

第三十八章-廉价冗余磁盘阵列(RAID)

预备知识

本部分介绍 raid.py，这是一个简单的 RAID 模拟器，可用于增强你对 RAID 系统的工作原理的了解。它具有许多选项，如下所示：

Usage: raid.py [options]

```
Options:
-h, --help                查看帮助信息
-s SEED, --seed=SEED      设置随机种子
-D NUMDISKS, --numDisks=NUMDISKS
                           设置 RAID 磁盘数量
-C CHUNKSIZE, --chunkSize=CHUNKSIZE
                           设置 RAID 块大小
-n NUMREQUESTS, --numRequests=NUMREQUESTS
                           需要模拟的请求数量
-S SIZE, --reqSize=SIZE   请求大小
-W WORKLOAD, --workload=WORKLOAD
                           使用"rand"或"seq"指定随机或顺序工作方式
-w WRITEFRAC, --writeFrac=WRITEFRAC
                           设置写入所占百分比（例如设置为，100，则所有请求为写入，设置为0，则
所有请求为读取）
-R RANGE, --randRange=RANGE
                           请求块范围（当使用"rand"工作方式时生效）
-L LEVEL, --level=LEVEL   RAID 等级（0，1，4，5）
-5 RAID5TYPE, --raid5=RAID5TYPE
                           RAID-5 left-symmetric "LS" or left-async "LA"
-r, --reverse              不显示逻辑读写操作，而是显示读写物理操作详情
-t, --timing               use timing mode, instead of mapping mode
-c, --compute              计算结果
```

-C 标志允许您设置 RAID 的块大小，而不是使用默认大小（默认每个块 4 KB）。

可以使用 -S 标志让模拟器调整每个请求的大小。

默认工作负载为随机访问块；使用 -W 探索顺序访问的行为。

使用 RAID-5 时，可以使用两种不同的布局方案，即左对称（left-symmetric）和非左对称（left-asymmetric）。使用 -5 LS 或 -5 LA 来使用（-L 5）。

最后，在计时模式（-t）中，模拟器使用了一个非常简单的磁盘模型来估计一组请求所花费的时间，而不仅仅是关注映射。在这种模式下，随机请求需要 10 毫秒，而顺序请求则需要 0.1 毫秒。

假定该磁盘的每个磁道的块数量很少（100），并且磁道的数量也类似（100）。

因此，您可以使用模拟器来估计某些不同工作负载下的 RAID 性能。

在其基础模式下，您可以使用它来了解不同的 RAID 级别如何将逻辑块映射到物理磁盘与计算偏移量。例如，假设我们希望看到具有四个磁盘的简单条带化 RAID（RAID-0）如何进行映射。

```
prompt> ./raid.py -n 5 -L 0 -R 20
```

```
...
```

```
LOGICAL READ from addr:16 size:4096
Physical reads/writes?

LOGICAL READ from addr:8 size:4096
Physical reads/writes?

LOGICAL READ from addr:10 size:4096
Physical reads/writes?

LOGICAL READ from addr:15 size:4096
Physical reads/writes?

LOGICAL READ from addr:9 size:4096
Physical reads/writes?
```

在此示例中，我们模拟五个请求（-n 5），将 RAID 级别指定为 0（-L 0），并将随机请求的范围限制为 RAID 的前二十个块（-R 20）。

结果是对 RAID 的前二十个块进行了一系列的随机读取。然后，对于每次逻辑读取，模拟器都会要求您猜测访问了哪些底层磁盘/偏移量来为请求服务。

在这种情况下，计算答案很容易：在 RAID-0 中，回想起为请求提供服务的基础磁盘和偏移量是通过模运算来计算的：

```
disk    = address % number_of_disks
offset  = address / number_of_disks
```

因此，对地址为 16 的第一个请求应由磁盘 0（偏移量 4）来服务。

依此类推。您可以照常查看答案（一旦计算出结果！），方法是使用方便的"-c"标志来计算结果。

```
prompt> ./raid.py -R 20 -n 5 -L 0 -c

...
LOGICAL READ from addr:16 size:4096
read  [disk 0, offset 4]

LOGICAL READ from addr:8 size:4096
read  [disk 0, offset 2]

LOGICAL READ from addr:10 size:4096
read  [disk 2, offset 2]

LOGICAL READ from addr:15 size:4096
read  [disk 3, offset 3]

LOGICAL READ from addr:9 size:4096
read  [disk 1, offset 2]
```

