

实验 5 RAID 实验报告

班级	计算机 1 班	实验日期		实验成绩	
姓名	李梓涵	学号	34520212201574		
实验名称	实验 5 RAID 实验				
实验目的、要求	编译 Disksim，测试单个磁盘的性能（Response time） 配置 RAID0、RAID1、RAID5 并做性能测试 探究性实验（2 选 1）				
实验内容、步骤及结果	<div>一、 解压 DiskSim-64.zip，编译并运行 DiskSim</div> <pre>[cs211574@mc000 valid]\$ rm -rf cheetah4LP.outv [cs211574@mc000 valid]\$./src/disksim cheetah4LP.parv cheetah4LP.outv validate cheetah4LP.trace 0 DiskSim v2.01.016 [cs211574@mc000 valid]\$ grep 'IOdriver Response time average' cheetah4LP.outv IOdriver Response time average: 5.070471</pre> <div>二、 配置 RAID0、RAID1、RAID5 并做性能测试</div> <div>1. RAID5</div> <div>修改 Redundancy scheme 属性为 Parity_rotated 以模拟 RAID5，且将磁盘数改为 5 个，这样实际使用磁盘数为 4 个，对应后面的有效空间大小为 2056008×4=8224032</div> <div><pre>disksim_logorg org0 { Addressing mode = Array, Distribution scheme = Striped, Redundancy scheme = Parity_rotated, devices = [disk0 .. disk4], stripe unit = 64, Synch writes for safety = 0,</pre></div> <div>修改设备有效容量为 8224032</div>				

```

disksim_synthio Synthio {
    Number of I/O requests to generate = 10000,
    Maximum time of trace generated = 1000.0,
    System call/return with each request = 0,
    Think time from call to request = 0.0,
    Think time from request to return = 0.0,
    Generators = [
        disksim_synthgen { # generator 0
            Storage capacity per device = 8224032,
            devices = [ disk0 .. disk3 ],
        }
    ]
}

```

运行结果:

```

[cs211574@mc core valid]$ ../src/disksim synthraid5.parv synthraid5.outv ascii 0 1
DiskSim v2.01.016
[cs211574@mc core valid]$ grep "IOdriver Response time average" synthraid5.outv
IOdriver Response time average:          24.225320

```

2. RAID1

为保持有效空间大小一致, RAID1 下要修改磁盘数量为 8 个, 这样 4 个盘作为备份盘, 仍然有 4 个盘正常读写。

修改 Redundancy scheme 属性为 Shadowed 以模拟 RAID1

```

disksim_logorg org0 {
    Addressing mode = Array,
    Distribution scheme = Striped,
    Redundancy scheme = Shadowed,
    devices = [ disk0 .. disk7 ],
    Stripe unit = 64,
    Synch writes for safety = 0,
    Number of copies = 2,
}

```

按照 1 中修改设备有效容量为 8224032

测试结果为:

```

[cs211574@mc core valid]$ ../src/disksim synthraid1.parv synthraid1.outv ascii 0 1
DiskSim v2.01.016
[cs211574@mc core valid]$ grep "IOdriver Response time average" synthraid1.outv
IOdriver Response time average:          20.896989

```

3. RAID0

修改 Redundancy scheme 属性为 Noredun 以模拟 RAID0。

为保持有效空间大小一致, RAID0 下需要 4 块磁盘, 即[disk0...disk3]。

```
198 disksim_logorg org0 {  
199     Addressing mode = Array,  
200     Distribution scheme = Striped,  
201     Redundancy scheme = Noredun,  
202     devices = [ disk0 .. disk4 ],  
203     Stripe unit = 64,  
204     Synch writes for safety = 0,  
205     Number of copies = 2,
```

按照 1 中修改设备有效容量为 8224032
运行结果为:

```
[cs211574@mc core valid]$ ../src/disksim synthraid0.parv synthraid0.outv ascii 0 1  
DiskSim v2.01.016  
[cs211574@mc core valid]$ grep "IOdriver Response time average" synthraid0.outv  
IOdriver Response time average: 20.083489
```

