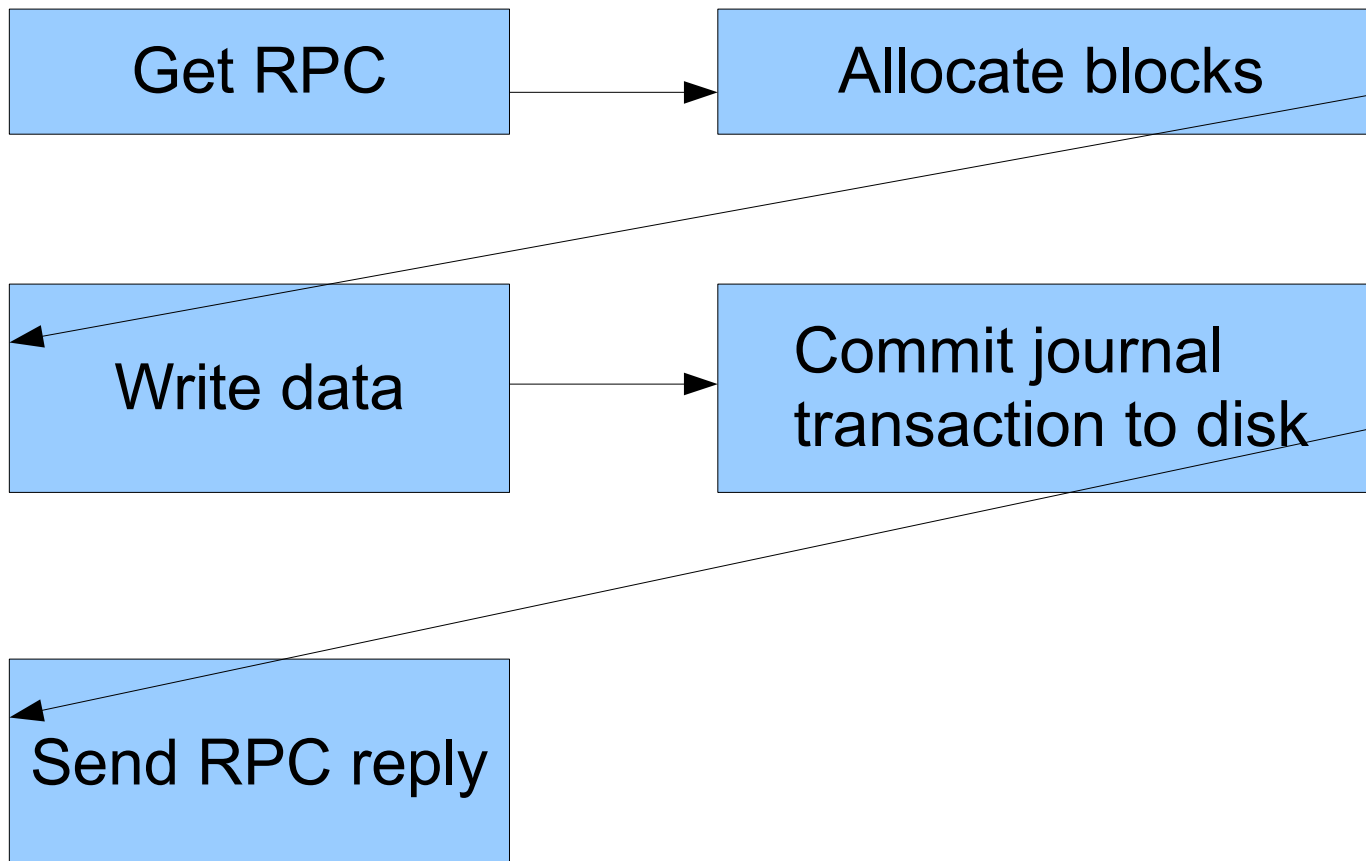
