Tornado Case Study

Our Case Study, News Service Provider (NSP) had an original setup of 4 servers, each carried all groups, and were as follows:

Name	Machine	Disk	Capacity
East	Sparc 4500, 6 processors, 6G RAM	3T	max 2000 users (40% broadband)
West	Sparc 4500, 6 processors, 6G RAM	3T	max 2000 users (40% broadband)
ISP	Sparc 420, 4 processors, 4G RAM	2T	max 800 users (20% broadband)
Web	Sparc 450, 4 processors, 2G RAM	1T	max 500 users (20% broadband)
Text	Sparc 420, 4 processors, 1G RAM	720G	max 800 users(10% broadband)

The NSP made the business decision to leverage cheap bandwidth, and gambled on a new software package to increase their load-carrying capacity without significant capital expenditure. In particular, the disk arrays (MTI controllers, state-of-the-art fibre channel) were very expensive.

Along with an increase in load carrying capacity, the NSP wanted to increase their retention, again without needing to buy more disk. It seemed obvious that there must be some way to aggregate the disks as a 'back end' since adding a new server meant adding another 3T, and increasing retention would require increasing all the disk capacities, a very expensive prospect.

To make a long story short, Tornado was written and deployed, without adding a single byte of disk to the above setup, and converting them all to back-ends, is now serving double and triple its old retention, to five times more customers, by only adding cheap Tornado front ends.

Their current setup is:

East-> Multimedia-only Backend (20 days retention)
West-> Binary-only Backend (8 days retention)
ISP->pictures only (2 months retention)
Web-> Binary load offset (only 1 day retention, offloading the busiest "most recent" load)
Text-> Text back end (10 months retention)

The above servers supply articles to 10 front ends, each serving over 1000 customers (10,000 total simultaneous connections) with a demonstrated capacity for 500 more each (15,000 customers, reading an aggregate of all back end content)

