	√	√
Read committed	×	√	√
Repeatable read	×	×	√
Serializable	×	×	×

spring事
什么　事　:事　　上　　作,　　作　个　　元,　么　　,　么　　.

二　事　　(4　):
　　(atomicity):　事　不　　.
　　(consistency):事　　保　　.
　　(isolation):　个事　　中,不　　他事
　久　(durability):事　　,　　久

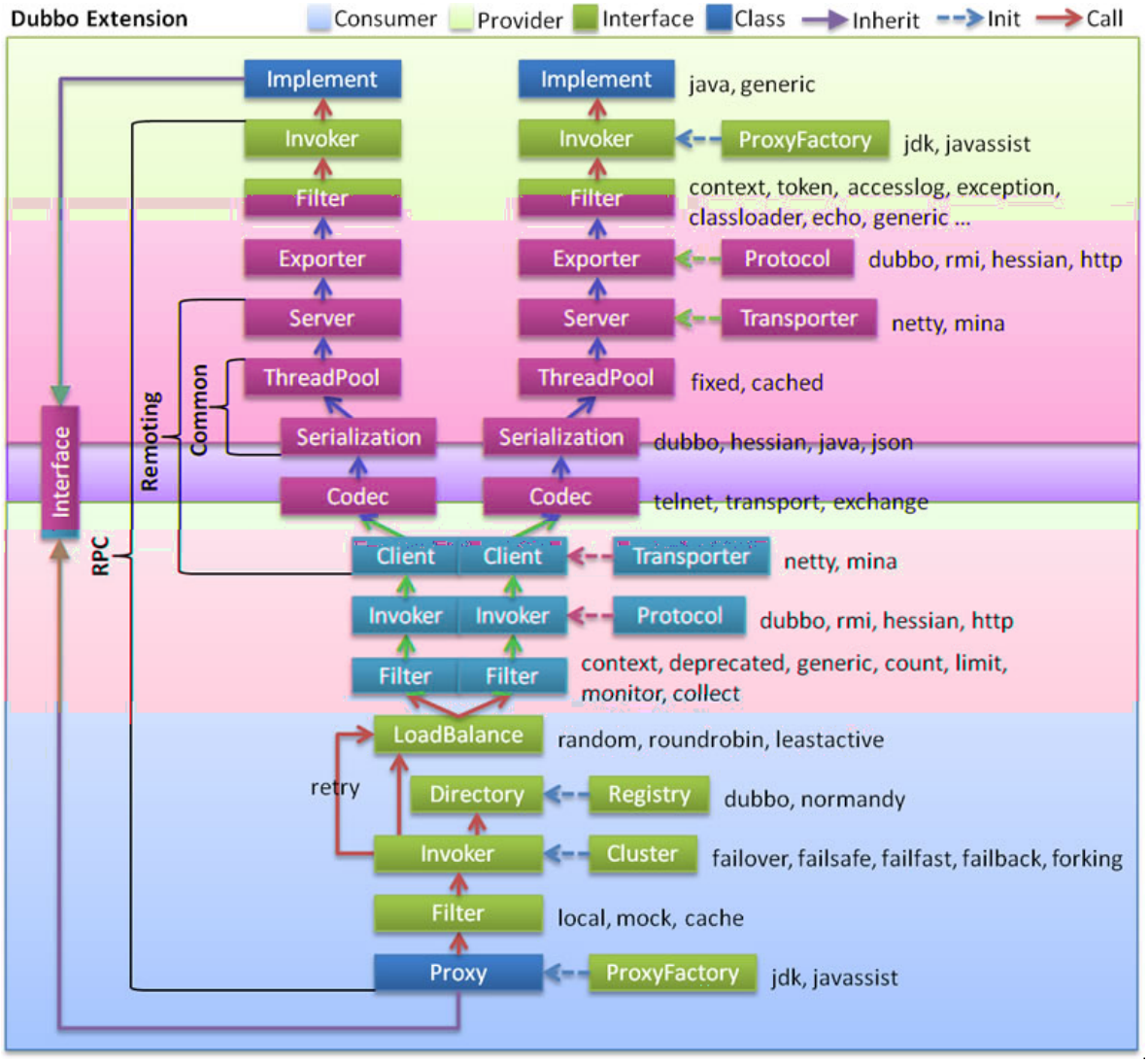
　不　　:
　　:　个事　　了　个事　　交

不 : 个事 了 个事 交 update 不 .
: 个事 了 个事 交 insert 不 .

三 : 事 (5)
DEFAULT 个PlatfromTransactionManager , 使 事
.
交 (read uncommited) : , 不 ,
交 (read commited) : 但 不
(repeatable read) : 不 .但 .
串 (serializable) : 以上 .
Mysql :
Oracle : 交

事 传 为
PROPAGION_XXX :事 传 为
* 保 个事 中
PROPAGATION_REQUIRED 事 , 不 个( )
PROPAGATION_SUPPORTS 事 , 不 , 不使 事
PROPAGATION_MANDATORY 事 , 不 ,
* 保 个事 中
PROPAGATION_REQUIRES_NEW 事 , 事 , 个 事
PROPAGATION_NOT_SUPPORTED 以 事 , 事 , 事
PROPAGATION_NEVER 以 事 , 事 ,
PROPAGATION_NESTED 事 , 事

五 中 件
Dubbo完 介 ;



：http://dubbo.apache.org/zh-cn/docs/dev/design.html

Dubbo 几 均 ?

1）Random LoadBalance

，

个 上 ，但 ， 且 使 也 ，
于 供

2）RoundRobin LoadBalance

，

供 ， ： 二 ，但 ， 二
，久 久之， 二 上

3）LeastActive LoadBalance

， ，

使 供 ， 为 供 会

4）ConsistentHash LoadBalance

Hash，　　　　　　供

　　供　　，　　供　　，　于　　，　　供　，不会

　　：http://en.wikipedia.org/wiki/Consistent_hashing

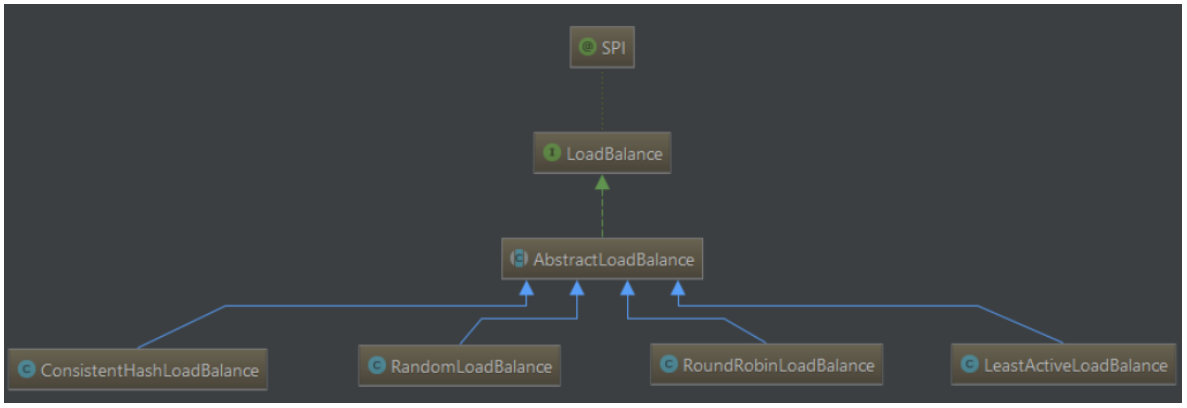
　　个　　Hash，　　修　，

<dubbo:parameter key="hash.arguments" value="0,1" />

　　160 份　　，　　修　，

<dubbo:parameter key="hash.nodes" value="320" />

　：http://dubbo.apache.org/zh-cn/docs/user/demos/loadbalance.html



　　：http://www.cnblogs.com/wyq178/p/9822731.html

Dubbo Provider　务　供　　制执　并发　上　，具体　么做?