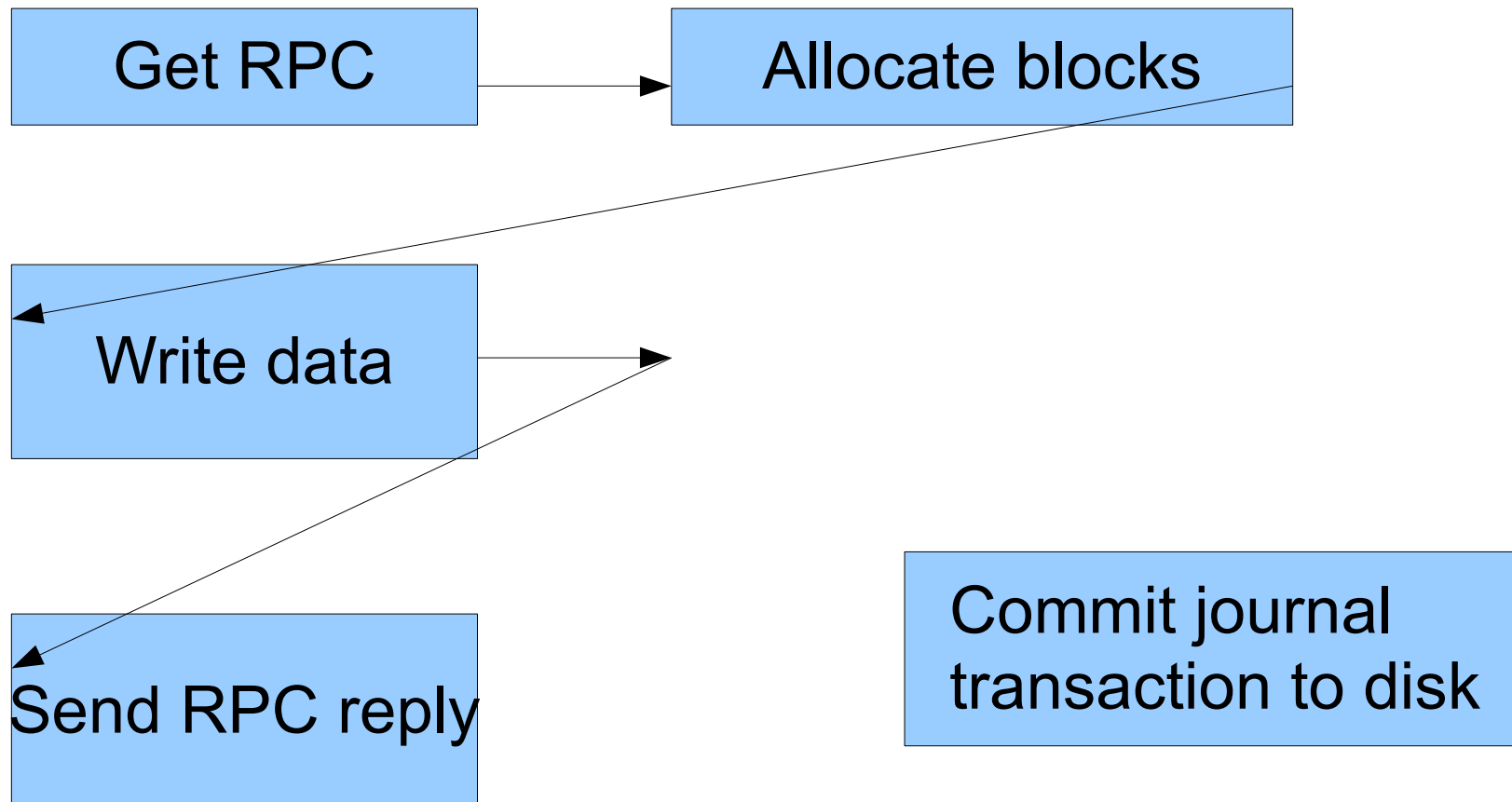