为了让我们玩得开心，也可以使用"-r"标志来反向解决此问题。

以这种方式运行模拟器将向您显示物理磁盘的读写，并要求您反向计算必须向 RAID 发出哪些逻辑请求：

```
prompt> ./raid.py -R 20 -n 5 -L 0 -r

...
LOGICAL OPERATION is ?
read  [disk 0, offset 4]
```

```
LOGICAL OPERATION is ?
  read  [disk 0, offset 2]

LOGICAL OPERATION is ?
  read  [disk 2, offset 2]

LOGICAL OPERATION is ?
  read  [disk 3, offset 3]

LOGICAL OPERATION is ?
  read  [disk 1, offset 2]
```

您可以再次使用 `-c` 查看答案。为了获得更多的变体，可以使用不同的随机种子 (`-s`)。

通过检查不同的 RAID 级别，甚至可以提供更多的多样性。在模拟器中，支持 RAID-0（块条带化），RAID-1（镜像），RAID-4（块条带化和一个奇偶校验磁盘）和 RAID-5（带奇偶校验的块条带化）。

在下一个示例中，我们显示如何在镜像模式下运行模拟器。我们直接显示答案来节省空间：

```
prompt> ./raid.py -R 20 -n 5 -L 1 -c

...
LOGICAL READ from addr:16 size:4096
  read  [disk 0, offset 8]

LOGICAL READ from addr:8 size:4096
  read  [disk 0, offset 4]

LOGICAL READ from addr:10 size:4096
  read  [disk 1, offset 5]

LOGICAL READ from addr:15 size:4096
  read  [disk 3, offset 7]

LOGICAL READ from addr:9 size:4096
  read  [disk 2, offset 4]
```

您可能会注意到有关这个示例的一些信息。首先，镜像的 RAID-1 采取条带化布局（有人可能将其称为 RAID-01），

其中逻辑块 0 映射到磁盘 0 和 1 的第 0 块，逻辑块 1 映射到磁盘 2 和磁盘 3 的第 0 块。依此类推（在这个示例中有四个磁盘）。

其次，从镜像 RAID 系统读取单个块时，RAID 可以选择读取两个块中的哪个。

我们还可以使用 `-w` 标志探索写操作的方式（而不是仅读取），该标志指定工作负载的“写比例”，即写请求的比例。

默认情况下，它设置为零，因此到目前为止的示例为 100% 读取。让我们看看引入一些写操作后镜像 RAID 会发生什么：

```
prompt> ./raid.py -R 20 -n 5 -L 1 -w 100 -c

...
LOGICAL WRITE to  addr:16 size:4096
  write [disk 0, offset 8]      write [disk 1, offset 8]

LOGICAL WRITE to  addr:8 size:4096
```

```

write [disk 0, offset 4]    write [disk 1, offset 4]

LOGICAL WRITE to  addr:10 size:4096
write [disk 0, offset 5]    write [disk 1, offset 5]

LOGICAL WRITE to  addr:15 size:4096
write [disk 2, offset 7]    write [disk 3, offset 7]

LOGICAL WRITE to  addr:9 size:4096
write [disk 2, offset 4]    write [disk 3, offset 4]

```

对于写入操作，RAID 必须更新两个磁盘，而不是仅对单个物理磁盘进行写操作，因此会发出两次写入操作的请求。

您可能会猜到，RAID-4 和 RAID-5 会发生更多有趣的事情。我们将在问题中探索这些事情。

Problem1

问题描述

使用模拟器执行一些基本的 RAID 映射测试。运行不同的级别（0、1、4、5），看看你是否可以找出一组请求的映射。对于 RAID-5，看看你是否可以找出左对称（left-symmetric）和左不对称（left-asymmetric）布局之间的区别。使用一些不同的随机种子，产生不同于上面的问题。

问题分析

不同的 RAID 级别，数据在磁盘上的布局有所不同。我们当然可以通过公式来计算出逻辑块映射到哪个磁盘的哪个位置，但是为了直观，在此我们画出不同级别 RAID 的数据分布表，在表中根据逻辑块查找磁盘和偏移量。

问题解答

RAID 0

```

python2 raid.py -n 3 -L 0 -R 16 -s 0
...
LOGICAL READ from addr:13 size:4096
Physical reads/writes?

LOGICAL READ from addr:6 size:4096
Physical reads/writes?

LOGICAL READ from addr:8 size:4096
Physical reads/writes?