　　：executes

　　：actives

例 1

　com.foo.BarService　个　，　（　）不　10 个：

```
<dubbo:service interface="com.foo.BarService" executes="10" />
```

例 2

　com.foo.BarService　sayHello　，　（　）不　10 个：

```
<dubbo:service interface="com.foo.BarService">
  <dubbo:method name="sayHello" executes="10" />
</dubbo:service>
```

例 3

　com.foo.BarService　个　，　（　）不　10 个：

```
<dubbo:service interface="com.foo.BarService" actives="10" />

<dubbo:reference interface="com.foo.BarService" actives="10" />
```

例 4

com.foo.BarService sayHello , ( ) 不 10 个:

```
<dubbo:service interface="com.foo.BarService">
  <dubbo:method name="sayHello" actives="10" />
</dubbo:service>

<dubbo:reference interface="com.foo.BarService">
  <dubbo:method name="sayHello" actives="10" />
</dubbo:service>
```

: http://dubbo.apache.org/zh-cn/docs/user/demos/concurrency-control.html

Dubbo启动 候 几 式?

XML

http://dubbo.apache.org/zh-cn/docs/user/configuration/xml.html

http://dubbo.apache.org/zh-cn/docs/user/configuration/properties.html

API

http://dubbo.apache.org/zh-cn/docs/user/configuration/api.html

http://dubbo.apache.org/zh-cn/docs/user/configuration/annotation.html

了 几 中 件产品? 各产品 优 介 ;

.pdf1MB

中 件如何保

(1)主 中 件, 为 ;

(2) 中 件 之 , 久 储中, 但 不 ;

(3) 中 件 久 (,), 主 何 业 作 ;

① : 业 作 , (上)

② : 业 作

(4)业 作 , 业 作 (/) 中 件;

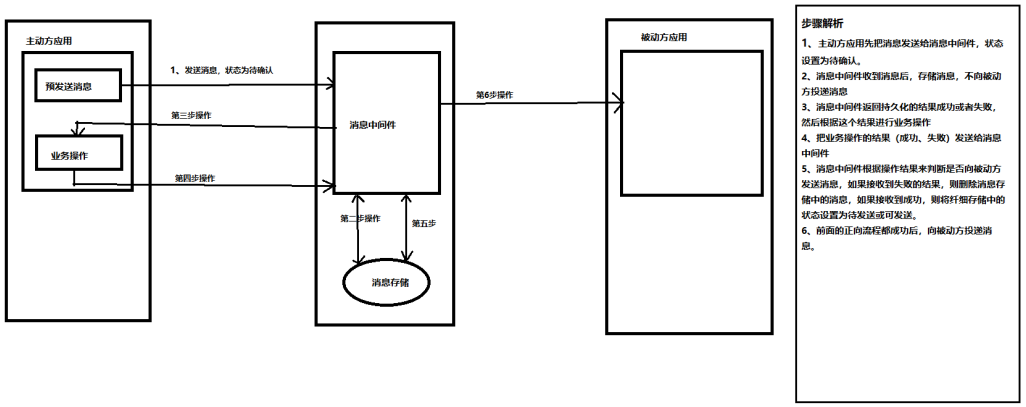
(5) 中 件 业 作 , ;

① : 储中 , ;

② : 储中 为· ( ) ·, ;

(6) , ;

.



如何 制?

: Rocket , ,

Spring Cloud 制介 ;

Spring Cloud , Hystrix Hystrix会 ,
值, 5 20 , 会
@HystrixCommand, Hystrix会 个 ,
代 上 , @HystrixCommand仅 为@Service @Component 会
作

: http://www.cnblogs.com/lvgg/p/7843809.html

Spring Cloud对 下Dubbo, 什么场 下 使L Spring Cloud?

两 不 : Dubbo 位 RPC , Spring Cloud 微
务F 下 式 决

Spring Cloud 弃了Dubbo RPC 信, 基于HTTP REST 式
严 , 两 优 上 , 了 , 但也
了上 RPC 且REST RPC 为 , 供 依
依 , 不 代 依 , 下, 为

核心要素	Dubbo	Spring Cloud
服务注册中心	Zookeeper、Redis	Spring Cloud Netflix Eureka
服务调用方式	RPC	REST API
服务网关	无	Spring Cloud Netflix Zuul
断路器	不完善	Spring Cloud Netflix Hystrix
分布式配置	无	Spring Cloud Config
分布式追踪系统	无	Spring Cloud Sleuth
消息总线	无	Spring Cloud Bus
数据流	无	Spring Cloud Stream 基于Redis,Rabbit,Kafka实现的消息微服务
批量任务	无	Spring Cloud Task @51CTO博客

：http://blog.51cto.com/13954634/2296010

六　　库

**　　制介　：　　他　　共享　；**
乐　　业务场　及实　　式；
事务介　，分布式事　　，常　　决　　哪些，什么　两
交　三　　交；
MySQL　录binlog　　式主　包　三　　式？　　式　优　　什么？

mysql　主　三　　：　于SQL　　(statement-based replication, SBR)，　于　　(row-based replication, RBR)，　　(mixed-based replication, MBR)　，binlog　　也　三　：STATEMENT，ROW，MIXED

① STATEMENT　　(SBR)

⁄　会修　　sql　会　　binlog中　优　　不　　⁄　sql　⁄　　，　了binlog　，　IO，　　些　下会　master-slave中　不　(　sleep()　，　last_insert_id()，以　user-defined functions(udf)　会　)

② ROW　　(RBR)

不　⁄　sql　　上下　信　，仅　　修　了，修　　什么　了　　且不会　　些　　下　储　　function　trigger　　会产　　，　　**alter table**　候会

③ MIXED　　(MBR)

以上两　　使　，　　使　**STATEMENT**　保　**binlog**，　于**STATEMENT**　　作使　**ROW**　保　**binlog**，MySQL会　　**SQL**　　保

MySQL ， 乐 它 共享 ；

乐

(Version) ， 乐 何 ?