Slow streaming writes



Slow streaming writes (cont'd)

2-4x performance improvement



Bug 16919

Lock callback timeouts



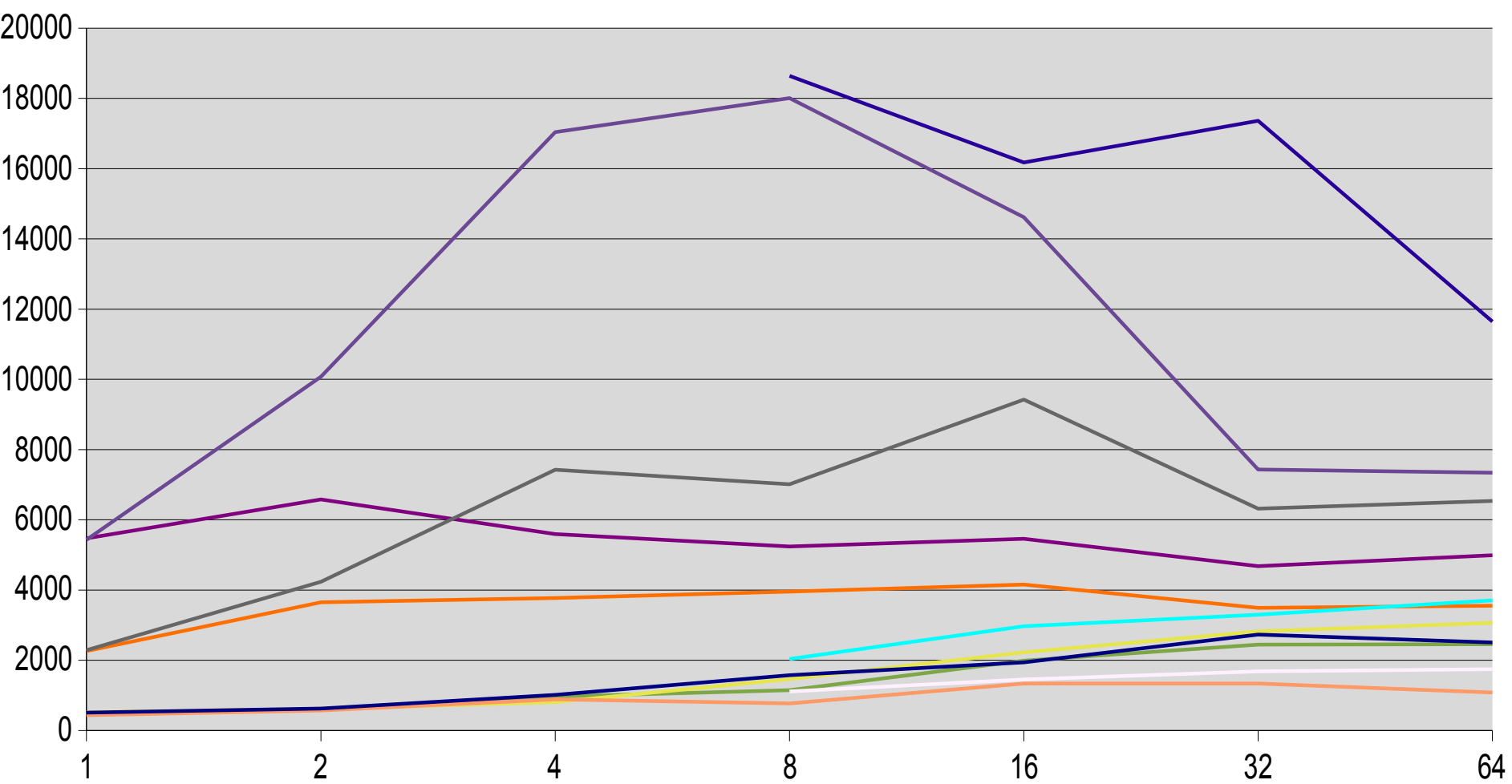
Lock callback timeouts cont'd



Lock callback timeouts

- Two request queues
 - > pre-process handler decides which queue to use and links request to export
 - > Cancel can requeue requests
 - > callback timeout handler can iterate through export requests and see if there is anything.
- On client
 - > Ability to send extra RPC if all current slots are busy
 - > Sorting OAPs in priority based on lock cancellation status

