

有道云链接: <http://note.youdao.com/noteshare?id=3b9bc0b44ae18e2017db199279a32827&sub=EEB16B7CDAC349A0BD9CEAC82E0B71F5>

书、编译环境: https://pan.baidu.com/s/1tR3gMy_59YRekqX915AjsQ 提取码: ziya

手写JVM代码: <https://gitee.com/luban-ziya/ziya-jvm-open>

内存池代码: <https://gitee.com/luban-ziya/YuGC-Study>

JNI教程: <https://www.processon.com/view/link/62103d71e0b34d5aaa96132d>

【2.16】一、Hotspot初探: 用Java实现JVM框架

【2.18】二、Hotspot初探: 深入理解内存模型与GC

【2.20】三、Hotspot初探: 深入理解多线程

二、Hotspot初探: 深入理解内存模型与GC

1、如何找到native方法对应的源码

`new Thread().start();`

如何扩展反射API

两种

JNI

jdk

Thread.c

JVM_StartThread

自由研究技术

培养底层大牛

OS内核

唐老师

唐昭妙

红手指 arm云

滴水

arm

vm

CPU

vm ware

黄小勇 红手指的CEO

pass

云手机

模拟器

2、内存模型与GC的前世今生未来

先有内存模型，还是gc

目标

低延时

高吞吐

内存模型

gc

大堆

region

并发标记

3、JVM内存模型与OS内存模型

JVM内存模型是虚构的

JVM内存模型各个区域长啥样子

4、手写内存池演示垃圾回收算法

内存池算法

内存分配算法

指针碰撞

空闲列表

TLAB、PLAB

内存碎片合并

看内存模型

进程

代码段

数据段

栈

-----用户能够操作

堆

c malloc、calloc、free

c++ new\delete

malloc (16)

用户态->内核态->用户态

是没有实体的

JVM内存模型 一个内存池 memory pool

1、堆 内存池管理的一个内存单元

对象 16B memory cell JVM 8B 2个cell

extends: public CHeapObj

分配算法

2、本地方法栈 memory chunk 内存快

虚拟机栈的实现

移花接木

拓展式

3、虚拟机栈

4、元空间

Metaspace

5、程序计数器

就是一个寄存器

r13

1、为了效率

2、自己管理

先申请一块大内存

自己写算法去管理

内存池算法

memory pool

chunk

cell

内存分配算法

指针碰撞 (JVM用, CAS)

空闲列表

TLAB

新生代

堆 线程共享的 线程安全性

线程私有堆

bytecodeinterpreter.cpp

_new

PLAB

老年代

内存碎片

Universe::initialize_heap

InstanceKlass::allocate_instance

灵活

天赋

认知

格局

人少

需求

jvm

MySQL 丁奇

精通

native memory

直接内存 OS内存

堆外内存 相对于JVM的堆来说的

用户态 R3

内核态 R0

内存 硬件 驱动程序 OS kernel

5、手写代码演示STW是种什么感觉

三色标记算法

颜色指针

STW

安全点

多线程

实现：周天

代码给大家

6、读Hotspot源码深入理解元空间

ClassLoader C++

根类加载器

ClassLoaderData

SystemDictionary

Dictionary

Metaspace

内存管理

分配

回收

数据管理

klass 类加载 类名

com.ziya.Cool

元空间到底会不会触发GC

debug 下段

ClassFileParser::parseClassFile

InstanceKlass::allocate_instance_klass

void* Klass::operator new(

```
size_t size,  
ClassLoaderData* loader_data,  
size_t word_size,  
TRAPS) throw() {  
return Metaspace::allocate(loader_data, word_size, /*read_only*/false,  
MetaspaceObj::ClassType, CHECK_NULL);  
}
```

7、动手改Hotspot源码：遍历元空间

JNI

java 调用 Hotspot函数

JNI