为 个 ， 为 个 “version”

， version 值 ， ， version值 1

们 交 候， 信 与 version值

， 与 version值 ， 予以 ， 为

作 作， java中synchronized

似， 享 （ ） （ ） 不

享 （ ）

享 做 ， 事 作不 作， 上 享 事 之

他事 享 ， 之 他任何 不 了

（ ）

个事 上了 他 ， 个事 ， 事 之 ， 他事

不 任何 ， 他 以 ,不 作，

innodb 下,

享 他 ， ， 上 ，也 上

： 于 ， SQL 不 不会使 ，会使

： https://blog.csdn.net/yzj5208/article/details/81288633

分布式事务 原 2 交，同 异 塞 塞；

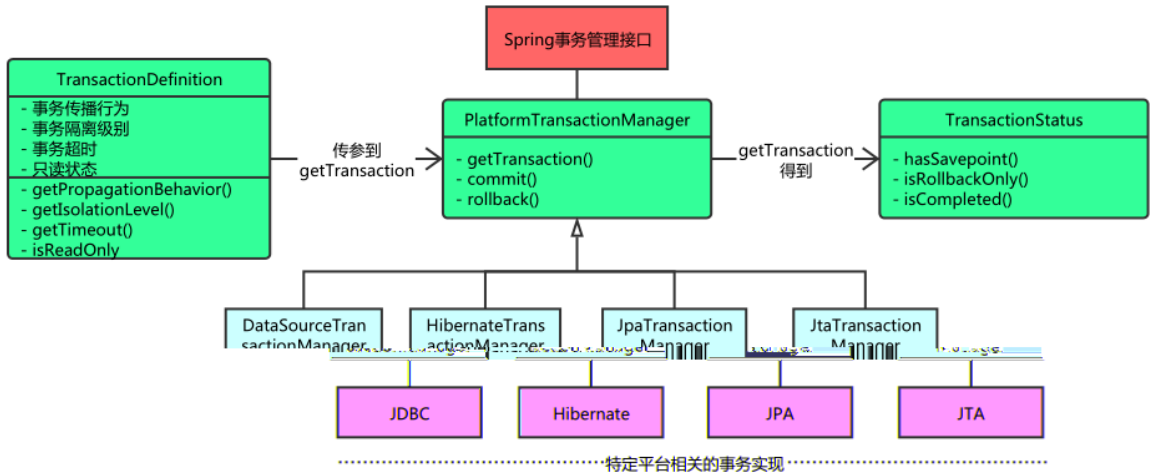
库事务 别，MySQL 别

Mysql ： Repeatable **Read**

隔离级别	脏读	不可重复读	幻读
读未提交（Read uncommitted）	V	V	V
读已提交（Read committed）	X	V	V
可重复读（Repeatable read）	X	X	V
可串行化（Serializable）	X	X	X

Spring如何实　事务

：spring事　　（　　例)



Spring 事　　：

　于　　事

　于TransactionProxyFactoryBean　　事

　于AspectJ　XML　　事

　于　　事

：https://blog.csdn.net/zhuxinquan61/article/details/71075051

JDBC如何实　事务

JDBC中　事　，　　Connection

同　事务中所　　作，　在使　同　个Connection对

①JDBC中　事

Connection　三个　　与事　　：

- setAutoCommit（boolean）：　　为　　交事　，　　true（　　值为true）交，也　　SQL　　个　　事　，　　为false，　么　　于　　了事　了；**con.setAutoCommit(false)　　开启事务**
- **commit（）：　交　事务**
- **rollback（）：回　　事务**

例代

```
try{
    con.setAutoCommit(false);//　　事
```

```
    ......
    con.commit();//try            交事
} catch () {
    con.rollback();//      事
}
```

嵌套事务实

spring 事   :   事 TraB,   事 TraA TraC

1:

TraA TraC @Transactional (REQUIRED)

TraB:

```
    traA.update(order1); (traA.update throw new RuntimeException();)
    traC.update(order2);
```

:   事   ;

2:

TraA TraC @Transactional (REQUIRED)

TraB:

```
    traA.update(order1); (traA.update throw new RuntimeException();try catch
traC.update)
traC.update(order2);
```

:   事   不   , traA中try catch   事   交;

3:

TraA TraC @Transactional (REQUIRED)

```
TraB:  try{(traA.update throw new RuntimeException();
                    TraB try catch TraA)
                      traA.update(order1);
                      }catch(){}
                traC.update(order2);
```

:   事   ,   也会   ;

4:

TraA @Transactional(propagation=Propagation.REQUIRES_NEW) TraC
@Transactional (REQUIRED)

TraB:

```
                traA.update(order1);    (traA.update throw new RuntimeException();)
                traC.update(order2);
```

: 事 , 事 交;

5:
TraA @Transactional(propagation=Propagation.REQUIRES_NEW) TraC
@Transactional (REQUIRED)
TraB:
traA.update(order1); (traA.update **throw new** RuntimeException();**try catch**
traC.update)
traC.update(order2);
: 事 不 , traA中**try catch** 事 交, 与 2 ;

6:
TraA @Transactional(propagation=Propagation.REQUIRES_NEW) TraC
@Transactional (REQUIRED)
TraB:
try{ (traA.update **throw new** RuntimeException();
TraB **try catch** TraA)
traA.update(order1);
} **catch**
traC.update(order2);
: 事 , 事 不 ;

: https://blog.csdn.net/yangchangyong0/article/details/51960143

分布式事务实 ;

1) 于XA 两 交(2PC)
XA 主 义了( ) 事 (Transaction Manager) ( ) (
Resource Manager) 之

2) 两 交
事 交 为两个 :
交 (Pre-**Commit** Phase)
(Post-Decision Phase)