Create performance



```

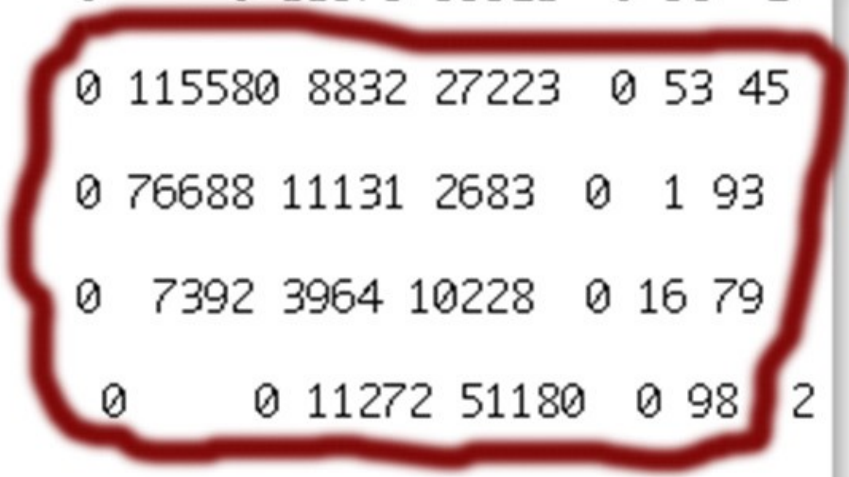
green@inuyasha:~ — bash — 80x24
R 939732 1 5000 1 0 31 16139 2709 2716
R 939733 0 5000 1 0 31 16456 2725 2731
R 939734 1 5000 2 0 32 16662 2745 2751
R 939735 2 5000 2 0 31 16655 2755 2761
R 939736 2 5000 4 0 31 16906 2770 2776
R 939737 4 5000 1 0 31 17099 2784 2790
R 939738 0 5000 1 0 31 17253 2796 2802
R 939739 1 5000 1 0 32 17375 2805 2811
R 939740 0 5000 1 0 31 17542 2807 2813
R 939741 1 5000 2 0 32 17710 2832 2838
R 939742 2 5000 2 0 34 17781 2843 2849
R 939743 2 5000 4 0 34 17957 2859 2865
R 939744 3 5000 0 1 33 18154 2876 2882
R 939745 0 5000 1 0 31 18269 2897 2903
R 939746 1 5000 1 0 34 18327 2895 2902
R 939747 1 5000 1 1 34 18534 2911 2917
R 939748 0 5000 2 0 34 18761 2924 2930
R 939749 2 5000 2 0 34 18868 2940 2946
R 939750 2 5000 5 0 15 9977 700 702
R 939751 0 1675 0 2 208 22184 25468 25519
R 939752 0 1803 7 2 667 22210 25488 25539
C 939750 24 692 8 0
R 939753 8 1848 13 1 796 21928 25499 25550
greenmacpro:~ green$