4. 结论: 在有效空间一致的情况下, RAID0 的平均 IO 响应时间最短, 因此 RAID0 的速度最快。

三、探究实验 B

1. 磁盘个数

保持条带大为 64, 改变磁盘个数, 注意此时要对 disk、bus 等定义做修改

```
110 # component instantiation  
111 instantiate [ statfoo ] as Stats  
112 instantiate [ ctrl0 .. ctrl11 ] as CTRL0  
113 instantiate [ bus0 ] as BUS0  
114 instantiate [ disk0 .. disk11 ] as HP_C3323A  
115 instantiate [ driver0 ] as DRIVER0  
116 instantiate [ bus1 .. bus12 ] as BUS1  
117 # end of component instantiation
```

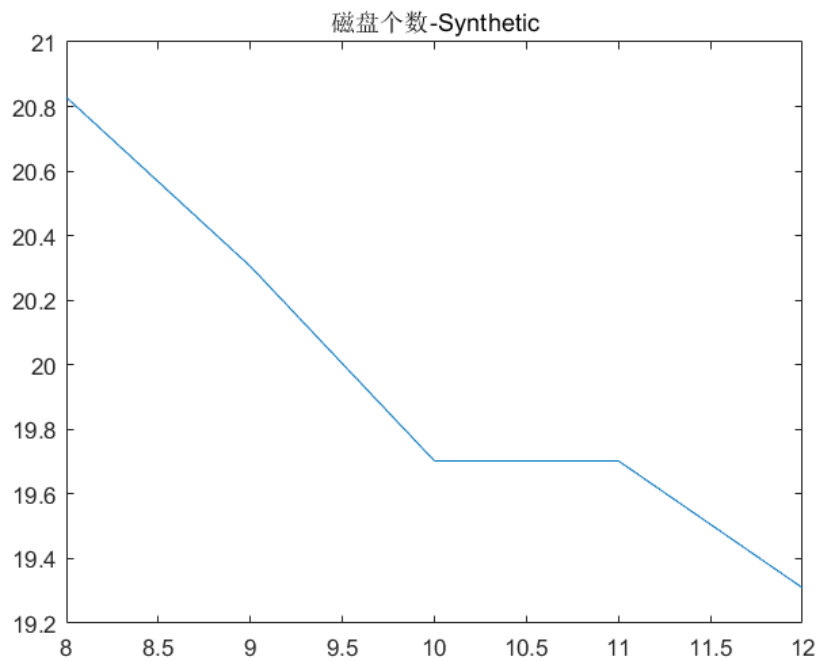
```
188 ],  
189 disksim_ctrl ctrl9 [  
190     disksim_bus bus10 [  
191         disksim_disk disk9 [  
192             # end of bus10  
193         ]  
194     # end of ctrl9  
195 ],  
196 disksim_ctrl ctrl10 [  
197     disksim_bus bus11 [  
198         disksim_disk disk10 [  
199             # end of bus11  
200         ]  
201     # end of ctrl10  
202 ],  
203 disksim_ctrl ctrl11 [  
204     disksim_bus bus12 [  
205         disksim_disk disk11 [  
206             # end of bus12  
207         ]  
208     # end of ctrl11  
209 ],  
210 # end of disksim_logorg  
211 ]
```