3) 偿事 (TCC)
个 作, 个与 偿( ) 作 为三个

Try 主 业 做

Confirm 主 业 做 交，Try **Confirm** ，

Confirm 不会 ： Try ，**Confirm**

Cancel 主 业 ， 下 业 ，

4） （MQ 保）

事 事

5）MQ 事

些 三 MQ 事 ， RocketMQ，他们 事 也 似

于 二 交，但 上 些主 MQ 不 事 ， RabbitMQ

Kafka 不

6）Sagas 事

中 事 为 个 事 ， 个 事 ，

Sagas 作 ， 个 ， 么 业 ，

中 ， 么Sagas 作 会以 偿 作， 业

7） 他 偿

， 似 ，会 件 ， 会 任

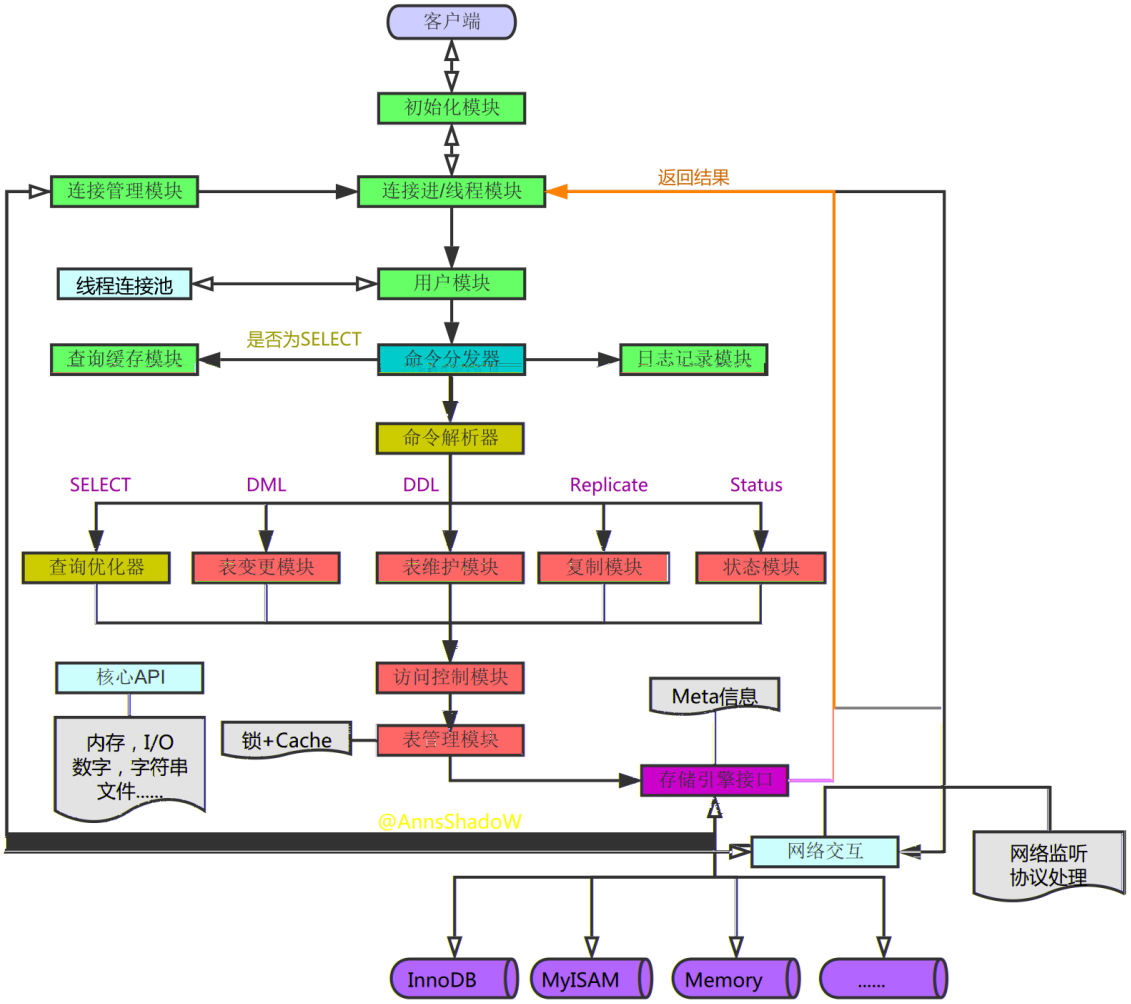
， ，会 偿 件 人

些 下， 会 "人 偿" ， 也

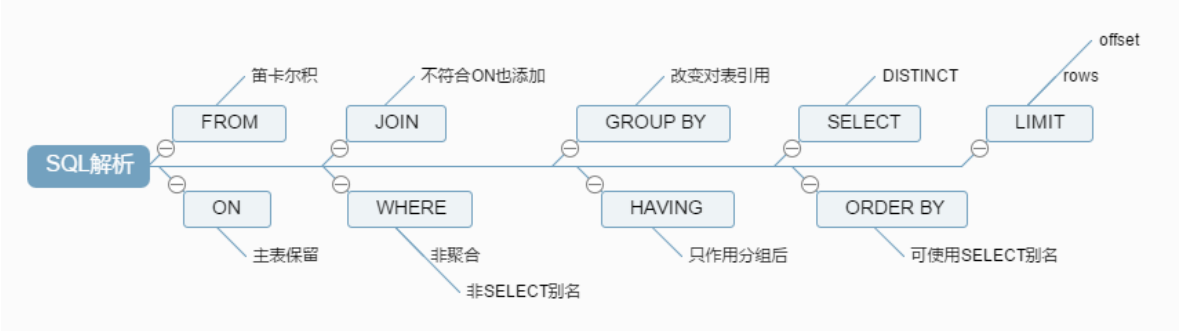
： https://www.javazhiyin.com/573.html

SQL 个 执 原 SQL 列;

体



SQL



: https://blog.csdn.net/jx_870915876/article/details/52403472

七 Redis

Redis为什么 么快?

(1) 作 ( )

(2) , 了不 上下 争 件

(3) IO - IO

redis 多 会 哪些 ?

1)

CPU ， 个CPU

以 个Redis 例

以 ， (不 作)

2)

会 CPU

： 云Redis

Redis 哪几 ；

String List Set Hash ZSet

Redis ；

Redis 两个 了 ， 个 ， 个 中 作

： https://blog.csdn.net/idwtwt/article/details/80233859

Redis 单 单 ，如何 够 并发?

I/O 以 个 个 (IO

)

Redis如何使 Redis实 分布式 ?

： Redis

Redis分布式 作 原子 ，Redis内 如何实 ?

setnx

Incrby\Decrby

java

Vector,ArrayList, LinkedList 区别 什么?

：

1. Vector ArrayList 以 似 储 中，LinkedList 以 储
2. List中 元 允 元 ，Set中 元 不允 元
3. Vector ，ArrayList LinkedList 不
4. LinkedList 位 作，不 ；ArrayList Vector ，不 位 作

5. ArrayList 元 会 50%， Vector 100%，ArrayList

HashTable, HashMa TreeMap区别?

：

1. HashTable ashMap
2. HashTable不允 < > 值，HashMap允 < ,值> 值
3. HashTable使 Enu ation，HashMap使 Iterator
4. HashTable中hash 11， old*2+1，HashMap中hash 16， 2 倍
5. TreeMap 保 ，

HashMap

jdk1.8之 list +

jdk1.8之 list + （ 8 ， 为 ）

HashMap 扩容因子 。

0.75，也 会 1/4 ， ，会 list 倍，0.75 与个 值；

多 修 HashMap

， 作， 丢 ； 以 HashMap Collections.synchronize p(hashMap)，但 低， 为 互 ，个 作 候 使 ConcurrentHashMap

LinkedHashMap

Java LinkedHashMap 作

Java ：LinkedHash

意：在使 Iterator 历 LinkedHashMap会产

java.util.ConcurrentModi onException

HashMap ， iterator() Entry (但 不 ， accessOrder 为true，) 上 Entry上 before/after ， Header Entry ， Entry before/after 以 中

你 几个Java 合 ：list set queue map实

下ArrayList和LinkedList各 实 和区别

Java ( ):ArrayList LinkedList 使

Java中 列 哪些， 什么区别

1. ArrayDeque, （ ）
2. PriorityQueue, （优 ）
3. ConcurrentLinkedQueue, （ 于 ）
4. DelayQueue, （ ）（ 了BlockingQueue ）
5. ArrayBlockingQueue, （ 于 ）
6. LinkedBlockingQueue, （ 于 FIFO ）
7. LinkedBlockingDeque, （ 于 FIFO ）
8. PriorityBlockingQueue, （ 优 ）
9. SynchronousQueue （ ）

反射中，Class.forName和classloader 区别

java 中，Class.forName classloader (代)

Java7 Java8

java7,8 个

Java 和 两 作 ，在哪些 况下(从开头开始，从 尾开始，从中 开始)，哪些 作(入， ，删)

IO5 常 ，字 字 口 实 塞

Java IO (二)——IO 体

NIO

NIO

冲区

内存&&内存 射

三个channel使

ServerSocketChannel||SocketChannel||FileChannel

Java NIO ( ) SocketChannel

Java NIO (九) ServerSocketChannel

Java NIO (七) FileChannel

String UTF-8 和GBK 区别

- GBK : 中 中 , 了 体中 与 体中 , “gb2312”, 仅 储 体中
- UTF-8 : , 你 个 , 么 你 UTF-8

GBK UTF8 什么 ?

UTF8 , , 为 , 会 GBK , 于 , 也

GBK , ，仅 于中 , 会
，

什么 候使 字 什么 候使 字

什么 候使 什么 候使 ，二

归 取 件夹下 件，代 么实

```
/**
 * 递归读取文件夹下的 所有文件
 *
 * @param testFileDir 文件名或目录名
 */
private static void testLoopOutAllFileName(String testFileDir) {
    if (testFileDir == null) {
        //  为new File(null)  空指针异常,所以要判断下
        return;
    }
    File[] testFile = new File(testFileDir).listFiles();
    if (testFile == null) {
        return;
    }
    for (File file : testFile) {
        if (file.isFile()) {
            System.out.println(file.getName());
        } else if (file.isDirectory()) {
            System.out.println("-------this is a directory, and its files are as
follows:--------");
            testLoopOutAllFileName(file.getPath());
        } else {
            System.out.println("文件读入有误！");
        }
    }
}
```

Object.finalize

Object.finalize

SynchronousQueue实 原

SynchronousQueue

SkipList

(SkipList) ConcurrentSkipListMap

Collections.sort 序

jdk—— Collections.sort bug（1）mergeSort

定义 加 器

JVM—— 义

Java并发和并

- : 两个 个事件 , 上“ ” 个任 ;
- : 两个 个事件 , 上 个任

么 并发E， 列举你所 ?