```

disk0	disk1	disk2	disk3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

disk、offset 均从 0 开始计数，查表不难得出

- LOGICAL READ from addr:13 size:4096
read [disk 1, offset 3]
- LOGICAL READ from addr:6 size:4096
read [disk 2, offset 1]
- LOGICAL READ from addr:8 size:4096
read [disk 0, offset 2]

RAID 1

```
python2 raid.py -n 3 -L 1 -R 16 -s 1
...
LOGICAL READ from addr:2 size:4096
Physical reads/writes?

LOGICAL READ from addr:12 size:4096
Physical reads/writes?

LOGICAL READ from addr:7 size:4096
Physical reads/writes?
```

disk0	disk1	disk2	disk3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7
8	8	9	9
10	10	11	11
12	12	13	13
14	14	15	15

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:2 size:4096
read [disk 1, offset 1]
- LOGICAL READ from addr:12 size:4096
read [disk 0, offset 6]
- LOGICAL READ from addr:7 size:4096
read [disk 3, offset 3]

RAID 4

```
python2 raid.py -n 3 -L 4 -R 16 -s 2
...
LOGICAL READ from addr:15 size:4096
Physical reads/writes?

LOGICAL READ from addr:0 size:4096
Physical reads/writes?

LOGICAL READ from addr:13 size:4096
Physical reads/writes?
```

disk0	disk1	disk2	disk3
0	1	2	P
3	4	5	P
6	7	8	P
9	10	11	P
12	13	14	P
15	16	17	P

disk、offset均从0开始计数，disk3用作奇偶校验磁盘，查表不难得出

- LOGICAL READ from addr:15 size:4096
read [disk 0, offset 5]
- LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
- LOGICAL READ from addr:13 size:4096
read [disk 1, offset 4]

对于 RAID 5 我们采用顺序读的方式找出左对称（left- symmetric）和左不对称（left-asymmetric）布局之间的区别。

RAID 5 LS

```
python2 raid.py -n 18 -L 5 -5 LS -W seq -s 3 -c
...
LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
  read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
  read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
  read [disk 3, offset 1]
LOGICAL READ from addr:4 size:4096
  read [disk 0, offset 1]
LOGICAL READ from addr:5 size:4096
  read [disk 1, offset 1]
LOGICAL READ from addr:6 size:4096
  read [disk 2, offset 2]
LOGICAL READ from addr:7 size:4096
  read [disk 3, offset 2]
LOGICAL READ from addr:8 size:4096
  read [disk 0, offset 2]
LOGICAL READ from addr:9 size:4096
  read [disk 1, offset 3]
LOGICAL READ from addr:10 size:4096
  read [disk 2, offset 3]
LOGICAL READ from addr:11 size:4096
  read [disk 3, offset 3]
LOGICAL READ from addr:12 size:4096
  read [disk 0, offset 4]
LOGICAL READ from addr:13 size:4096
  read [disk 1, offset 4]
LOGICAL READ from addr:14 size:4096
  read [disk 2, offset 4]
LOGICAL READ from addr:15 size:4096
  read [disk 3, offset 5]
LOGICAL READ from addr:16 size:4096
```

```
read [disk 0, offset 5]
LOGICAL READ from addr:17 size:4096
read [disk 1, offset 5]
```

易知左对称布局如下:

