Alternative Infrastucture

if we do it, we might as well do it right

Experiments

Twitter clone "retwis": Key Value Store: redis Web Framework: sinatra

Mediawiki: Webserver: Nginx (!= Apache) PHP: php-fpm (!= mod_php)

Retwis-RB

"An example Twitter application using the Redis key-value database" --> http://github.com/danlucraft/retwis-rb

sinatra	redis
require 'rubygems'	require 'rubygems'
require 'sinatra'	require 'redis'
get '/hi' do	r = Redis.new
"Hello World!"	puts "set foo to bar"
end	r['foo'] = 'bar'

Redis Benchmark

root@keyvalue:~# redis-benchmark -q

SET: 105273.69 requests per second GET: 107526.88 requests per second INCR: 95238.10 requests per second LPUSH: 121987.80 requests per second LPOP: 108728.26 requests per second PING: 133386.66 requests per second

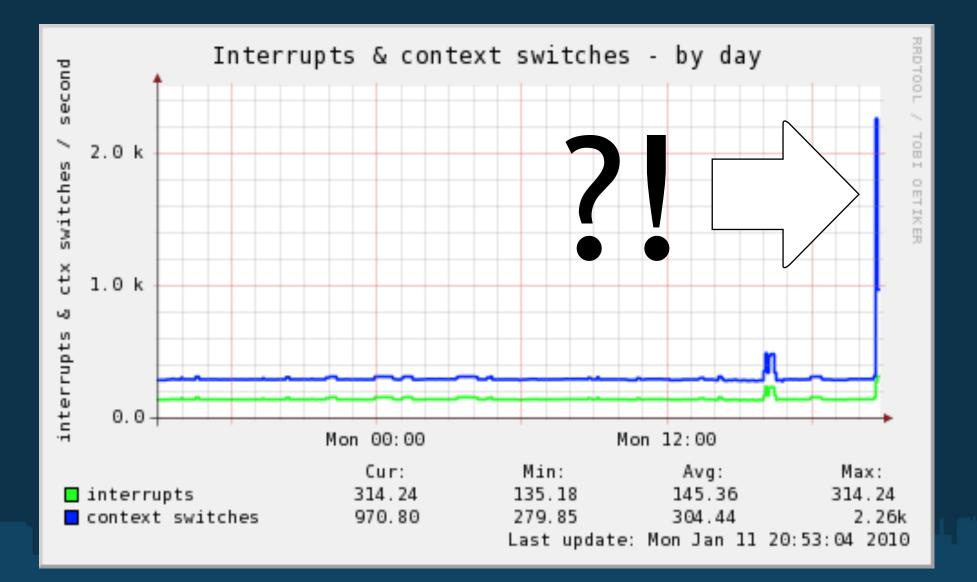
testing @ HdM

CPU: 100% httperf 57% redis-server 40% ruby1.9.1

Durchsatz: 6 req/s



Context Switches?