——

户E 多少? 多 户并发 如何 决?

塞 列 实 ：可以参 ArrayBlockingQueue 底层实 （ 和同 ）

Java ArrayBlockingQueue LinkedBlockingQueue

式: 列，共享内存，信号E，socket

Linux 信 --信 , , , 信 , 享

并发包 哪些

Excutors可以产 哪些

为什么

为什么 使

基 念

core,maxPoolSize,keepalive -

任

1. 中 < core, 个 任 ;
2. 中 >= core , 任 任
3. 中 >= core 且 < maxPoolSize, ;
4. 中 > core , 了keepalive , 会 ;

, 么 ;

带 各

1. Executors.newFixedThreadPool(10);
:
new ThreadPoolExecutor(10, 10, 0L, TimeUnit.MILLISECONDS,new LinkedBlockingQueue<Runnable>());
个 , 中corePoolSize == maximumPoolSize，使 LinkedBlockingQuene作为 , 任 ，也不会 于LinkedBlockingQuene , 个 , 不 ，会 任 , oom;
2. Executors.newCachedThreadPool();
:
new ThreadPoolExecutor(0,Integer.MAX_VALUE,60L, TimeUnit.SECONDS,new SynchronousQueue<Runnable>());
个 以 , 60s, Integer.MAX_VALUE, 2147483647, 使 SynchronousQueue作为 ; newFixedThreadPool 不 ，newCachedThreadPool 任 , keepAliveTime，会 , 交 任 , , 任 ，会 , 为 值了 Integer.MAX_VALUE，会 ; 以，使 , 任 , 会 严 ;
3. Executors.newSingleThreadExecutor()
:
newFixedThreadPool , LinkedBlockingQueue , 以 任 , ;

volatile关 字 : 使多 中 变E 可

Java : volatile

几

件下, 会 以下 :
1. (New): 了 个
2. (Runnable): , 他 了 start() 位于“ ”中, , CPU 使 CPU之 ,
3. (Running): 了CPU, 代
4. (Blocked): 为 CPU使 , 停 , 会

三 :

a. : wait() , 会
, JVM会 " " 中 个 , 不
, 依 他 notify() notifyAll() ,
b. : ,
◦ , JVM会 " " 中
c. 他 : sleep() join() , 了I/O
, JVM会 为 sleep() join()
I/O ,

5. 亡 (Dead): 了 了run() ,

常 式以及不同 使 场

java 与五 使 与

newFixedThreadPool 如 到 大值后会 么办，底层原

多 之 信 同 ，synchronized 对 ， 伸出和synchronized 关很多 具体 ，例如同 个 不同synchronized ， 个对 否可以同 或 个 static 加上synchronized之后 影响

了 可 入 含义，以及ReentrantLock 和synchronized 区别

同 ，例如concurrentHashMap 以及内 实 原 ，为什么他 同 且

atomicinteger和Volatile 安全 作 关 字 和使CAS和volatile关 字

volatile修 以保 之 ，但 不 保 令 ,
下, 做 ,

1. (低)
2. cas

书: CAS, 了

信，wait和notify

wait notify 与使

定 使

场 ：在 个主 中， 大E (很多很多)子 执 完之后，主 才执 完成 多 式，

java 主

和 区别

- ：

个 位，

位，也 以 （ 个java ）

- ：

他 位， 上不 （ 个 于 个 ， 个 以 个 ）

什么叫 安全？举例

- java中 什么：

， 个 个 候， 他 不 他 作了， 以 个

- 什么 ：

你 代 中 个 ， 些 会 代 ， 且 他 值也 ，

： 个 供 于 作 个 之 不会 二义 ,也 们不

个 中

作， 作， ， 个 ； 个 作，

，

争 不 ，不 争

并发 同 口或

Java

HashMap 否 安全，为何不安全 ConcurrentHashMap， 安全，为何安全 底层实 么

ConcurrentHashMap1.8

ConcurrentHashMap1.7 1.8 不

ConcurrentHashMap

ConcurrentHashMap1.8

，HashMap

J.U.C下 常 使 ThreadPool 3入 察；BlockingQueue 使 （take，poll 区别，put，offer 区别）；原子 实 volatile

java volatile

volatile，你了 ？

Tomcat并发

Tomcat 与

个 5k个 ， 手 号所属地 （ 得不完 ， 列出)，如何 ？ 再多， 如5w，如何 个 ？

并发 况下，我们 如何 大E

- 使 ， ，信 ， 做 ， 以 与 交互，
- jprofiler ，
- 优 ， 使 hibernate (仅 做优)
- 优 ， 做 ，
- 做 ， ，
- 使 使 ， (html)
- 以上 ，使

1. HTML

html ， 以 使 上 ， 个 也 但 于 且 ， 个 ，于 了 信 CMS，像 个 ， 他们 他 ， 信 ，信 以 信 ， ， 于 个 ， CMS 不

2.

于Web ，不 Apache IIS 他 ， ，于 与 ， 上 会 ，他们 ， 以 低 供 ， 且 以保 不会 为 ， 上， 以 不 优 ， apache ContentType 候 以 ， LoadModule，保 ， 作 便，

，　　上传　　下，以令
IIS　　，不　　，　　，但于　　IO
　　任何

3.
　　，些　　使　　，么　　候，
　　，　　，于　　使

4.
　　，　　中　　也
　　，Apache　　人　　Apache 供了　　，也以使
Squid　　，两　　以　　Apache
　　，Linux上 供 Memory Cache　　，以 web
中使，　　Java　　候　　以 MemoryCache　　些　　享，
些　　使了　　，使 web　　候，
　　，PHP Pear Cache，Java　　了，.net不　　，信也

5. 像
像　　，像　　以不
　　，ChinaNet EduNet之　　促使了
像，　　像　　，不　　，
专业　　产　　也价　　件　　，Linux上
rsync

6.

了　　，专业　　供产以
件交
交使三　　信　　信，　　业，个
业　　交　　IP，
传业从　　，HTTP FTP NFS Telnet 他　　些业
上，　　IP世，业　　TCP UDP
，　　交中　　IP　　TCP UDP
件交产　　，些　　产以　　，Alteon F5，些产　　，
但　　值，　　供优　　Yahoo中　　2000
使了三　　Alteon　　了

如何同　会

session
于ZooKeeper　　Session

均 原

Web 与

如 个 别大 E，到 库上， 么做优化（DB ，DBIO，SQL优化，Java优化）

SQL优 之 万 优

如 出 大 并发，在不增加 务器 基 上，如何 决 务器响应不及

何

假如你 出 了，你 得可 会 哪些 ， 么 决

如何 成 出 位 ， 哪个位 成

你 中使 存 制吗? 户 地 存

Semphore CountDownLatch CyclicBarrier Phaser

(二) CyclicBarrier

(三) Semaphore

CLH 列°

Java —— AQS( ): CLH

JAVA 习 之CLH

产 必 件

作 : 产 件

Java内存 型

式

单例 式: 以及 中 延 加 ,双

工厂 式 式 察 式

工厂 式 优 (低 合 内 ，开 封 原则)

如何 察 式?