disk0	disk1	disk2	disk3
0	1	2	P
4	5	P	3
8	P	6	7
P	9	10	11
12	13	14	P
16	17	P	15

RAID 5 LA

```
python2 raid.py -n 18 -L 5 -5 LA -W seq -s 3 -c
...
LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
read [disk 0, offset 1]
LOGICAL READ from addr:4 size:4096
read [disk 1, offset 1]
LOGICAL READ from addr:5 size:4096
read [disk 3, offset 1]
LOGICAL READ from addr:6 size:4096
read [disk 0, offset 2]
LOGICAL READ from addr:7 size:4096
read [disk 2, offset 2]
LOGICAL READ from addr:8 size:4096
read [disk 3, offset 2]
LOGICAL READ from addr:9 size:4096
read [disk 1, offset 3]
LOGICAL READ from addr:10 size:4096
read [disk 2, offset 3]
LOGICAL READ from addr:11 size:4096
read [disk 3, offset 3]
LOGICAL READ from addr:12 size:4096
read [disk 0, offset 4]
LOGICAL READ from addr:13 size:4096
read [disk 1, offset 4]
LOGICAL READ from addr:14 size:4096
read [disk 2, offset 4]
LOGICAL READ from addr:15 size:4096
read [disk 0, offset 5]
LOGICAL READ from addr:16 size:4096
read [disk 1, offset 5]
LOGICAL READ from addr:17 size:4096
read [disk 3, offset 5]
```

易知左不对称布局如下:

disk0	disk1	disk2	disk3
0	1	2	P
3	4	P	5
6	P	7	8
P	9	10	11
12	13	14	P
15	16	P	17

答案验证

```
python2 raid.py -n 3 -L 0 -R 16 -s 0 -c
...
LOGICAL READ from addr:13 size:4096
  read [disk 1, offset 3]

LOGICAL READ from addr:6 size:4096
  read [disk 2, offset 1]

LOGICAL READ from addr:8 size:4096
  read [disk 0, offset 2]
```

```
python2 raid.py -n 3 -L 1 -R 16 -s 1 -c
...
LOGICAL READ from addr:2 size:4096
  read [disk 1, offset 1]

LOGICAL READ from addr:12 size:4096
  read [disk 0, offset 6]

LOGICAL READ from addr:7 size:4096
  read [disk 3, offset 3]
```

```
python2 raid.py -n 3 -L 4 -R 16 -s 2 -c
...
LOGICAL READ from addr:15 size:4096
  read [disk 0, offset 5]
LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]
LOGICAL READ from addr:13 size:4096
  read [disk 1, offset 4]
```

经验证，答案正确。

Problem2

问题描述

与第一个问题一样，但这次使用 -C 来改变大块的大小。大块的大小如何改变映射？

问题分析

不同的 RAID 级别，数据在磁盘上的布局有所不同。画出不同级别 RAID、不同大块大小的数据分布表，在表中根据逻辑块查找磁盘和偏移量。

问题解答

RAID 0

```
python2 raid.py -n 3 -L 0 -R 16 -s 0 -C 8K
...
LOGICAL READ from addr:13 size:4096
Physical reads/writes?

LOGICAL READ from addr:6 size:4096
Physical reads/writes?

LOGICAL READ from addr:8 size:4096
Physical reads/writes?
```

disk0	disk1	disk2	disk3
0	2	4	6
1	3	5	7
8	10	12	14
9	11	13	15

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:13 size:4096
read [disk 2, offset 3]
- LOGICAL READ from addr:6 size:4096
read [disk 3, offset 0]
- LOGICAL READ from addr:8 size:4096
read [disk 0, offset 2]

RAID 1

```
python2 raid.py -n 3 -L 1 -R 16 -s 1 -C 8K
...
LOGICAL READ from addr:2 size:4096
Physical reads/writes?

LOGICAL READ from addr:12 size:4096
Physical reads/writes?

LOGICAL READ from addr:7 size:4096
Physical reads/writes?
```

disk0	disk1	disk2	disk3
0	0	2	2
1	1	3	3
4	4	6	6
5	5	7	7
8	8	10	10
9	9	11	11
12	12	14	14
13	13	15	15

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:2 size:4096
read [disk 2, offset 0]
- LOGICAL READ from addr:12 size:4096
read [disk 0, offset 6]
- LOGICAL READ from addr:7 size:4096
read [disk 3, offset 3]

RAID 4

```
python2 raid.py -n 3 -L 4 -R 16 -s 2 -C 8k
...
LOGICAL READ from addr:15 size:4096
Physical reads/writes?

LOGICAL READ from addr:0 size:4096
Physical reads/writes?

LOGICAL READ from addr:13 size:4096
Physical reads/writes?
```

disk0	disk1	disk2	disk3
0	2	4	P
1	3	5	P
6	8	10	P
7	9	11	P
12	14	16	P
13	15	17	P

disk、offset均从0开始计数，disk3用作奇偶校验磁盘，查表不难得出

- LOGICAL READ from addr:15 size:4096
read [disk 1, offset 5]
- LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
- LOGICAL READ from addr:13 size:4096
read [disk 0, offset 5]

