Best Practices For Magento Hosting

Chris Wells – Nexcess

Today's Topics

- A note on permissions
- L**AMP** L**AMP** L**AMP**
  - PHP basics & the PHP handoff
  - Apache vs. Nginx
  - MySQL vs. Percona
- Caching best practices
- Using Varnish with Magento
- Final notes
- Questions etc.

Permissions Matter

- chmod 666 / 777 = “make it work good”
- Fix it (relative to your web root)
  1. Own your files / directories
     - find \-exec chown magento.magento {} \;
  2. Make sure the webserver can read it all
     - find -type f \-exec chmod 644 {} \;
  3. PHP is for your eyes only
     - find -type f -name “*.php” \-exec chmod 600 {} \;
  4. So are config files!
     - chmod 600 app/etc/*.xml

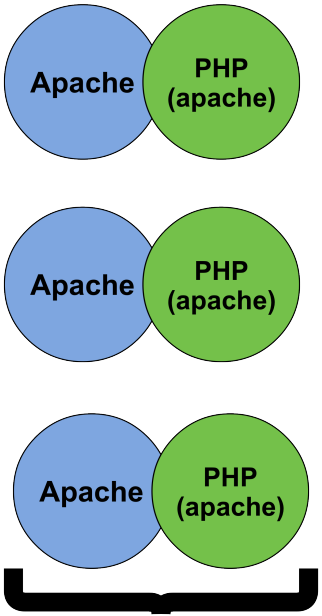
PHP Basics

- Use APC (as an opcode cache only)
  - apc.shm_size = 256M (at least)
  - apc.num_files_hint = 10000 (at least)
  - apc_stat = 0 (for production)
- Bump memory_limit (512M works well)
- Turn OFF open_basedir
  - Leaving it ON kills the realpath cache
- Use a recent version of PHP
  - 5.4.x may be too recent

The PHP Hand-off

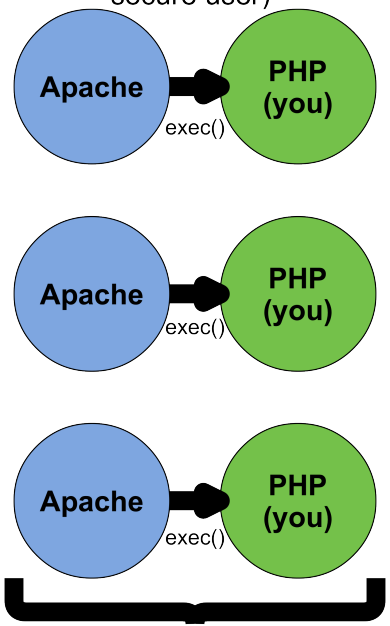
- Apache + mod_php
  - Runs as webserver user
  - Unified mega-process
- Apache + suPHP / phpSuEXEC
  - Runs as you
  - Expensive to create
- Apache/Nginx + PHP-FPM
  - Runs as you
  - Cheap to use (processes are waiting)
  - Scales more efficiently than all of the above

Apache + mod_php
(Big process, web user)



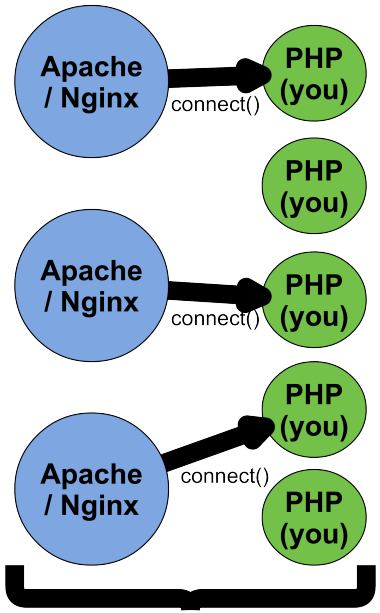
One-to-one server/PHP mapping

Apache + suPHP
(Huge cost / script, secure user)



One-to-one server/PHP mapping

Apache / Nginx + FPM
(Efficient separation, secure user)



Many-to-many server/PHP mapping

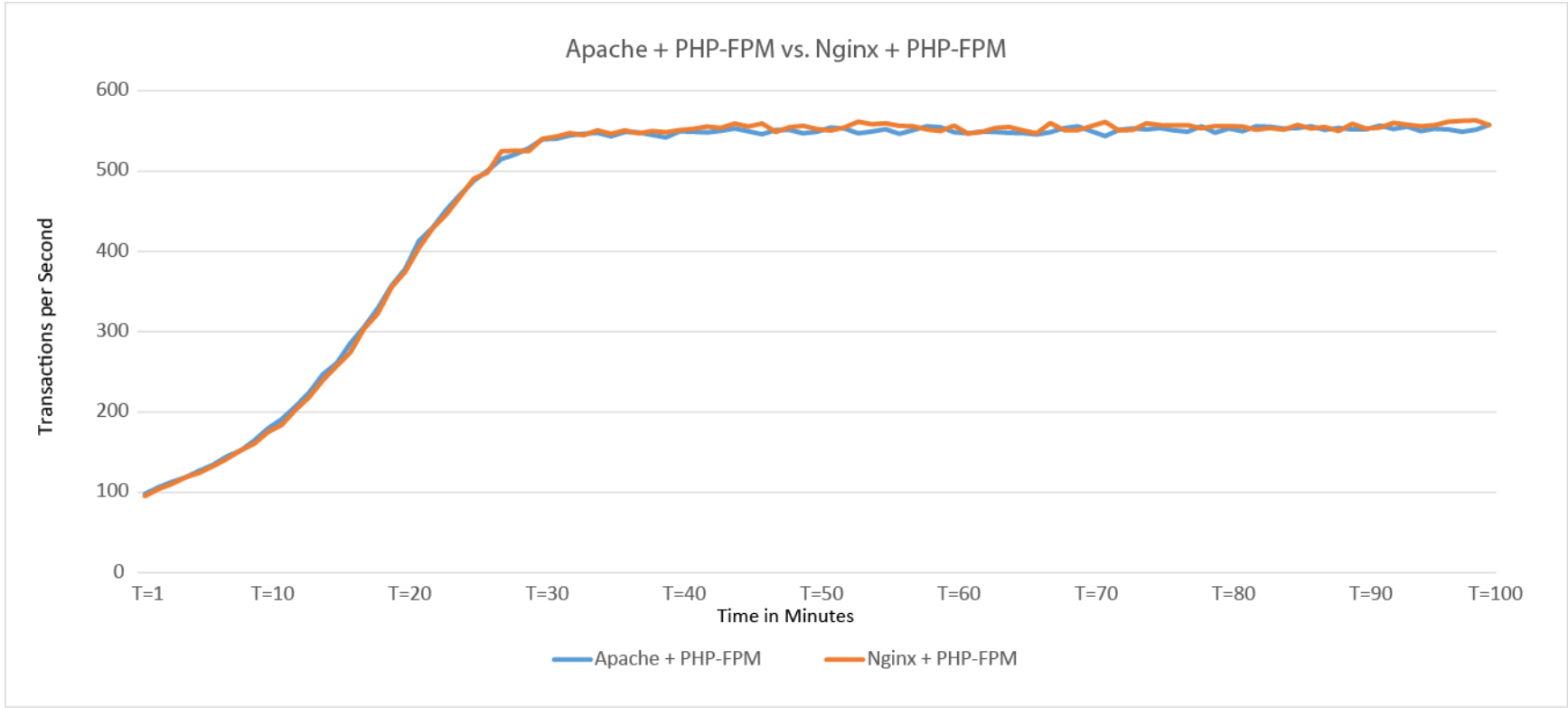
The Webserver Showdown

VS.