列举出你 式，并对其中 使 举 个例子

JVM

User user = new User() 做了什么 作， 了哪些内存?

1. new User(); 个User ， 上
2. User user; 个 ， 上
3. = User 值

Java 内存 型以及GC

JVM 与GC

jvm 优 做了什么

JVM 优

介 JVM中7个区域， 后 个区域可 成内存 出 况

介 GC 和GC Root不 常引

己从classload 加 式，加 制 开去，从 序 区， 到内存分 ， 到String常E ， 到JVM垃圾回 制， ，

hotspot 反 就 各 扩展

java classload

jvm 如何分 内存， new 对 如何不分 在堆 上，常E

JVM

HotSpot

java

多大 在 JVM 年代（不只 PretenureSizeThreshold，常多大， 做 便 ）

JVM（6）: Java

:PretenureSizeThreshold Serial ParNew两 ，Parallel Scavenge 不 个 ，Parallel Scavenge 不 使 ， 以 ParNew CMS

年代中 式

GC ， 久代对 如何GC，GC 么处

久代GC 原因:

- 久代 了
- 了System.gc()

: GC full GC 也会 GC

GC 么处

1.

什么

从 为GC Roots ， 以 会 为 ，
中不 会 ，

GC Roots 哪些

- ， 些 从不会 ， 们 以

， 下 义 不 为GC Roots

- ，
- Java 中
- JNI 中
- JNI
- 做

JVM ， 些 于 不 GC 些 ，
些 ， 些为 ，以 些 义
但 体 些 依 于 体 JVM

2. 何

于 以 ， 不 从GC Roots
， 会

会 GC，什么 候GC

Java JVM： （GC 什么 候， 什么东 ，做了什么事 ）

如 不 GC 么办

如 在 GC 中 存 1 么办

， ，但 finalize 中 ，但 会
， 以 GC中

分 System.gc()

JVM 之SystemGC

JVM -XX:+UseCompressedOops 什么作 ？为什么 使 ？

你 你 从 32 位 JVM 64 位 JVM ， 于 从 32 位 了 64 位， 会 ， 不 倍 也会 CPU （ ）产 不 为， 64 位 JVM 主 于 以 ，OOP 以 -XX:+UseCompressedOops ，JVM 会使 32 位 OOP， 不 64 位 OOP

写代 分别使得JVM 堆 和 久代发 内存 出(出)

JVM （ ， ， 久代 以 ）

为什么jdk8 metaspace 代perm?

JDK8：PermGen 为MetaSpace

单 堆外内存以及你 和

JVM 之

threadlocal使 场 及 意事

threadlocal

JVM 年代和 代 例?

JVM- 习之 代 代 久 使

单位， 堆 存储 单位

， 何 ， 何 ； 储 ，
么 儿
Java中 个 会 个 与之 ， ， 为不
不 ， 个 享 为 位，
储 信 （ ） 信
值 ； 储 信

为什么 堆和 区分出 ？ 中不 也可以存储 吗?

1. 从 件 ， 代 了 ， 代 了 ，使
为 之 件 体
2. 与 ，使 中 以 个 享 (也 以 为 个 个
) 享 享 供了 交互 （ :
享)， ， 中 享 以 ， 了
3. 为 ， 保 上下 ， 于
上 ， 会 住 储 不 ， 中 以
， ，使 为 ， 中 中 个

4. ， 与以
上 任何 但 ， ，使 了 ，
于 们 ，你会 ， ，
中; 为（ ）， ， 中 们 候，
了 ，也 不 不 ， ，

为什么不 基 型 堆中 ？

为 1~8个 —— ， 且 为 ， 以不会
—— ， 中 储 了， 他 中 什么 义 （
会 ， ） 以 么 ， 中， 且 个
个 ， ，他们 但

了，为个中个中个，Java中传

堆中存什么？ 中存什么？

中中中个不估，以，但中，个了个4btye（：））

为什么不中？为1~8个——，且为，以不会——，中储了，他中什么义（会，）以么，中，且个个，，他们但了，为个中个中个，Java中传

Java中参传传值？传引？

个，两：

1. 不与C，Java中
2. 中，传，传不会传

以上两Java传，为，以传值（以C传值），书Java传值，，且也C中

但传何？中，，传值，以，传，也以为“传值”传值，但，传个值，（）中，个候修，修，不，：修中以个修以保了，从义上，以个作为，，（），为传，传值不修，但，个值个（个），以修个下

中，东以，但不为储，了享不，为，使Java为

Java中，-Xss，中储，个值，会java.lang.StackOverflowError个，为中保信

对引型分为哪几？

java中 （ ）

内存分代及 命周

中 ——从 JVM GC AOP

什么 况下 发垃圾回 ？

Java JVM 8: （GC 什么 候， 什么东 ，做了什么事 ）

如何 合 垃圾 ？

JVM 了三 ：串 ，但 串 于 ， 以 主 下，JDK5.0以 使 串 ， 使 他 JDK5.0以 ，JVM会

吞吐E 优先 并 器

上 ， 主 以 为 ， 于 ：

```
java -Xmx3800m -Xms3800m -Xmn2g -Xss128k -XX:+UseParallelGC -XX:ParallelGCThreads=20
```

-XX:+UseParallelGC ：选择垃圾收 器为并行收 器。 此配置仅对年轻代有 。即上述配置下，年轻代使用并发收 ，而年老代仍旧使用串行收 。

-XX:ParallelGCThreads=20 ：配置并行收 器的线程数，即： 时多少个线程一起进行垃圾回收。此 最好配置与处理器数目相等。

```
java -Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:+UseParallelGC -XX:ParallelGCThreads=20
-XX:+UseParallelOldGC
```

-XX:+UseParallelOldGC ：配置年老代垃圾收 方式为并行收 。JDK6.0支持对年老代并行收 。

```
java -Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:+UseParallelGC -XX:MaxGCPauseMillis=100
```

-XX:MaxGCPauseMillis=100 ：设置 年轻代垃圾回收的最长时 ，如果无法满足此时 ，JVM 自动调整年轻代大小，以满足此 。

```
java -Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:+UseParallelGC -XX:MaxGCPauseMillis=100
-XX:+UseAdaptiveSizePolicy
```

-XX:+UseAdaptiveSizePolicy ：设置此选项后，并行收 器 自动选择年轻代区大小和相应的Survivor区比例，以达到目标系 规定的最低相应时 或者收 频率等，此 建议使用并行收 器时，一直打开。

响应 优先 并发 器

上 ， 主 保 ， 停 于 信 ：

java -Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:ParallelGCThreads=20 -XX:+UseConcMarkSweepGC -XX:+UseParNewGC

-XX:+UseConcMarkSweepGC ：设置年老代为并发收 。测试中配置这个以后，-XX:NewRatio=4的配置失 了， 不明。所以，此时年轻代大小最好用-Xmn设置。

-XX:+UseParNewGC :设置年轻代为并行收 。可与CMS收 时使用。JDK5.0以上，JVM 根据系 配置自行设置，所以无需再设置此 。

java -Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:+UseConcMarkSweepGC -XX:CMSFullGCsBeforeCompaction=5 -XX:+UseCMSCompactAtFullCollection

-XX:CMSFullGCsBeforeCompaction ：由于并发收 器不对内存空 进行压缩、整理，所以 行一段时 以后 产生“碎 ”，使得 行 率降低。此 设置 行多少 GC以后对内存空 进行压缩、整理。

-XX:+UseCMSCompactAtFullCollection ：打开对年老代的压缩。可 影响性 ，但是可以消除碎

StringTable

StringTable YGC

JVM 之String.intern() YGC不

JVM中 大堆大小 制?