运行结果如下表, 此时 Stripe unit = 64, Parity stripe unit = 64,

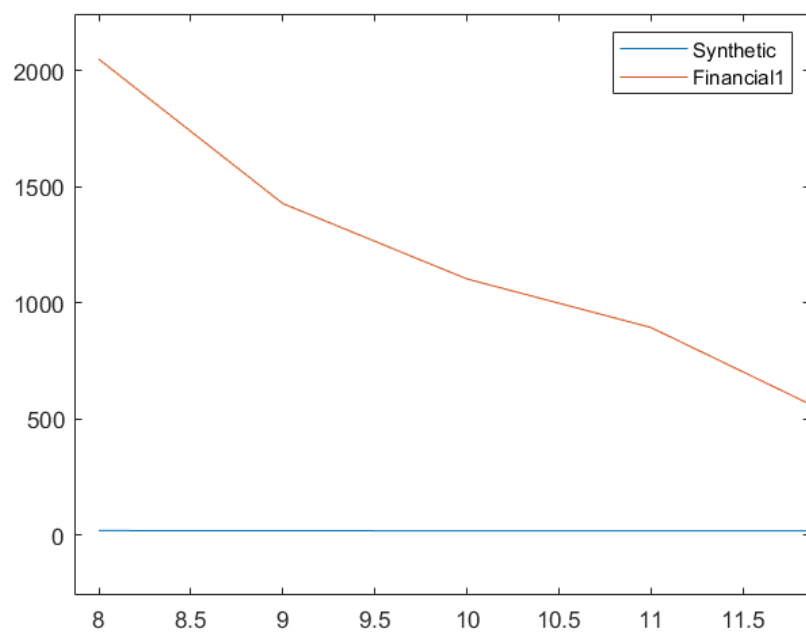
磁盘个数	8 个	9 个	10 个	11 个	12 个
Synthetic	20.828627	20.304895	19.702550	19.702550	19.309508
Financial1_10k.ascii	2048.4437	1427.3613	1103.2566	894.50524	512.68609

在作图结果如下:

对于 Synthetic 情况，作图如下



将两种情况一起作图如下



由图像可知，磁盘性能受磁盘个数影响比较明显，无论是 Synthetic 负载还是真实 workload 负载，随着磁盘数量增加，花费时间都减少，尤其是 Financial1 情况下，时间下降明显，性能提升明显。

2. 条带大小

修改 disksim_logorg org0 下 Stripe unit 和 Parity stripe unit，二者需要保持一致。

```

202     devices = [ disk0 .. disk8 ],
203     Stripe unit = 128,
204     Synchron writes for safety = 0,
205     Number of copies = 2,
206     Copy choice on read = 6,
207     R/W vs. reconstruct = 0.5,
208     Parity stripe unit = 128,
209     Parity rotation type = 1,
210     Time stamp interval = 0.000000,

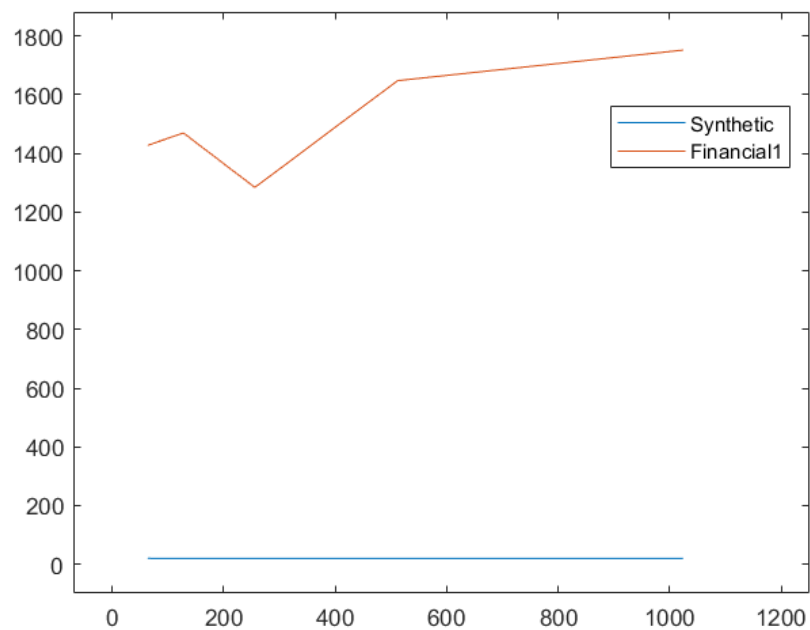
```