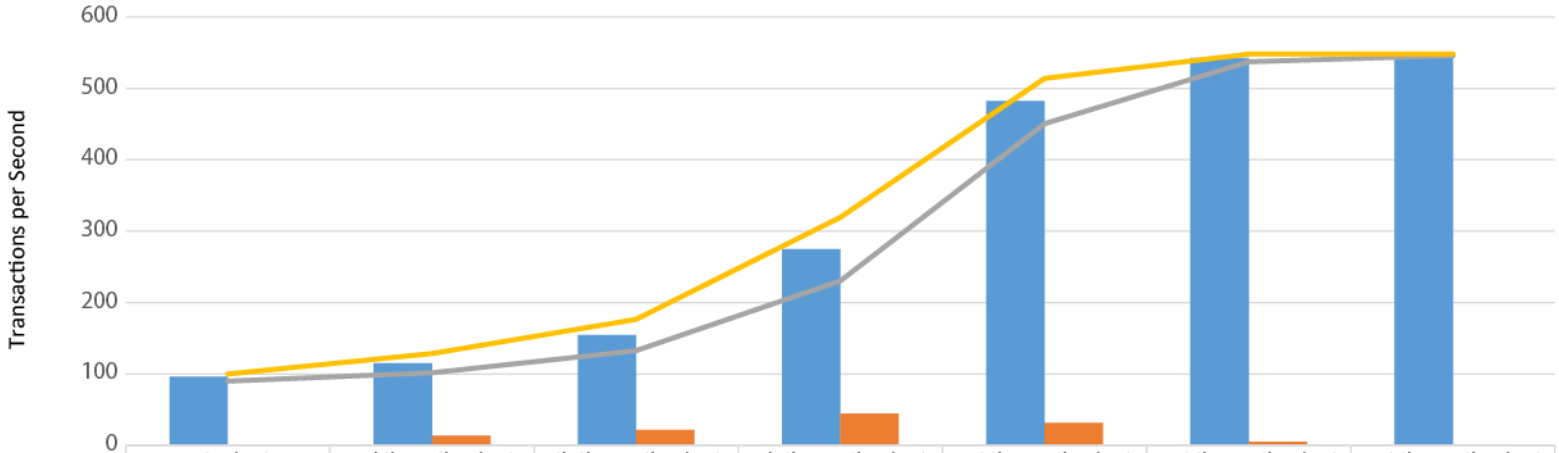
NGINX

So … Apache vs. Nginx?

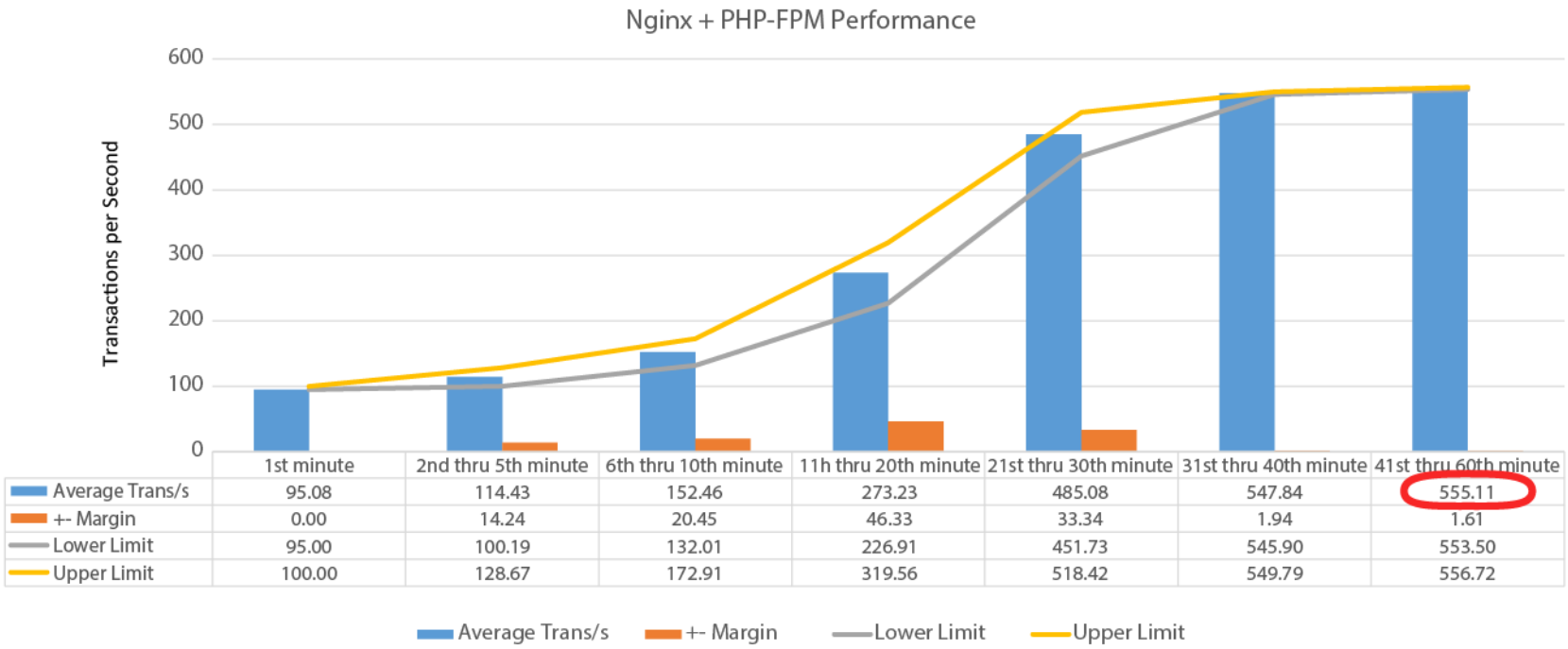
- The answer is …
  - YES!
- Apache comes bloated – remove needless modules!
- Magento supports Apache out-of-the-box
  - Rewrites work as expected
  - Extensions may assume Apache-like features exist
- PHP-FPM levels the performance / scalability field
- Varnish helps as well (we're getting ahead of ourselves)
- Go with what you know!!