RAID 5 LS

```
python2 raid.py -n 3 -L 5 -5 LS -R 16 -s 3 -C 8K
...
LOGICAL READ from addr:3 size:4096
Physical reads/writes?

LOGICAL READ from addr:5 size:4096
Physical reads/writes?

LOGICAL READ from addr:10 size:4096
Physical reads/writes?
```

disk0	disk1	disk2	disk3
0	2	4	P
1	3	5	P
8	10	P	6
9	11	P	7
16	P	12	14
17	P	13	15

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:3 size:4096
read [disk 1, offset 1]
- LOGICAL READ from addr:5 size:4096
read [disk 2, offset 1]
- LOGICAL READ from addr:10 size:4096
read [disk 1, offset 2]

RAID 5 LA

```
python2 raid.py -n 3 -L 5 -5 LA -R 16 -s 3 -C 8K
...
LOGICAL READ from addr:3 size:4096
Physical reads/writes?

LOGICAL READ from addr:5 size:4096
Physical reads/writes?

LOGICAL READ from addr:10 size:4096
Physical reads/writes?
```

disk0	disk1	disk2	disk3
0	2	4	P
1	3	5	P
6	8	P	10
7	9	P	11
12	P	14	16
13	P	15	17

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:3 size:4096
read [disk 1, offset 1]
- LOGICAL READ from addr:5 size:4096
read [disk 2, offset 1]

- LOGICAL READ from addr:10 size:4096
read [disk 3, offset 2]

答案验证

```
python2 raid.py -n 3 -L 0 -R 16 -s 0 -C 8K -c
...
LOGICAL READ from addr:13 size:4096
read [disk 2, offset 3]

LOGICAL READ from addr:6 size:4096
read [disk 3, offset 0]

LOGICAL READ from addr:8 size:4096
read [disk 0, offset 2]
```

```
python2 raid.py -n 3 -L 1 -R 16 -s 1 -C 8K -c
...
LOGICAL READ from addr:2 size:4096
read [disk 2, offset 0]

LOGICAL READ from addr:12 size:4096
read [disk 0, offset 6]

LOGICAL READ from addr:7 size:4096
read [disk 3, offset 3]
```

```
python2 raid.py -n 3 -L 4 -R 16 -s 2 -C 8K -c
...
LOGICAL READ from addr:15 size:4096
read [disk 1, offset 5]
LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
LOGICAL READ from addr:13 size:4096
read [disk 0, offset 5]
```

```
python2 raid.py -n 3 -L 5 -5 LS -R 16 -s 3 -C 8K -c
...
LOGICAL READ from addr:3 size:4096
read [disk 1, offset 1]
LOGICAL READ from addr:5 size:4096
read [disk 2, offset 1]
LOGICAL READ from addr:10 size:4096
read [disk 1, offset 2]
```

```
python2 raid.py -n 3 -L 5 -5 LA -R 16 -s 3 -C 8K -c
...
LOGICAL READ from addr:3 size:4096
read [disk 1, offset 1]
LOGICAL READ from addr:5 size:4096
read [disk 2, offset 1]
LOGICAL READ from addr:10 size:4096
read [disk 3, offset 2]
```

经验证，答案正确。

Problem3

问题描述

执行上述测试，但使用 `-r` 标志来反转每个问题的性质。

问题分析

不同的 RAID 级别，数据在磁盘上的布局有所不同。画出不同级别 RAID 的数据分布表，根据指定磁盘和偏移量在表中查找逻辑块。

问题解答

RAID 0

```
python2 raid.py -n 3 -L 0 -R 16 -s 0 -r
...
LOGICAL OPERATION is ?
  read  [disk 1, offset 3]

LOGICAL OPERATION is ?
  read  [disk 2, offset 1]

LOGICAL OPERATION is ?
  read  [disk 0, offset 2]
```

disk0	disk1	disk2	disk3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

disk、offset 均从 0 开始计数，查表不难得出

- LOGICAL READ from addr:13 size:4096
read [disk 1, offset 3]
- LOGICAL READ from addr:6 size:4096
read [disk 2, offset 1]
- LOGICAL READ from addr:8 size:4096
read [disk 0, offset 2]