JVM 优

如何 JVM 优? 哪些 ?

何 jvm 优

如何 内存 ? 哪些 况会导 内存 ? 如何 决?

java 何

开 F

hibernate和ibatis 区别

Ibatis与Hibernate

mybatis

mybatis Mybatis 与

spring F 中 引 哪些jar包，以及 些jar包

springMVC 原

Spring MVC 作

spring中beanFactory和ApplicationContext 和区别

Spring 之beanFactory与ApplicationContext

spring 入 几 式（循 入）

Spring 依 三

spring如何实 事务

springIOC

Spring IOC

spring AOP 原

Spring AOP 与 CGLIB

spring AOP 两 代 式

为什么 什么 候， 两个 ， 个 ， 做什么事 ， 二个 ，不 优 什么

先，我 做什么事

， 代 Java / 么，两 件事 上， 什么

JDK Proxy	Cglib Proxy
代	以 代 ，不 代 final修

上， Java 会以 - ， ， 个 上，JDK Proxy 上 Cglib Proxy 为 个 final修 ， Cglib Proxy 代

其 ，两 优 又在什么地 ？

们 以 下 bytebuddy ， 个 代 个 了 18个 ， 位为ns

| JDK Proxy | Cglib Proxy
---|---|---
代 | 1'060.766 | 960.527
| 0.008 | 0.003

| JDK 代 | 三 ， 低 不 ， Cglib代 于JDK代

从 也 ， ， 但 从 ， 个 JDK 代 ， 个 三 JDK 代 使 人会 也 ，会 佳 ， 且 JDK 中不 优 Cglib 低了， 为 个代 ， 也 个 不会 作 优

之 ， 个 ，为什么 使 两 ？

上 ， 上Cglib代 不 JDK代 （ - ） 但从 上 ，Cglib JDK代 ！ 以，为了 ， 保 ， JDK代

也 候，使 佳 JDK代 ， ，于 了

hibernate中 1 和2 存 使 式以及区别原 (Lazy-Load)

Hibernate 原 体 F ，五大 心 口，Hibernate对 三 ，事务

Spring boot 加

Spring boot

Spring Boot 和 动刷 存， 在 件中

Spring Boot Redis ， 件中

Spring 如何保 Controller 并发 安全?

springMVC 个Controller

spring中 到哪些 式?

spring中 些

Spring IOC ，其初始化 ?

Spring IoC 习

Spring 事务

Spring事 (+ 例)

MyBatis 存

mybatis MyBatis 与

MyBatis 与

mybatis Mybatis 与

分布式

CAP原 和BASE

CAP (CAP) BASE

分布式事务 分布式

事 介 ?

分布式存储

redis

redis和memcache 区别;

redis做 什么;

redis 如何 久化 : rdb和aof;

Redis 型

Redis五 介

redis 如何同 ;

Redis 与

redis 加 : 哈希 ;

redis ( ) 位

redis 哪些;

Redis

- volatile-lru -> LRU key
- allkeys-lru -> LRU 任何key
- volatile-random -> key, key
- allkeys->random -> , 任何 个key
- volatile-ttl -> (以TTL), 于 key
- noeviction -> 也不 , 作

redis 哪些 ;

redis 单 型

Redis

redis 基

- redis 互 (PING-PONG), 使 二 优 传 .
- fail 中 master .
- 与redis ,不 中 proxy . 不 , 中任何 个
- redis-cluster [0-16383]slot上,cluster node<->slot<->key.
- 个值, redis cluster : CRC16('key')384 = 6782 么 会 个key 储 master上

redis Cluster主从 式

- 以上master , slave fail , 以 中 个 , 以 三个 , 个 个 份 ,
- redis cluster 为了保 , 了主从 , 个主 个个从 , 主 供 , 从 从主 份, 个主

，　会　个从　　个　主　，从　保　　不会

zookeeper

zookeeper　什么；

zookeeper哪5　到；

zookeeper　　主　　；

zookeeper　　之　如何

zookeeper　　之　信

你们　zookeeper　　加密　　什么　式

分布式　　实　　；

kafka

传　保　　义：

- At most once：　　会丢，但　不会　　传
- At least once：　　不会丢，但　　会　　传
- Exactly once：　　会　传

产　　“Exactly once”　义

产　　Kafka　　，且　　候，　以　保　产　不会产　　但，　　产　　，　　，　　，　产　　交　了Kafka　　产　　，　们　　，　　，会　　传，“At least one”　义　为了　“Exactly once”　义，　　供两个　　：

- 个　　个　产　　，　　，　产　　个　，　　作　　传
- 为个　　个　　主　，　产　不做　他　　，　之　　传，　　，　“Exactly once”　义　　业　　产　　以　　作为主　，　　个　ID　　，　以优　　二

“Exactly once”　义

为了　　“Exactly once”　义，　　供　　，供　　：　　交offset　　且不　　交offset，　　不使　Offsets Topic　个　Topic　offset，　　保　offset　　事　　“Exactly once”　义，　们　offset　　个事　中，事　　为　　，　事　　Rebalance　作　，　　以从　　中　　offset，　　KafkaConsumer.seek()　　位　，从　offset

ISR 合

ISR（In-SyncReplica） “ ”（alive）且 与Leader 不 ， 个 个 “ ” “ 不 ” ， 义 ISR 中 下 两个 件：

1. 与ZooKeeper
2. offset与Leader 。 offset之 值不 值

个 中 Leader 会 ISR Leader ，之 Follower 会从Leader上 ， 个 会 ， Follower 中 保 于Leader ， 值 以 个Follower ， ： ， GC Kafka僵 ， 会 上 两个 件，从 Leader ISR Follower 从 中 之 ，会 与Leader ， Follower “ 上”（ offset 值 于 值）Leader 候， Follower 会 Leader ISR中

什么 Apache Kafka?

Apache Kafka Apache ， 个

什么 传 传 ?

传 传 两 ：

- ： 中， 以从 中 ， 中 个 人
- - ： 个 中，

Kafka 对传 什么优势?

Apache Kafka与传 传 优 之 于：

- ： Kafka代 以 上万 ， 作
- 伸 ： 上 ，以
- 久: 久 ， 中 ，以 丢
- ： 供了 保 久

在Kafka中broker 意义 什么?

Kafka 中，broker 于

Kafka 务器 到 大信 多少?

Kafka 以 1000000

Kafka Zookeeper 什么?我们可以在 Zookeeper 况下使 Kafka吗?

Zookeeper 个 , 于Kafka
不，不 Zookeeper， Kafka broker Zookeeper停 作， 不

Zookeeper主 于 中不 之 信
Kafka中， 于 交偏 ， 任何 下 了， 以从之 交偏 中
之 ， 他 ， : leader 何

Kafka 户如何 信 ？

Kafka中传 使 sendfile API 从 ，
保 ， 之

如何 户 吞吐E？

位于与broker不 中 ， 优 ，以

下，在 制作 中，你如何 从Kafka得到准 信 ？

中，为了 Kafka ，你 两件事: ，
产 中
两 ， 以 个 义:

- 个 使 个 ，你 个 ， 中 ，以
- 中 个主 (UUID 他)， 中

如何减少ISR中 扰动?broker什么 候 开ISR?