Apache + PHP-FPM vs. Nginx + PHP-FPM
600
500
400
300
200
100
0
Transactions per Second
T=1
T=10
T=20
T=30
T=40
T=50
T=60
T=70
T=80
T=90
T=100
Time in Minutes
Apache + PHP-FPM
Nginx + PHP-FPM

Apache + PHP-FPM Performance



	1st minute	2nd thru 5th minute	6th thru 10th minute	11h thru 20th minute	21st thru 30th minute	31st thru 40th minute	41st thru 60th minute
Average Trans/s	96.34	115.12	154.53	274.87	482.03	542.34	546.87
+- Margin	0.00	13.63	21.90	44.48	31.57	5.39	1.15
Lower Limit	90.00	101.49	132.63	230.39	450.46	536.95	545.72
Upper Limit	100.00	128.75	176.43	319.35	513.60	547.72	548.02



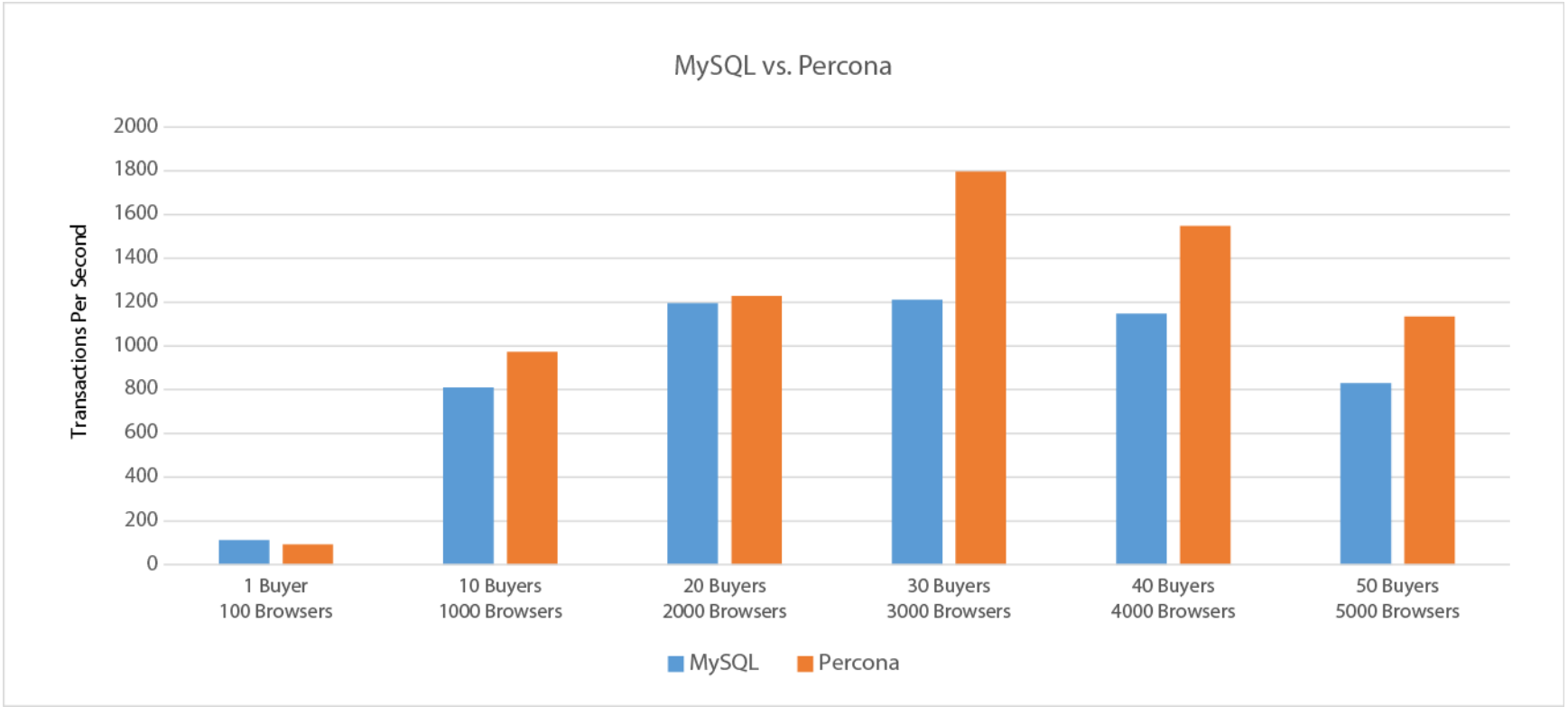
	1st minute	2nd thru 5th minute	6th thru 10th minute	11h thru 20th minute	21st thru 30th minute	31st thru 40th minute	41st thru 60th minute
Average Trans/s	95.08	114.43	152.46	273.23	485.08	547.84	555.11
+- Margin	0.00	14.24	20.45	46.33	33.34	1.94	1.61
Lower Limit	95.00	100.19	132.01	226.91	451.73	545.90	553.50
Upper Limit	100.00	128.67	172.91	319.56	518.42	549.79	556.72