0					praktiku	m@ke	valu	e: ~		$\mathbf{\nabla}$					
Fil	e I	<u>E</u> dit <u>V</u> i	ew <u>T</u> er	minal <u>F</u>					\backslash						
pra				-\$ vmsta					\rightarrow						
pro			men			swap	o	io-	-/		m		сри	1	-
'n	b	swpd	free	buff	cache	si	SO	bi	DO	ın	cs u	IS S	sy i	d w	а
1	Θ	180	74700	162160	3009088	Θ	Θ	2	22	1	1	3	5	92	Θ
Θ	Θ	180	74692	162160	3009088	Θ	Θ	Θ	Θ	303	661	1	2	97	Θ
0	Θ	180	74692	162160	3009088	Θ	Θ	Θ	Θ	134	377	Θ	Θ	100	Θ
Θ	Θ	180	74692	162160	3009088	Θ	Θ	Θ	Θ	148	390	Θ	Θ	100	Θ
Θ	Θ	180	74692	162160	3009088	Θ	Θ	Θ	Θ	230	472	Θ	Θ	100	Θ
Θ	Θ	180	74692	162160	3009088	Θ	Θ	Θ	Θ	234	477	Θ	Θ	100	Θ
Θ	Θ	180	74692	162160	3009088	Θ	Θ	Θ	Θ	233	476	Θ	Θ	100	Θ
Θ	Θ	180	74468	162160	3009088	Θ	Θ	Θ	13	194	552	19	12	69	Θ
0	Θ	180	74468	162160	3009088	Θ	Θ	Θ	13	134	376	Θ	Θ	100	Θ
3	Θ	180	73964	162160	3009092	Θ	Θ	Θ	Θ	220	3 95	17	11	72	Θ
4	Θ	180	73940	162164	3009100	Θ	Θ	Θ	13	317	949	61	37	2	Θ
4	Θ	180	73824	162164	3009100	Θ	Θ	Θ	11	320	931	58	40	1	Θ
6	Θ	180	73072	162164	3009108	Θ	Θ	Θ	11	318	933	60	37	2	Θ
3	Θ	180	72892	162164	3009116	Θ	Θ	Θ	21	342	981	60	40	1	Θ
4	Θ	180	73824	162164	3009120	Θ	Θ	Θ	11	314	919	60	39	1	Θ
9	Θ	180	52480	162164	3009124	Θ	Θ	Θ	14	318	954	59	39	1	Θ
6	Θ	180	52348	162164	3009132	Θ	Θ	Θ	11	313	1007	60	39	1	Θ
5	Θ	180	52216	162164	3009140	Θ	Θ	Θ	11	313	1011	60	40	Θ	Θ
		m-calls	=50								-				
0		in-catts	5=50		mar	c@xps	s: ~				1			-	_ >
∘ <u>F</u> il				ninal <u>H</u>	mar <u>l</u> elp	c@xps	s: ~							-	• >
			ew <u>T</u> err											-	• •
<u>F</u> il mar	e <u>I</u> c@x	<u>E</u> dit <u>V</u> io	ew <u>T</u> err count mstat 5	is the	<u>l</u> elp number o	f upda	ites.							-	
<u>F</u> il mar pro	e <u>I</u> c@x	<u>E</u> dit <u>V</u> i¢ ps:~\$ v	ew <u>T</u> err count mstat 5 mem	is the ory	lelp number o	f upda swap	ites.	io	<u> </u>				cpu		
<u>F</u> il mar pro	e <u>I</u> c@x cs b	<u>E</u> dit ⊻io ps:~\$ v swpd	ew <u>T</u> err count mstat 5 mem free	is the ory buff	<u>l</u> elp number o cache	f upda swap si	ites.) so	bi	bo	in) CS U	s s	y i	d wa	
<u>F</u> il nar pro r 0	e <u>I</u> c@x cs b 0	Edit <u>V</u> i ps:~\$ v swpd 0	ew <u>T</u> err count mstat 5 mem free 1614328	is the ory buff 264496	<u>l</u> elp number o cache 1039968	f upda swap si 0	ntes.) so 0	bi 47	bo 43	215	495	ss 6	y i 2	 d wa 86	5
Fil nar oro r 0	e <u> </u> c@x cs b 0	Edit Vi ps:~\$ v swpd 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536	is the ory buff 264496 264504	<u>elp</u> number o cache 1039968 1039972	f upda swap si 0 0	otes. 0 so 0 0	bi 47 0	bo 43 3	215 760	495 1113	s s 6 3	y i 2 1	d wa 86 95	5 1
<u>F</u> il mar pro r 0 0	e <u> </u> c@x cs b 0 0 0	Edit <u>V</u> io ps:~\$ v swpd 0 0 0	ew <u>T</u> en count mstat 5 mem free 1614328 1614536 1614888	is the buff 264496 264504 264512	elp number o cache 1039968 1039972 1039976	f upda swap si 0 0 0	otes. so 0 0 0	bi 47 0 0	bo 43 3 8	215 760 879	495 1113 1621	s s 6 3 3	y i 2 1 3	d wa 86 95 94	5 1 0
<u>F</u> il mar pro r 0 0 0	e <u> </u> c@x o b 0 0 0 0	Edit <u>V</u> io ps:~\$ v swpd 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614872	is the buff 264496 264504 264512 264524	Lelp number o cache 1039968 1039972 1039976 1039976	f upda swap si 0 0 0	otes. so 0 0 0 0	bi 47 0 0 0	bo 43 3 8 45	215 760 879 790	495 1113 1621 1185	s s 6 3 3 2	y i 2 1 3 1	d wa 86 95 94 95	5 1 0 2
<u>F</u> il mar pro r 0 0 0	e <u> </u> c@x ccs b 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0	ew <u>T</u> en count mstat 5 mem free 1614328 1614536 1614888 1614872 1614808	is the buff 264496 264504 264512 264524 264524 264524	Lelp number o cache 1039968 1039972 1039976 1039976 1039976	f upda swap si 0 0 0 0 0	otes. 50 0 0 0 0 0	bi 47 0 0 0	bo 43 3 8 45 4	215 760 879 790 761	495 1113 1621 1185 1111	s s 6 3 3 2 1	y i 2 1 3 1 1	d wa 86 95 94 95 98	5 1 0 2 0