```

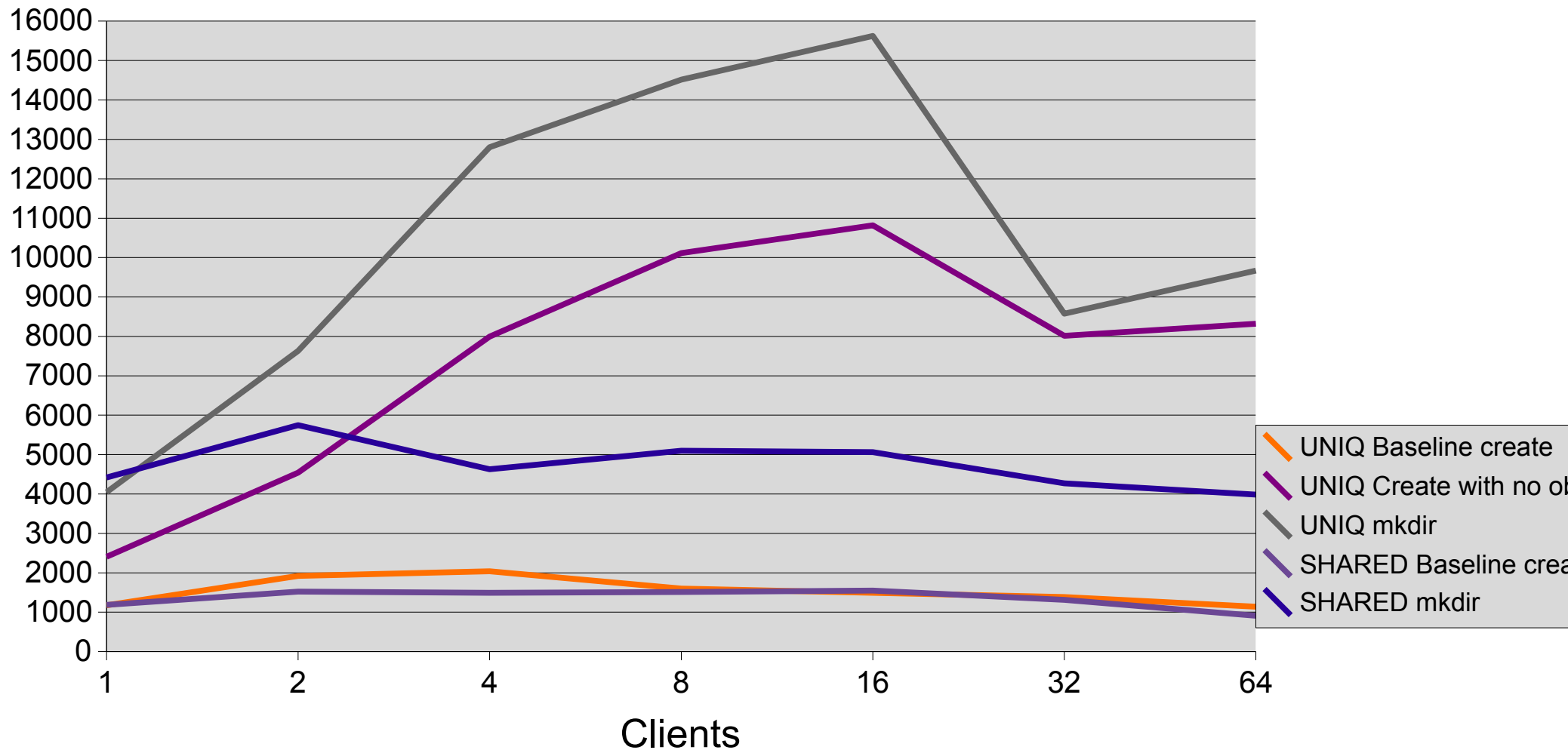
```

green@inuyasha:~ — bash — 80x24
86 0 0 28197792 2824180 731760 0 0 0 0 11373 53521 0 98 2
0 0
0 1 0 28188368 2824796 731700 0 0 0 115580 8832 27223 0 53 45
2 0
0 1 0 28188384 2824796 732552 0 0 0 76688 11131 2683 0 1 93
5 0
79 0 0 28187268 2825028 732328 0 0 0 7392 3964 10228 0 16 79
5 0
103 0 0 28169660 2826496 731708 0 0 0 0 0 11272 51180 0 98 2
0 0
0 1 0 28153912 2827376 731284 0 0 0 94928 6850 23282 0 45 52
3 0
0 2 0 28158500 2827376 732548 0 0 0 74756 9429 3969 0 2 91 8
0
75 0 0 28146472 2828460 731632 0 0 0 228788 16509 46618 0 78 21
2 0
44 0 0 28125708 2829608 731632 0 0 0 0 0 14218 63166 0 94 7
0 0
35 0 0 28108464 2829608 732292 0 0 0 0 0 16887 61379 0 95 5
0 0
90 1 0 28090496 2829608 732560 0 0 0 89788 16050 46542 0 98 2
0 0
79 0 0 28074568 2829608 732560 0 0 0 79996 17013 46773 0 97 3
0 0

```



Jaguar PF create performance



CPU bound, now what?

- Oprofile
 - > (hw counters based, if possible)
 - > Collect callgraph! (important to better understand the heavy paths)

A little bit on client cache

- Client cache currently is very limited
 - > 32M per OSC by default
 - > Should not be this limited
 - > `/proc/fs/lustre/osc/*/max_dirty_mb`
 - > Current possible max is $\frac{1}{4}$ of system RAM
- Possible downsides
 - > Interference with application-originated latency-critical RPC traffic
 - > Only works when there is enough free RAM. (if there is not, just use `O_DIRECT` io anyway).