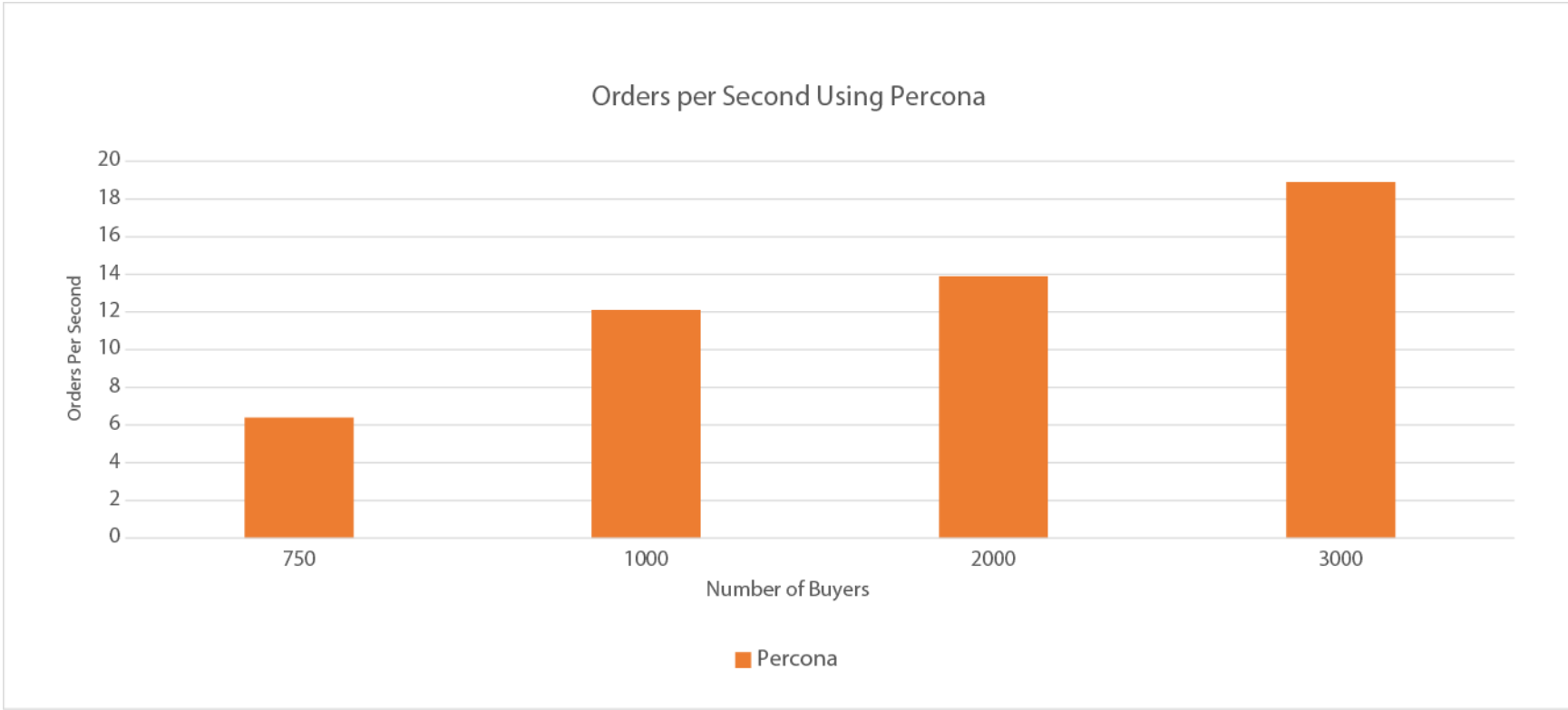
The DB Showdown

PERCONA

VS.

Percona wins ... no need to use fade-ins ... next slide


MySQL vs. Percona
Transactions Per Second
2000
1800
1600
1400
1200
1000
800
600
400
200
0
1 Buyer
100 Browsers
10 Buyers
1000 Browsers
20 Buyers
2000 Browsers
30 Buyers
3000 Browsers
40 Buyers
4000 Browsers
50 Buyers
5000 Browsers
MySQL
Percona


Orders per Second Using Percona
Orders Per Second
20
18
16
14
12
10
8
6
4
2
0
750
1000
2000
3000
Number of Buyers
Percona

Percona's Benefits / Tweaks

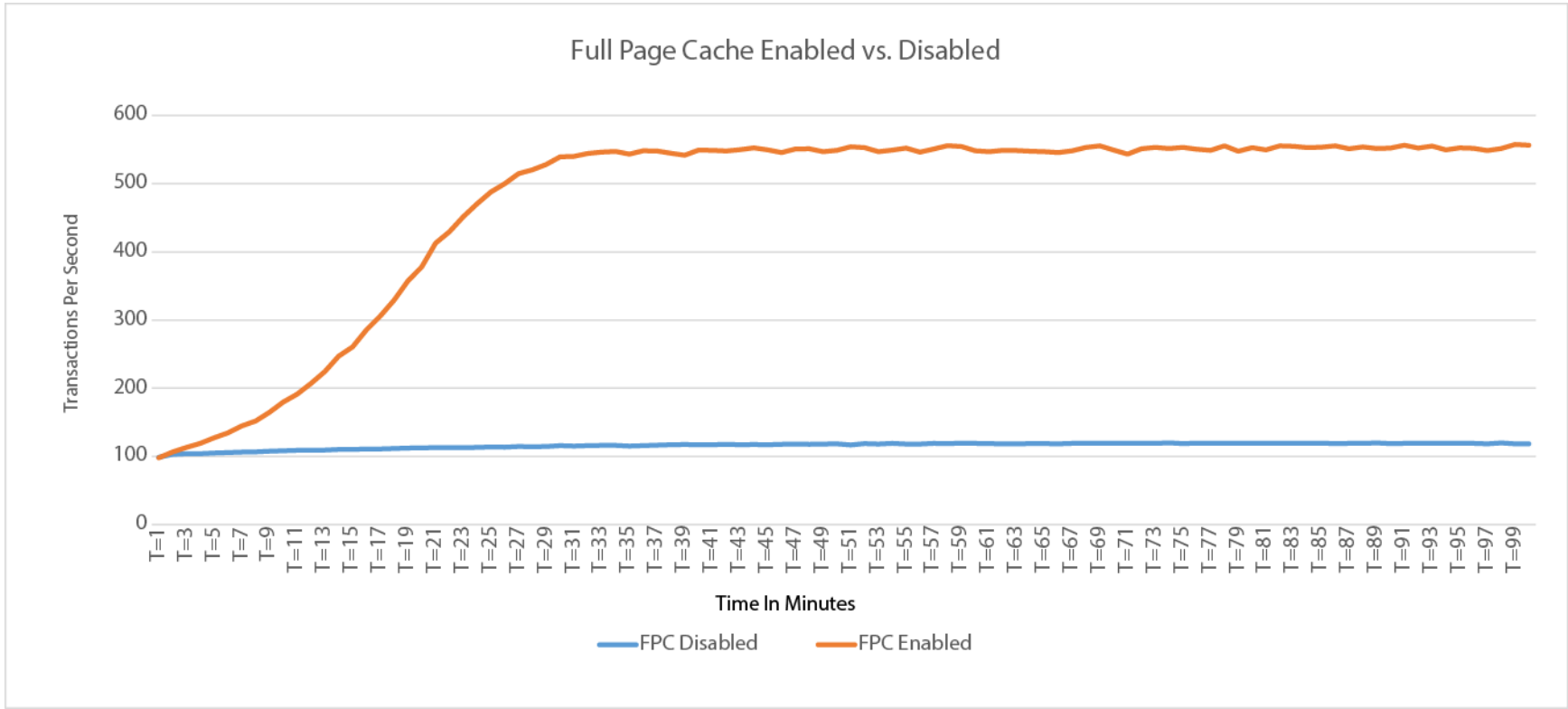
- Percona’s Xtra DB is fast – especially under load
- Percona is a simple replacement
- My.cnf tweaks:
  - innodb_thread_concurrency = 24 (1 – 2x # of cores)
  - innodb_buffer_pool_size = 16G (at least)
  - innodb_flush_log_at_trx_commit = 1
  - innodb_io_capacity = 800
  - innodb_flush_method = O_DIRECT