RAID 1

```
python2 raid.py -n 3 -L 1 -R 16 -s 1 -r
...
LOGICAL OPERATION is ?
  read  [disk 1, offset 1]

LOGICAL OPERATION is ?
  read  [disk 0, offset 6]

LOGICAL OPERATION is ?
  read  [disk 3, offset 3]
```

disk0	disk1	disk2	disk3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7
8	8	9	9
10	10	11	11
12	12	13	13
14	14	15	15

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:2 size:4096
read [disk 1, offset 1]
- LOGICAL READ from addr:12 size:4096
read [disk 0, offset 6]
- LOGICAL READ from addr:7 size:4096
read [disk 3, offset 3]

RAID 4

```
python2 raid.py -n 3 -L 4 -R 16 -s 2 -r
...
LOGICAL OPERATION is ?
  read  [disk 0, offset 5]
LOGICAL OPERATION is ?
  read  [disk 0, offset 0]
LOGICAL OPERATION is ?
  read  [disk 1, offset 4]
```

disk0	disk1	disk2	disk3
0	1	2	P
3	4	5	P
6	7	8	P
9	10	11	P
12	13	14	P
15	16	17	P

disk、offset均从0开始计数，disk3用作奇偶校验磁盘，查表不难得出

- LOGICAL READ from addr:15 size:4096
read [disk 0, offset 5]
- LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]

- LOGICAL READ from addr:13 size:4096
read [disk 1, offset 4]

RAID 5 LS

```
python2 raid.py -n 3 -L 5 -5 LS -R 16 -s 3 -r
...
LOGICAL OPERATION is ?
  read [disk 3, offset 1]
LOGICAL OPERATION is ?
  read [disk 1, offset 1]
LOGICAL OPERATION is ?
  read [disk 2, offset 3]
```

disk0	disk1	disk2	disk3
0	1	2	P
4	5	P	3
8	P	6	7
P	9	10	11
12	13	14	P
16	17	P	15

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:3 size:4096
read [disk 3, offset 1]
- LOGICAL READ from addr:5 size:4096
read [disk 1, offset 1]
- LOGICAL READ from addr:10 size:4096
read [disk 2, offset 3]

RAID 5 LA

```
python2 raid.py -n 3 -L 5 -5 LA -R 16 -s 3 -r
...
LOGICAL OPERATION is ?
  read [disk 0, offset 1]
LOGICAL OPERATION is ?
  read [disk 3, offset 1]
LOGICAL OPERATION is ?
  read [disk 2, offset 3]
```

disk0	disk1	disk2	disk3
0	1	2	P
3	4	P	5
6	P	7	8
P	9	10	11
12	13	14	P
15	16	P	17

disk、offset均从0开始计数，查表不难得出

- LOGICAL READ from addr:3 size:4096
read [disk 0, offset 1]

- LOGICAL READ from addr:5 size:4096
read [disk 3, offset 1]
- LOGICAL READ from addr:10 size:4096
read [disk 2, offset 3]

答案验证

```
python2 raid.py -n 3 -L 0 -R 16 -s 0 -r -c
...
LOGICAL READ from addr:13 size:4096
read [disk 1, offset 3]

LOGICAL READ from addr:6 size:4096
read [disk 2, offset 1]

LOGICAL READ from addr:8 size:4096
read [disk 0, offset 2]
```

```
python2 raid.py -n 3 -L 1 -R 16 -s 1 -r -c
...
LOGICAL READ from addr:2 size:4096
read [disk 1, offset 1]

LOGICAL READ from addr:12 size:4096
read [disk 0, offset 6]

LOGICAL READ from addr:7 size:4096
read [disk 3, offset 3]
```

```
python2 raid.py -n 3 -L 4 -R 16 -s 2 -r -c
...
LOGICAL READ from addr:15 size:4096
read [disk 0, offset 5]
LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
LOGICAL READ from addr:13 size:4096
read [disk 1, offset 4]
```

```
python2 raid.py -n 3 -L 5 -5 LS -R 16 -s 3 -r -c
...
LOGICAL READ from addr:3 size:4096
read [disk 3, offset 1]
LOGICAL READ from addr:5 size:4096
read [disk 1, offset 1]
LOGICAL READ from addr:10 size:4096
read [disk 2, offset 3]
```



```
python2 raid.py -n 3 -L 5 -S 5 LA -R 16 -s 3 -r -c
...
LOGICAL READ from addr:3 size:4096
  read  [disk 0, offset 1]
LOGICAL READ from addr:5 size:4096
  read  [disk 3, offset 1]
LOGICAL READ from addr:10 size:4096
  read  [disk 2, offset 3]
```