All front-end servers are identically configured- Sparc 420, 4G RAM, 44-disk JBOD (9G each, total capacity 400G, but is only barely $1/10^{th}$ used –

```
/dev/dsk/c2t0d0s6    8705501  640289 7978157     8%    /spool1
/dev/dsk/c2t1d0s6    8705501  558833 8059613     7%    /spool2
/dev/dsk/c0t0d0s4    6555869  270835 6219476     5%    /news
/dev/dsk/c2t3d0s6    8705501  349041 8269405     5%    /spool4
/dev/dsk/c2t2d0s6    8705501  549537 8068909     7%    /spool3
/dev/dsk/c2t6d0s6    8705501  106021 8512425     2%    /spool7
/dev/dsk/c2t5d0s6    8705501  157513 8460933     2%    /spool6
/dev/dsk/c2t4d0s6    8705501  238113 8380333     3%    /spool5
/dev/dsk/c0t0d0s5    2508555  394148 2064236    17%    /opt
/dev/dsk/c2t7d0s6    8705501  151801 8466645     2%    /spool8
/dev/dsk/c2t9d0s6    8705501  384813 8233633     5%    /spool10
/dev/dsk/c2t8d0s6    8705501  140749 8477697     2%    /spool9
/dev/dsk/c2t10d0s6   8705501  444933 8173513     6%    /spool11
/dev/dsk/c2t18d0s6   8705501  760465 7857981     9%    /spool14
/dev/dsk/c2t19d0s6   8705501  732889 7885557     9%    /spool15
/dev/dsk/c2t20d0s6   8705501  867937 7750509    11%    /spool16
/dev/dsk/c2t16d0s6   8705501  381221 8237225     5%    /spool12
/dev/dsk/c2t17d0s6   8705501  749889 7868557     9%    /spool13
```

```
/dev/dsk/c2t21d0s6   8705501  857445 7761001    10%    /spool17
/dev/dsk/c2t22d0s6   8705501  858029 7760417    10%    /spool18
/dev/dsk/c2t23d0s6   8705501 1273661 7344785    15%    /spool19
/dev/dsk/c2t26d0s6   8705501  313021 8305425     4%    /spool22
/dev/dsk/c2t24d0s6   8705501  875609 7742837    11%    /spool20
/dev/dsk/c3t32d0s6   8705501  397785 8220661     5%    /spool23
/dev/dsk/c2t25d0s6   8705501  864225 7754221    11%    /spool21
/dev/dsk/c3t33d0s6   8705501  416493 8201953     5%    /spool24
/dev/dsk/c3t35d0s6   8705501 1666585 6951861    20%    /spool26
/dev/dsk/c3t34d0s6   8705501 1428945 7189501    17%    /spool25
/dev/dsk/c3t36d0s6   8705501 1667841 6950605    20%    /spool27
/dev/dsk/c3t38d0s6   8705501 1489713 7128733    18%    /spool29
/dev/dsk/c3t37d0s6   8705501 1646809 6971637    20%    /spool28
/dev/dsk/c3t40d0s6   8705501 1790905 6827541    21%    /spool31
/dev/dsk/c3t39d0s6   8443727 2490357 5868933    30%    /spool30
/dev/dsk/c3t41d0s6   8705501  875385 7743061    11%    /spool32
/dev/dsk/c3t42d0s6   8705501  803717 7814729    10%    /spool33
/dev/dsk/c3t48d0s6   8705501 1209501 7408945    15%    /spool34
/dev/dsk/c3t50d0s6   8705501  401809 8216637     5%    /spool36
/dev/dsk/c3t52d0s6   8705501  647621 7970825     8%    /spool38
/dev/dsk/c3t51d0s6   8705501  244385 8374061     3%    /spool37
/dev/dsk/c3t54d0s6   8705501  768521 7849925     9%    /spool40
/dev/dsk/c3t53d0s6   8705501  646245 7972201     8%    /spool39
/dev/dsk/c3t55d0s6   8705501  465633 8152813     6%    /spool41
/dev/dsk/c3t56d0s6   8705501  366661 8251785     5%    /spool42
/dev/dsk/c3t57d0s6   8705501  427857 8190589     5%    /spool43
/dev/dsk/c3t58d0s6   8705501  351445 8267001     5%    /spool44
/dev/dsk/c0t0d0s6    3530374    1533 3493538     1%    /export/home
/dev/dsk/c3t49d0s6   8705501  888017 7730429    11%    /spool35
```

The footprint of a full tornado is around 60G. This is accomplished by storing the headers more efficiently than twister, and doing away with overview caches in favor of a completely retooled xover system. This system has been shown to be 500% faster than the old system, under full load. The typical 420 serving 1000 customers has a profile as follows:

```
load averages:  4.82,  5.13,  5.74                                  23:14:28
37 processes:  36 sleeping, 1 on cpu
CPU states: 25.1% idle,  7.1% user, 56.8% kernel, 10.9% iowait,  0.0% swap
Memory: 4096M real, 1980M free, 1935M swap in use, 2051M swap free

   PID USERNAME THRD PRI NICE  SIZE   RES STATE    TIME    CPU COMMAND
 27108 news     1228  59    0 1398M 1393M sleep  184.8H 29.92% tornadod
```

This front end was serving 919 users at the time this snapshot was taken.

When the NSP offered their new service, at an incredibly low price, it attracted many hard-core users, the broadband percentage has risen to approximately 70% (based on connection speeds) In this case, this server was outputting 310megabits/sec, or an average of 337kilobits/sec per connection, a bit low, but this was taken during the day. At night, the usage spikes as power users set broadband connections up for mass download and stop using them for other things.

The Tornado software has shown its ability to failover and scale multiple times, when the NSP had a switch fail, 4 of the servers were knocked out. The resulting 5 servers (1 is independent and used for outsourcing) spiked, but remained responsive, and customer impact was minimal.

In summary- the most expensive part of any news server, the massive disk capacity, can have its utilization (in terms of customers served) increased by a factor of 5 by the addition of cheap, 60G front end Tornado machines. At the same time, customers see a dramatic increase in availability and retention.