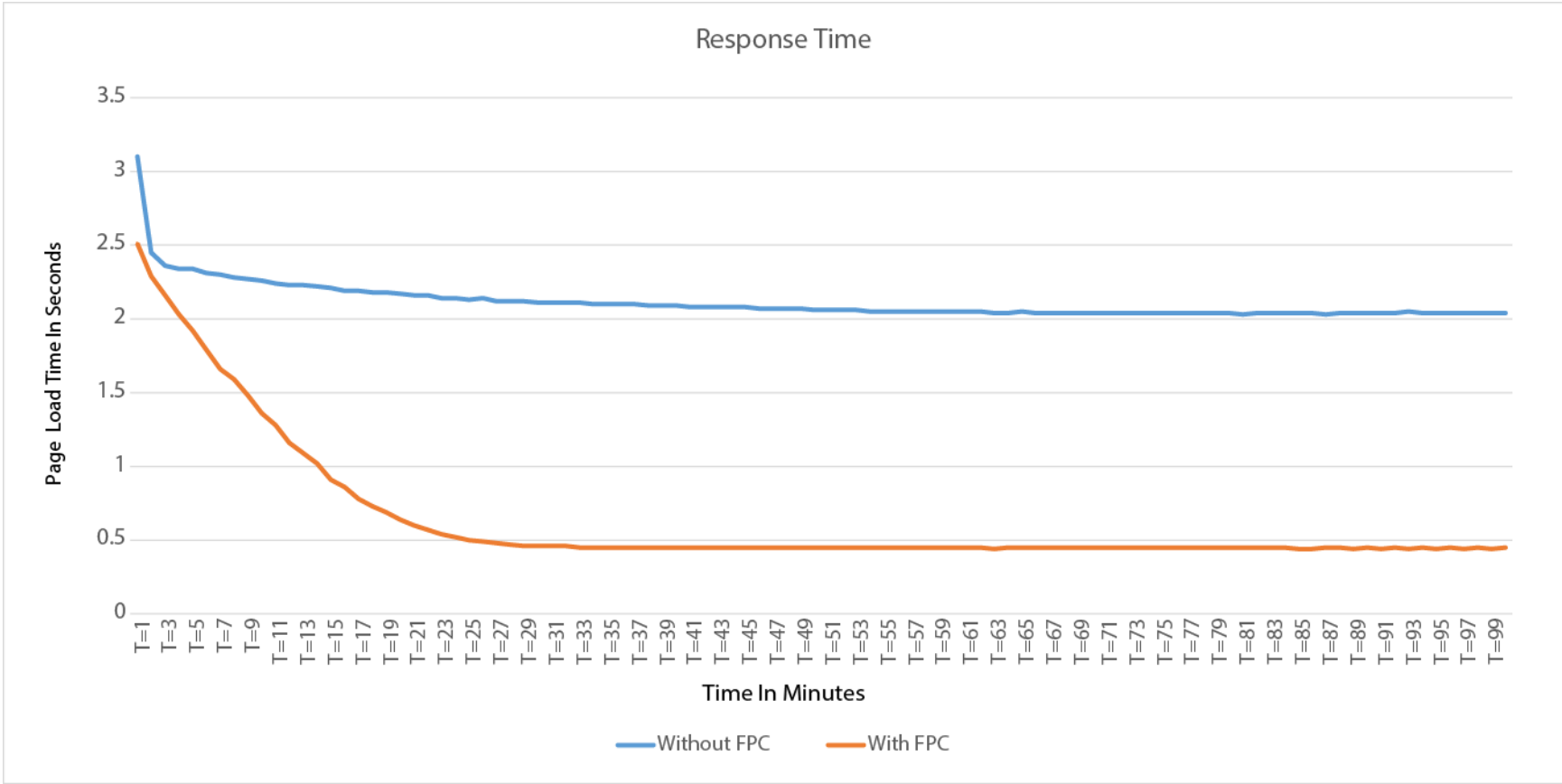
Time to Cache in

- Memcache:
  - PRO: multi-threaded, socket/tcp based
  - CON: no tagging
- Redis
  - PRO: TAGGING, fast, socket/tcp based
  - CON: single threaded
- APC:
  - PRO: provides op-code cache
  - CON: no CLI usage, can't share, no tagging, restart causes flush, we avoid for key/value pair caching

Caching Best Practices – Part 1

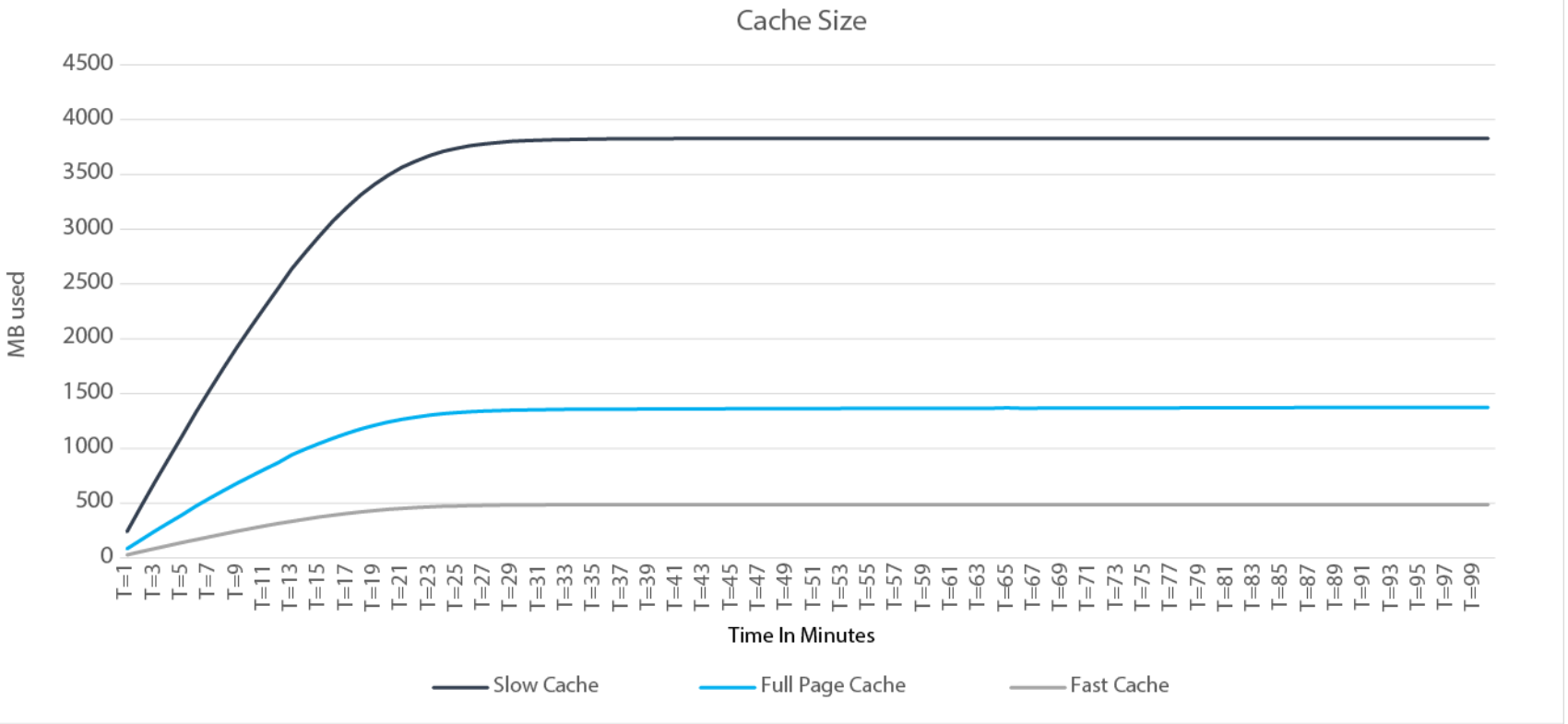
- Using Magento Enterprise?
  - Turn on the Full Page Cache (FPC)
  - Huge throughput gains
  - Huge response time gains
  - Use a dedicated Redis instance!
  - Quick, easy and it works
- Not Using Enterprise?
  - No simple “light -switch” solution
  - But … Varnish is a good option :)