经验证，答案正确。

Problem4

问题描述

现在使用反转标志，但用-S 标志增加每个请求的大小。尝试指定 8 KB、12 KB 和 16 KB 的大小，同时改变 RAID 级别。当请求的大小增加时，底层 I/O 模式会发生什么？请务必在顺序工作负载上尝试此操作（-W sequential）。对于什么请求大小，RAID-4 和 RAID-5 的 I/O 效率更高？

问题解答

当请求块大小超过磁盘块大小时，一个请求需要读写多个磁盘。

对于 RAID-4 和 RAID-5，请求块大小为 12KB 时（numDisks=4），效率更高，因为可以同时利用多个磁盘，相当于全条带写入（见书 P335 页）

```
python2 raid.py -n 12 -L 4 -W seq -S 8K -r -c
python2 raid.py -n 12 -L 4 -W seq -S 12K -r -c
python2 raid.py -n 12 -L 4 -W seq -S 16K -r -c

python2 raid.py -n 12 -L 5 -W seq -S 8K -r -c
python2 raid.py -n 12 -L 5 -W seq -S 12K -r -c
python2 raid.py -n 12 -L 5 -W seq -S 16K -r -c
```

Problem5

问题描述

使用模拟器的定时模式（-t）来估计 100 次随机读取到 RAID 的性能，同时改变 RAID 级别，使用 4 个磁盘。

问题解答

RAID 0

```
python2 raid.py -L 0 -t -n 100 -c -D 4

...
disk:0 busy: 100.00 I/Os: 28 (sequential:0 nearly:1 random:27)
disk:1 busy: 93.91 I/Os: 29 (sequential:0 nearly:6 random:23)
disk:2 busy: 87.92 I/Os: 24 (sequential:0 nearly:0 random:24)
disk:3 busy: 65.94 I/Os: 19 (sequential:0 nearly:1 random:18)

STAT totalTime 275.7
```

RAID 1

```
python2 raid.py -L 1 -t -n 100 -c -D 4

...
disk:0 busy: 100.00 I/Os: 28 (sequential:0 nearly:1 random:27)
disk:1 busy: 86.98 I/Os: 24 (sequential:0 nearly:0 random:24)
disk:2 busy: 97.52 I/Os: 29 (sequential:0 nearly:3 random:26)
disk:3 busy: 65.23 I/Os: 19 (sequential:0 nearly:1 random:18)

STAT totalTime 278.7
```

RAID 4

```
python2 raid.py -L 4 -t -n 100 -c -D 4

...
disk:0 busy: 78.48 I/Os: 30 (sequential:0 nearly:0 random:30)
disk:1 busy: 100.00 I/Os: 40 (sequential:0 nearly:3 random:37)
disk:2 busy: 76.46 I/Os: 30 (sequential:0 nearly:2 random:28)
disk:3 busy: 0.00 I/Os: 0 (sequential:0 nearly:0 random:0)

STAT totalTime 386.1
```

RAID 5

```
python2 raid.py -L 5 -t -n 100 -c -D 4

...
disk:0 busy: 100.00 I/Os: 28 (sequential:0 nearly:1 random:27)
disk:1 busy: 95.84 I/Os: 29 (sequential:0 nearly:5 random:24)
disk:2 busy: 87.60 I/Os: 24 (sequential:0 nearly:0 random:24)
disk:3 busy: 65.70 I/Os: 19 (sequential:0 nearly:1 random:18)

STAT totalTime 276.7
```

通过比较可以发现，RAID 0、RAID 1、RAID 5的耗时基本一致，而 RAID 4 耗时较长，约为前三者的 1.38倍。又观察到 RAID 4 的disk3始终处于空闲状态，没有处理读取请求，这是因为disk3是奇偶校验磁盘。

我们知道RAID 0、RAID 1、RAID 5的随机读吞吐量均为 $N \cdot R$ ，而 RAID 4 的随机读吞吐量为 $(N-1) \cdot R$ ，本题中 $N=4$ ， $4/3=1.33$ ，与实验数据基本相符。