<u>F</u> il mar pro 0 0 0 0 5	e <u>i</u> c@x b cs 0 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614872 1614808 1614132	is the buff 264496 264504 264512 264524 264524 264524 264536	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976	f upda swap si 0 0 0 0 0 0 0	otes. so 0 0 0 0 0 0	bi 47 0 0 0 0	bo 43 3 8 45 4 21	215 760 879 790 761 662	495 1113 1621 1185 1111 3965	s s 6 3 3 2 1 32	y i 2 1 3 1 1 10	d wa 86 95 94 95 98 57	5 1 0 2 0 1
<u>F</u> il mar pro 0 0 0 0 5 2	e <u>I</u> c@x b cs b 0 0 0 0 0 0 0	Edit <u>V</u> i ps:~\$ v swpd 0 0 0 0 0 0 0	ew <u>T</u> en count mstat 5 mem free 1614328 1614536 1614888 1614872 1614808 1614132 1613692	is the buff 264496 264504 264512 264524 264524 264524 264552	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988	f upda swap Si 0 0 0 0 0 0 0	otes. so 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0	b0 43 3 45 4 21 112	215 760 879 790 761 662 700	495 1113 1621 1185 1111 3965 4903	s s 6 3 2 1 32 41	y i 2 1 3 1 1 10 10	d wa 86 95 94 95 98 57 47	5 1 0 2 0 1 2
<u>F</u> il mar pro 0 0 0 0 5 2 6	e <u>I</u> cc@xl b 0 0 0 0 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> en count mstat 5 mem free 1614328 1614536 1614888 1614872 1614808 1614132 1613692 1613868	is the buff 264496 264504 264512 264524 264524 264524 264536 264552 264580	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039996 1039988 1039992	f upda swap Si 0 0 0 0 0 0 0 0 0	otes. so 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41	215 760 879 790 761 662 700 635	495 1113 1621 1185 1111 3965 4903 5898	s s 6 3 2 1 32 41 54	y i 2 1 3 1 10 10 10	d wa 86 95 94 95 98 57 47 27	5 1 0 2 0 1 2 1
<u>F</u> il mar pro 0 0 0 0 0 5 2 6 6	e <u>I</u> cc@x 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614872 1614808 1614132 1613692 1613868 1614536	is the buff 264496 264504 264512 264524 264524 264524 264552 264552 264580 264600	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988 1039988 1039982	f upda swap 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41 49	215 760 879 790 761 662 700 635 540	495 1113 1621 1185 1111 3965 4903 5898 7424	s s 6 3 2 1 32 41 54 70	y i 2 1 3 1 10 10 17 26	d wa 86 95 94 95 98 57 47 27 5	5 1 0 2 0 1 2 1 0
<u>F</u> il mar pro 0 0 0 0 0 0 5 2 6 6 3	e <u>I</u> cc@x 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614872 1614808 1614132 1613692 1613868 1614536 1613824	is the buff 264496 264504 264512 264524 264524 264524 264526 264526 264580 264600 264600	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039996 1039988 1039988 1039984 1040008	f upda swap 0 0 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41 49 13	215 760 879 790 761 662 700 635 540 523	495 1113 1621 1185 1111 3965 4903 5898 7424 7179	s s 6 3 2 1 32 41 54 70 71	y i 2 1 3 1 10 10 17 26 28	d wa 86 95 94 95 98 57 47 27 5 1	5 1 0 2 0 1 2 1 0 0
<u>F</u> il mar pro 0 0 0 0 0 0 5 2 6 3 6 3	e <u>I</u> ccs b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> i ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614872 1614808 1614132 1613692 1613868 1614536 1613824 1613880	is the buff 264496 264504 264512 264524 264524 264524 264552 264552 264580 264600 264604 264616	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988 1039988 1039984 1040008 1039984	f upda swap 0 0 0 0 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41 49 13 26	215 760 879 790 761 662 700 635 540 523 550	495 1113 1621 1185 1111 3965 4903 5898 7424 7179 6194	s s 6 3 2 1 32 41 54 70 71 67	y i 2 1 3 1 10 10 17 26 28 28	d wa 86 95 94 95 98 57 47 27 5 1 5	5 1 0 2 0 1 2 1 0 0 0
<u>F</u> il mar pro 0 0 0 0 0 5 2 6 5 3 6 3 8	e <u> </u> cc3 b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614536 1613868 1614536 1613824 1613880 1613356	is the buff 264496 264504 264512 264524 264524 264526 264552 264580 264600 264600 264604 264616 264632	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988 1039988 1039984 1040008 103984 103984	f upda swap 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 45 41 112 41 49 13 26 53	215 760 879 790 761 662 700 635 540 523 550 585	495 1113 1621 1185 1111 3965 4903 5898 7424 7179 6194 5656	s s 6 3 2 1 32 41 54 70 71 67 60	y i 2 1 1 10 10 17 26 28 28 27	d wa 86 95 94 95 95 95 27 27 5 1 5 12	5 1 0 2 0 1 2 1 0 0 0 1