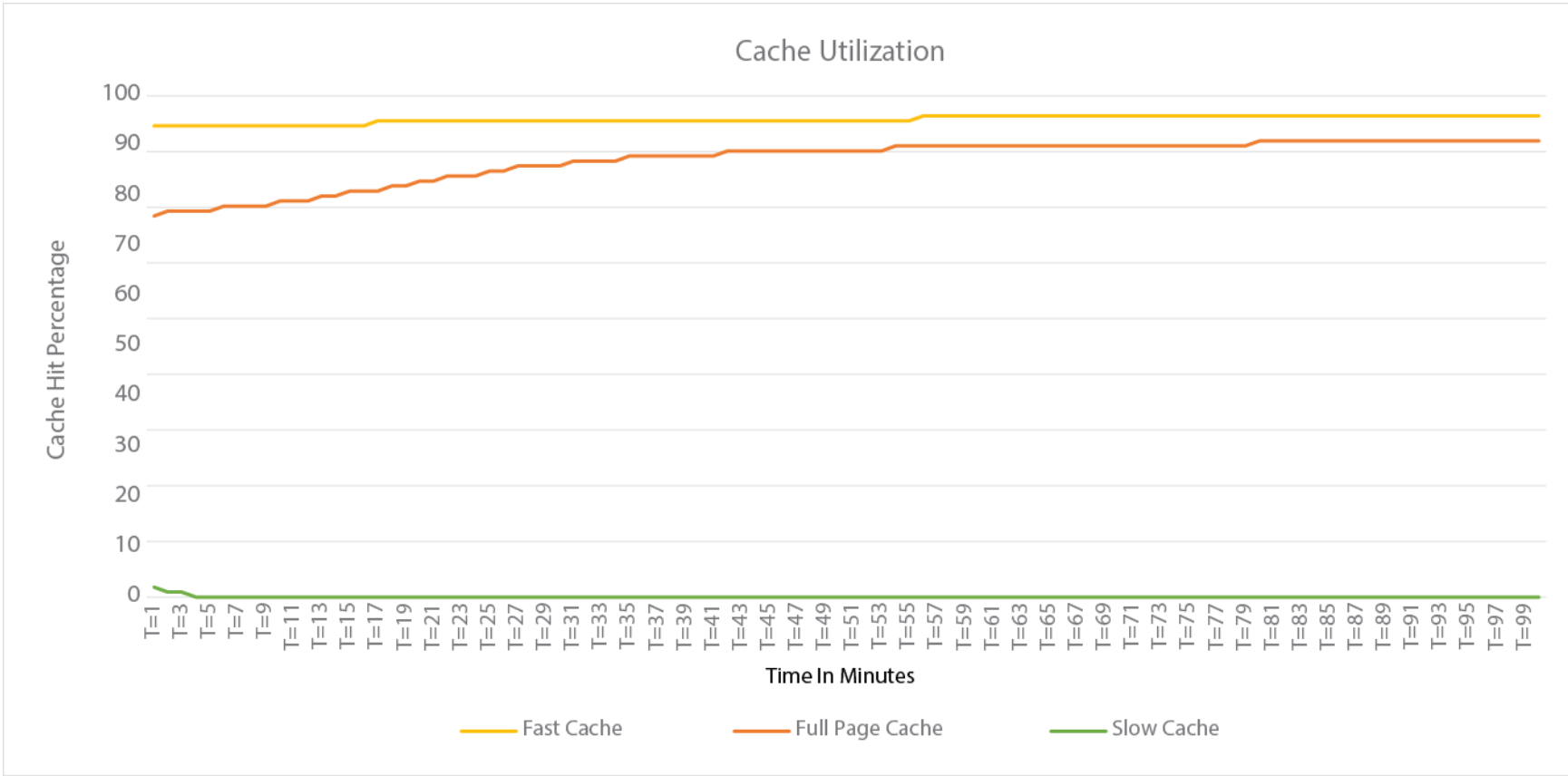

Full Page Cache Enabled vs. Disabled
Transactions Per Second
600
500
400
300
200
100
0
T=1
T=3
T=5
T=7
T=9
T=11
T=13
T=15
T=17
T=19
T=21
T=23
T=25
T=27
T=29
T=31
T=33
T=35
T=37
T=39
T=41
T=43
T=45
T=47
T=49
T=51
T=53
T=55
T=57
T=59
T=61
T=63
T=65
T=67
T=69
T=71
T=73
T=75
T=77
T=79
T=81
T=83
T=85
T=87
T=89
T=91
T=93
T=95
T=97
T=99
Time In Minutes
FPC Disabled
FPC Enabled


Response Time
Page Load Time In Seconds
3.5
3
2.5
2
1.5
1
0.5
0
T=1
T=3
T=5
T=7
T=9
T=11
T=13
T=15
T=17
T=19
T=21
T=23
T=25
T=27
T=29
T=31
T=33
T=35
T=37
T=39
T=41
T=43
T=45
T=47
T=49
T=51
T=53
T=55
T=57
T=59
T=61
T=63
T=65
T=67
T=69
T=71
T=73
T=75
T=77
T=79
T=81
T=83
T=85
T=87
T=89
T=91
T=93
T=95
T=97
T=99
Time In Minutes
Without FPC
With FPC

Caching Best Practices – Part 2

- Setup the Magento 2-level cache
  - Fast cache = memcache
    - Multi-threaded, responds better under heavy load
    - Run multiple Redis if you like
  - Slow cache = Redis
    - Single-threaded
    - Slow cache may not be used
- Setup a dedicated memcache for sessions
- Size the caches correctly! Fast cache should fit all data!