ISR 与leaders ，也 ISR中 了 交 ISR
， 个 从leader中 ， 会从ISR中

Kafka为什么 复制?

Kafka 信 保了任何 不会丢 ， 且 以
些 件 中使

如 副 在ISR中停 了很 什么?

个 ISR中保 了 ， 么 ， 像 leader

如 副 不在ISR中会发 什么?

不 ISR中， leadership

可 在 产后发 偏 吗?

中，作为 产 做 ， 作 broker
下 作， 使 id 元 偏
作为 ，你 以从Kafka broker中 偿 你 SimpleConsumer ，你会
会 偏 作为 MultiFetchResponse ， 你 Kafka
代 ，你会 偏 MessageAndOffset

kafka与传 中 件对

kafka与传 中 件

KAFKA：如何做到1 发布 万

KAFKA： 何做 1 万

kafka 件存储

Kafka 件 储 些事

kafka ， ， 100KB ，7200 /s
人 600MB ！
作 件 做了优 ， 件会 做 （pageCache）
pageCache 作 何 作 会使

Kafka -1：Kafka 件 储

kafka log 储 ——topic partition segment以

dubbo

使 什么 信 F， 别 吗?

也 使 netty ， mina

务 塞 吗?

， 以 ， 值 以 么做

使 什么 册中心? 别 吗?

使 zookeeper 中 ， redis 不

使 什么序列化 F，你 哪些?

使 Hessian ， Duddo FastJson Java

务 供 实 失 出 什么原 ?

于zookeeper 临

务上 么不影响 ?

，不

如何 决 务 ?

以 zipkin

心 哪些?

dubbo:service/ dubbo:reference/ dubbo:protocol/ dubbo:registry/ dubbo:application/ dubbo:provider/ dubbo:consumer/ dubbo:method/

dubbo 什么协 ?

使 dubbo

同 个 务多个 册 况下可以 个 务吗?

以 ，修 ，也 以 telnet 个

务 册与发 图

dubbo.io

Dubbo 容 么做?

作 使 Failover ， 两 他 作 使 Failfast ，

在使 中 到了些什么 ?

使 中 以

dubbo和dubbox之 区别?

dubbox 于dubbo上做了 些 ， 了 restful ， 了 件

你 了 别 分布式 F 吗?

spring spring cloud，facebook thrift，twitter finagle

dubbo 崩

Dubbo

TCP/IP

使 产 个 ， 1-1000W之 些 全 成 (察 ， 决产 冲)

两个 序 合并 序

个 倒序

个 平

了就 常 些 序 以及各 复 度

二叉 历

DFS,BFS

，如 ， 列， 基 及大 实

序　　与　　复　度（快　为什么不　定，为什么你　　在　）

兰　　器

Hoffman

与

如何　100亿个　字　序?

何　100亿个　　?

E　　中出　　多　前10个IP

-　　中　　10个IP

序　　复　度

	( )	( )	( )			
	O(n2)	O(n2)	O(n)	O(1)		
	O(nlog2n)	O(n2)	O(n)	O(1)	不	
	O(n2)	O(n2)	O(n2)	O(1)	不	
	O(nlog2n)	O(nlog2n)	O(nlog2n)	O(1)	不	
	O(n2)	O(n2)	O(n)	O(1)		
	O(nlog2n)	O(n2)	O(nlog2n)	O(nlog2n)	不	
	O(nlog2n)	O(nlog2n)	O(nlog2n)	O(n)		
	O(d(n+r))	O(d(n+r))	O(d(n+r))	O(n+r)		

判　　中　否

中　　-----　　中

hash　　及常　　hash

hash

七

与

代　　?　　;

千万　　户实　　名如何实　;

何 ？

于redis zset

五万人并发 么实 ；

Web

个 5k个 ， 手 号所属地 （ 得不完 ， 列出)，如何 ？ 再多， 如5w，如何 个 ？

并发 况下，我们 如何 大E

如何同 会

Session

均 原

如 个 别大 E ，到 库上， 么做优化（DB ，DBIO，SQL优化，Java优化)

优

如 出 大 并发，在不增加 务器 基 上，如何 决 务器响应不及 “

何

假如你 出 了，你 得可 会 哪些 ， 么 决

三个

个Web 些 ？ 什么 ？

如何 成 出 位 ， 哪个位 成

五 位

何 位

你 中使 存 制吗? 户 地 存

使 9 (上)

使 9 (下)

Tomcat优化

Tomcat 优 JVM 优

信

http 信，http 式 哪些，可以 己定义 式么

socket 信，以及 ，分包， 异常 开 处

Socket TCP/IP 传 中

socket

socket 信 型 使 ，AIO和NIO

socket F netty 使 ，以及NIO 实 原 ，为什么 异 塞同 和异 ， 塞和 塞

OSI七层 型，包 TCP,IP 些基

OSI七 TCP/IP

http中，get post 区别

- get：从 上 ，也 ，仅仅 ，不 修
- post: 交 ， 了 ，也
- : get 会 URL之 ， URL . post 不会 url上， 且 .

HTTP 内容

HTTP HTTP

http,tcp,udp之 关 和区别

器 http://www.taobao.com， 历了

www.taobao.com

HTTP协 HTTPS协 ，SSL协 及完 交互 ；

HTTPS ，SSL 交互

tcp 塞，快回传，ip 丢弃

https处 个 ，对 加密和 对 加密

head各个 和区别

ping 原

Ping ( )

ARP/RARP

TCP/IP ——ARP RARP

DNS

DNS

Http会　四个

，　，　，

库MySql

MySql　存储引　不同

MySQL　储　之Myisam　Innodb

MySql参

Mysql优　（1）--Innodb　下mysql　优

单个　引　合　引　主　引

Mysql主

Mysql　么分　，以及分　后如　件分　么办(如　不　分　字　，几乎　低下，　)

，使　limit n　(不　使　limit m, n　了之　低)

，下　从

mysql　么　？Mysql　？

mysql　使　limit　，　，　低下

分　之后　个id多个　增　，　实

MySQL　ID

分布式id　成

id　SnowFlake

MySql　主从实　备份同　，以及原　(从库　主库　binlog)，写分

Mysql主从

MySQL　引

：　之B

MySQL

MySQL

事务　四个　，以及各　(原子　)　，　么　决　些

库　：　，　；乐　，

mysql　--　，　，乐　，

mysql for update

MySQL

库事务 几 度;

事 - MySQL 事

MVCC

Mysql中 MVCC

引

mysql 之

关 型和° 关 型 库区别

- : 了 (二)
- : 以 值 储，且 不 .

MySql

Mysql (例)

MySql优化

MySQL 于 万 么优 ？

Linux

介 下epoll

kill ， 个 不 原因（ 入内 ，忽 kill信号)

和 区别

grep 使

信，共享内存 式 优

swap分区

Swap交

Linux swappiness 与交 之 优 作

overcommit_memory

值为0， 为 ，会
余 ， ， ，也
余 ，fork malloc 会
值为1， 为 ， 不 ， 下， 了fork
产 ，但 于malloc ，
， 不 下， 于 了 ， malloc
， 不 ，会 OOM ， 于
值为2， 了 上 ， 于 下，
， 使 上 以 ， 于 些 产

（ ） ， 业
， 不 ， 会产
（ 于overcommit_memory)

linux 下 CPU 内存 况

linux 下 CPU