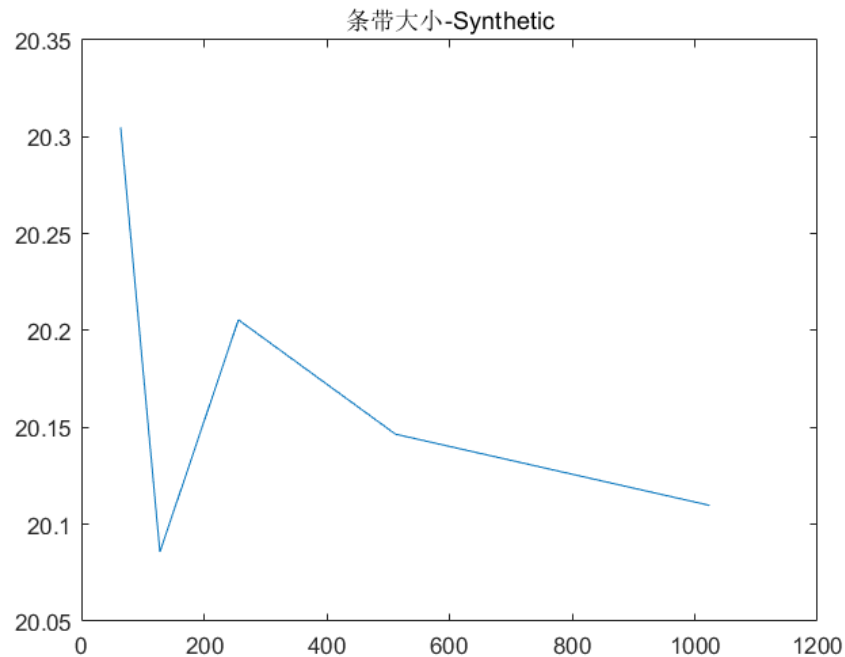
运行结果如下表，此时 devices = [disk0 .. disk8],

Storage capacity per device = 16448064

条块大小	64	128	256	512	1024
Synthetic	20.304895	20.085713	20.205540	20.146542	20.109807
Financial1	1427.3613	1469.2066	1283.8124	1647.5219	1751.5377
_10k.ascii	35	26	85	60	52

作图结果如下：





有这两幅可以看出，磁盘数量不变时，条带大小对于读写性能的影响没有明显的规律，而是在一定范围内波动变化的，因此可以认为磁盘对于条带大小的变化不是十分敏感。但是，当负载较大时，对应的波动范围也变大。

四、思考题

考虑以下应用场景，选择你认为合适的 RAID 设计：RAID 模式，条带大小（给出大致范围即可）。并给出理由

A.非线性编辑工作站（做视频编辑的电脑）

RAID0, 32-64; 涉及到视频编辑时文件较大，一方面需要提高读写速度，因此设置较小的条带大小，提升并行性，另一方面要提高恢复效率，选用 RAID0，一个磁盘出错不会带来太大的损失。

B.web 服务器

RAID1, 128; Web 服务器对数据的可靠性要求比较高，因此选用 RAID1，使用一半的磁盘做冗余备份，磁盘出现错误时仍然能工作。

C.代理服务器

RAID1, 128; 代理服务器与 Web 服务器类似，对于数据的可靠性和稳定性要求较高

D.FTP 服务器

RAID5, 128; RAID5 采用奇偶校验的方式，更容易发现错误出现的位置，符合 FTP 文件服务器的特性。

E.一卡通帐户数据服务器

RAID5, 128; 一卡通账户数据服务器对于数据的一致性要求比较高，选用 RAID5 同时读写所有磁盘，同时也容易找出出错的磁盘，及时纠错。

总结	<ol style="list-style-type: none">1. 在本次实验中，我对 RAID0、RAID1、RAID5 有了一定的了解。其中，RAID0 无冗余备份盘，将数据放在 N 块磁盘中并发读写；RAID1 就是 disk0-n 都用，一半作为冗余备份；RAID5 存储情况与 RAID0 类似，但是在 disk0 中采用奇偶校验2. 在修改 DiskSim 中磁盘数时，要对对应的组件初始化时的 Ctrl、disk、bus 数量进行修改，同时也要增加 bus0 中的 disksim_ctrl 数量。3. 对于 DiskSim 的解压问题，主要是在向服务器上传和修改文件时保磁盘空间不足的错误，猜测可能是同时操作服务器上文件的同学太多了，导致了错误。
----	--