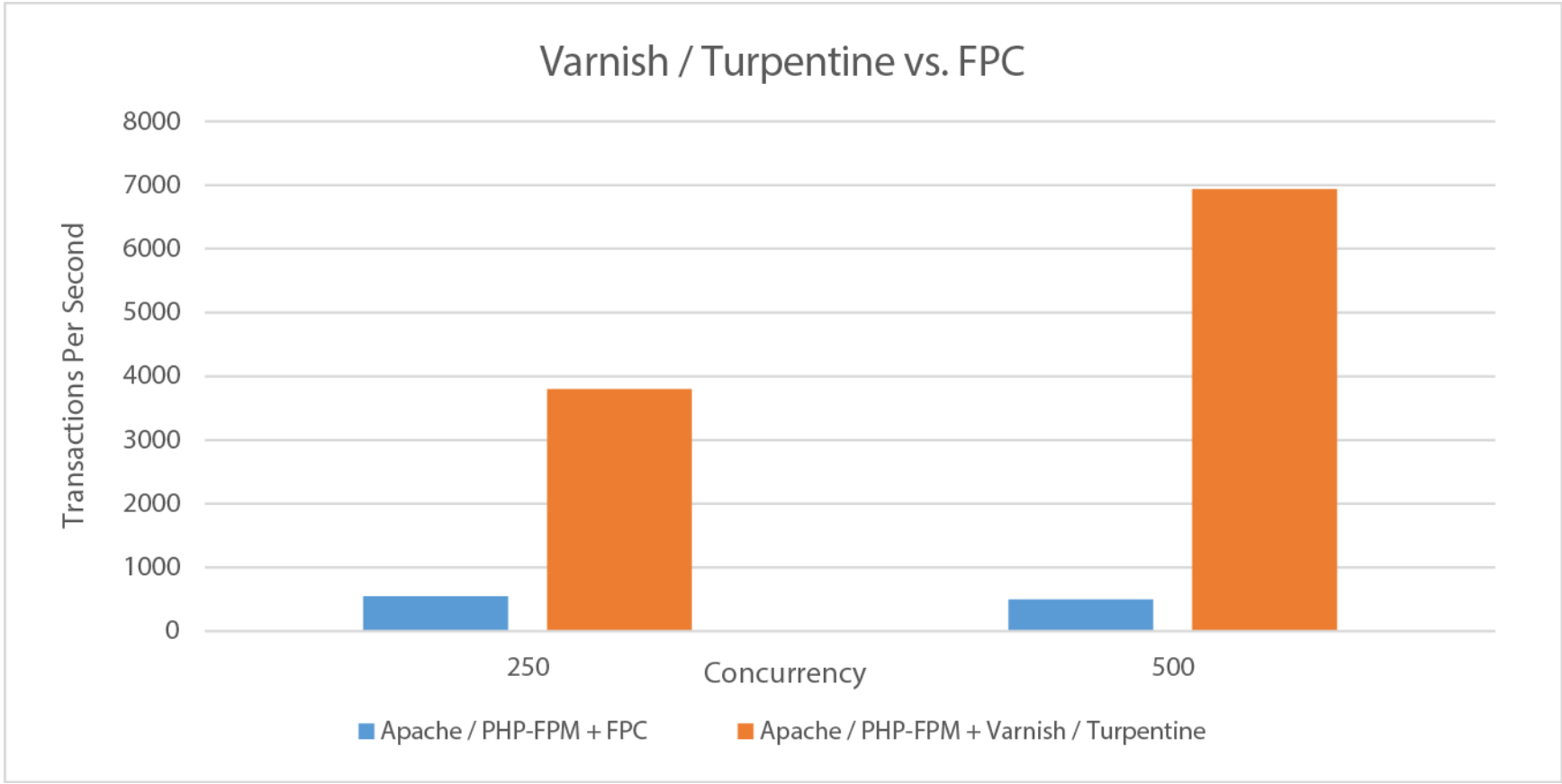

Cache Size
MB used
4500
4000
3500
3000
2500
2000
1500
1000
500
0
T=1
T=3
T=5
T=7
T=9
T=11
T=13
T=15
T=17
T=19
T=21
T=23
T=25
T=27
T=29
T=31
T=33
T=35
T=37
T=39
T=41
T=43
T=45
T=47
T=49
T=51
T=53
T=55
T=57
T=59
T=61
T=63
T=65
T=67
T=69
T=71
T=73
T=75
T=77
T=79
T=81
T=83
T=85
T=87
T=89
T=91
T=93
T=95
T=97
T=99
Time In Minutes
Slow Cache
Full Page Cache
Fast Cache

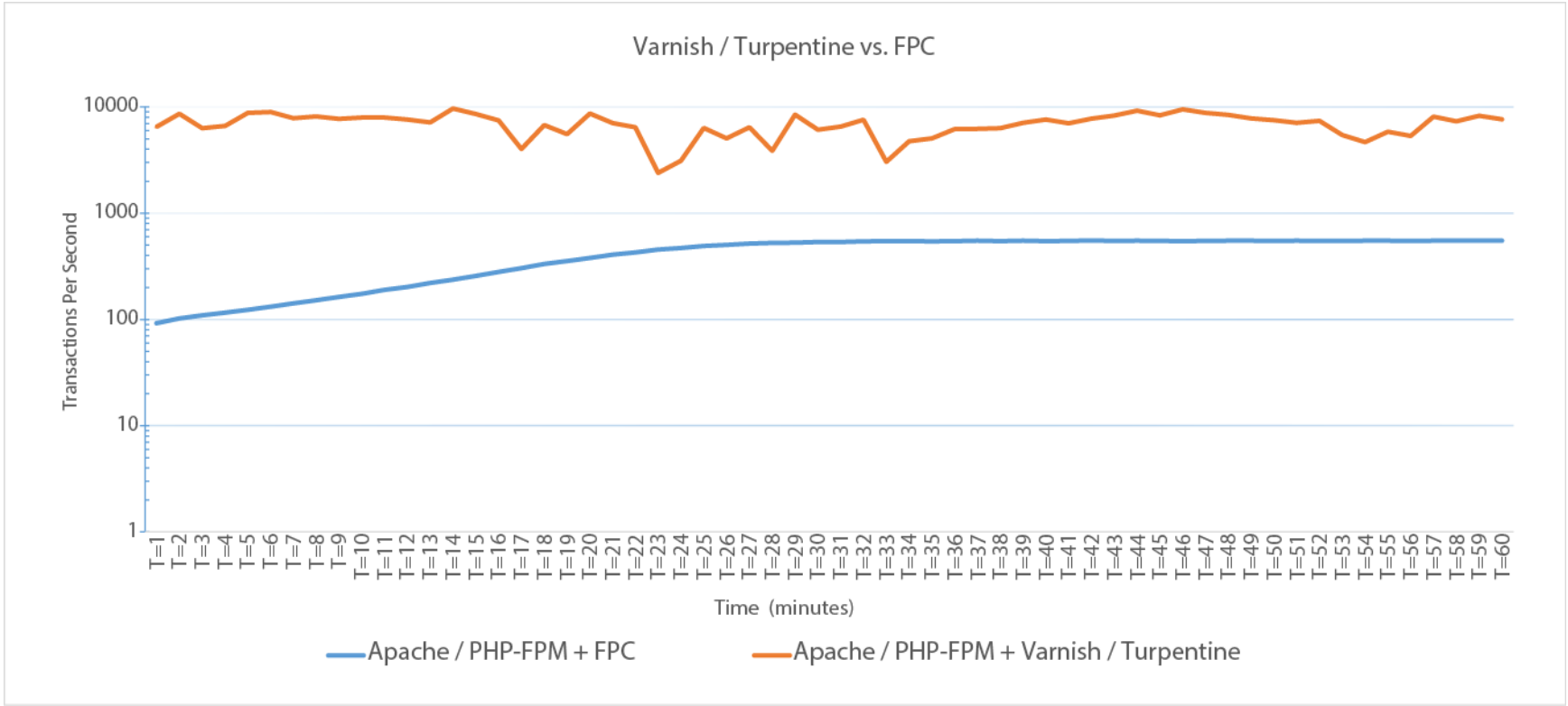

Cache Utilization
Cache Hit Percentage
100
90
80
70
60
50
40
30
20
10
0
T=1
T=3
T=5
T=7
T=9
T=11
T=13
T=15
T=17
T=19
T=21
T=23
T=25
T=27
T=29
T=31
T=33
T=35
T=37
T=39
T=41
T=43
T=45
T=47
T=49
T=51
T=53
T=55
T=57
T=59
T=61
T=63
T=65
T=67
T=69
T=71
T=73
T=75
T=77
T=79
T=81
T=83
T=85
T=87
T=89
T=91
T=93
T=95
T=97
T=99
Time In Minutes
Fast Cache
Full Page Cache
Slow Cache

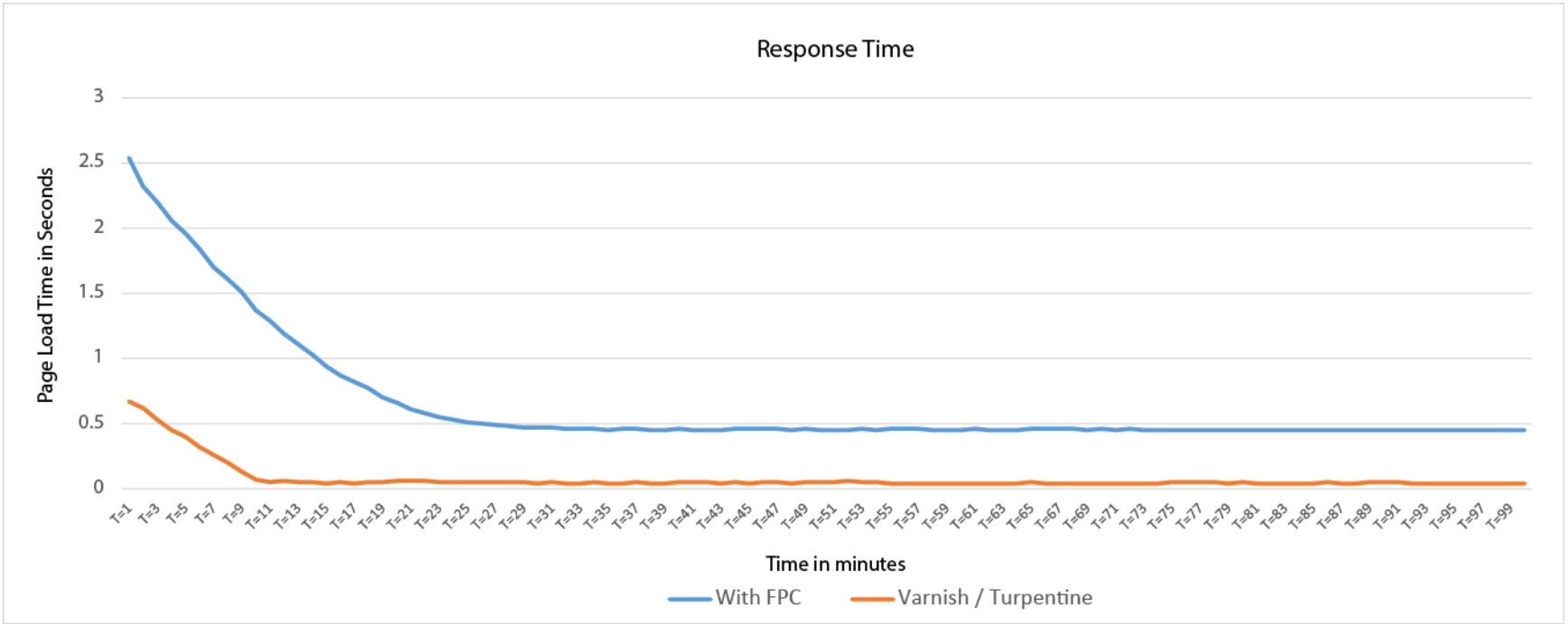
Caching the Caches With Varnish

Caching the Caches With Varnish

- Caches entire pages (or parts of them)
  - Use an extension to integrate with Magento
  - Turpentine is ours
    - Free / open source for all Magento versions
- HUGE performance gains for visitors
- Requires more thought than FPC
- SSL requires further hoop-jumping
- ESI requires yet further hoop-jumping


Varnish / Turpentine vs. FPC
Transactions Per Second
8000
7000
6000
5000
4000
3000
2000
1000
0
250
500
Concurrency
Apache / PHP-FPM + FPC
Apache / PHP-FPM + Varnish / Turpentine


Varnish / Turpentine vs. FPC
Transactions Per Second
10000
1000
100
10
1
T=1 T=2 T=3 T=4 T=5 T=6 T=7 T=8 T=9 T=10 T=11 T=12 T=13 T=14 T=15 T=16 T=17 T=18 T=19 T=20 T=21 T=22 T=23 T=24 T=25 T=26 T=27 T=28 T=29 T=30 T=31 T=32 T=33 T=34 T=35 T=36 T=37 T=38 T=39 T=40 T=41 T=42 T=43 T=44 T=45 T=46 T=47 T=48 T=49 T=50 T=51 T=52 T=53 T=54 T=55 T=56 T=57 T=58 T=59 T=60
Time (minutes)
Apache / PHP-FPM + FPC
Apache / PHP-FPM + Varnish / Turpentine


Response Time
Page Load Time in Seconds
3
2.5
2
1.5
1
0.5
0
T=1 T=3 T=5 T=7 T=9 T=11 T=13 T=15 T=17 T=19 T=21 T=23 T=25 T=27 T=29 T=31 T=33 T=35 T=37 T=39 T=41 T=43 T=45 T=47 T=49 T=51 T=53 T=55 T=57 T=59 T=61 T=63 T=65 T=67 T=69 T=71 T=73 T=75 T=77 T=79 T=81 T=83 T=85 T=87 T=89 T=91 T=93 T=95 T=97 T=99
Time in minutes
With FPC
Varnish / Turpentine

Final Thoughts

- 777 and 666 are both evil
- PHP-FPM is the way to go
- Apache and Nginx can be friends
- Percona
- FPC with Redis back-end
- Two-level caching
  - Fast cache = memcache
  - Slow cache = Redis
- Turpentine / Varnish if you can

More Information:

Turpentine is available at:

http://www.nexcess.net/turpentine or on Magento Connect

Our new performance whitepaper is available at:

http://www.nexcess.net/magento-best-practices-whitepaper

Thank You!

Questions?