<u>F</u> il mar pro 0 0 0 0 0 5 2 6 6 3 6 8 2	e <u>I</u> cc@x b cs b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614536 1613868 1614536 1613868 1613356 1612484	is the buff 264496 264504 264512 264524 264524 264526 264552 264580 264600 264600 264604 264616 264632 264636	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988 1039984 1039984 1039984 1039984 1039984	f upda swap 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41 49 13 26 53 26	215 760 879 790 761 662 700 635 540 523 550 585 498	495 1113 1621 1185 1111 3965 4903 5898 7424 7179 6194 5656 5901	s s 6 3 2 1 32 41 54 70 71 67 60 65	y it 2 1 3 1 10 10 17 26 28 28 28 27 32	d wa 86 95 94 95 95 57 47 27 5 1 5 12 3	5 1 0 2 0 1 2 1 0 0 0 1 0 0
<u>F</u> il mar pro 0 0 0 0 5 2 6 6 3 6 8 2 7	e <u>I</u> cc@x b cs b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> ie ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614536 1613868 1614536 1613824 1613868 1613356 1612484 1614586	is the buff 264496 264504 264512 264524 264524 264526 264526 264580 264600 264600 264604 264616 264632 264636	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988 1039988 1039984 1040008 1039984 1039984 1039984 1039984	f upda swap 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41 49 13 26 53 26 17	215 760 879 790 761 662 700 635 540 523 550 585 498 525	495 1113 1621 1185 1111 3965 4903 5898 7424 7179 6194 5656 5901 5768	s s 6 3 2 1 32 41 54 70 71 67 60 65 64	y ic 2 1 3 1 10 10 17 26 28 28 27 32 28	d wa 86 95 94 95 98 57 47 27 5 12 5 12 3 7	5 1 0 2 0 1 2 1 0 0 0 1 0 1 0
<u>F</u> il mar pro 0 0 0 0 0 5 2 6 6 3 6 8 2 7 1	e <u>I</u> cc@x b cs b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> i ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614522 1613868 1614324 1613868 1613356 1613824 1613868 1613356 1612484 1614586 1615112	is the buff 264496 264504 264512 264524 264524 264526 264552 264580 264600 264600 264606 264632 264636 264636 264636	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988 1039984 1039988 1039992	f upda swap 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41 49 13 26 53 26 17 23	215 760 879 790 761 662 700 635 540 523 550 585 498 525 768	495 1113 1621 1185 1111 3965 4903 5898 7424 7179 6194 5656 5901 5768 2450	s s 6 3 2 1 32 41 54 70 71 67 60 65 64 19	y it 2 1 3 1 10 10 10 17 26 28 28 27 32 28 8 8	d wa 86 95 94 95 98 57 47 27 5 12 3 7 71	5 1 0 2 0 1 2 1 0 0 0 1 0 1 1
<u>F</u> il mar pro 0 0 0 0 5 2 6 6 3 6 8 2 7	e <u>I</u> cc@x b cs b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Edit <u>V</u> i ps:~\$ v swpd 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ew <u>T</u> err count mstat 5 mem free 1614328 1614536 1614888 1614522 1613868 1614324 1613868 1613356 1613824 1613868 1613356 1612484 1614586 1615112	is the buff 264496 264504 264512 264524 264524 264526 264552 264580 264600 264600 264606 264632 264636 264636 264636	Lelp number o cache 1039968 1039972 1039976 1039976 1039976 1039976 1039988 1039988 1039984 1040008 1039984 1039984 1039984 1039984	f upda swap 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	so 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bi 47 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	bo 43 3 45 4 21 112 41 49 13 26 53 26 17	215 760 879 790 761 662 700 635 540 523 550 585 498 525 768	495 1113 1621 1185 1111 3965 4903 5898 7424 7179 6194 5656 5901 5768	s s 6 3 2 1 32 41 54 70 71 67 60 65 64 19	y it 2 1 3 1 10 10 10 17 26 28 28 27 32 28 8 8	d wa 86 95 94 95 98 57 47 27 5 12 5 12 3 7	5 1 0 2 0 1 2 1 0 0 0 1 0 1 0

httperf --client=0/1 --server=localhost --port=80 --uri=/praktikum --send-buffer -

server ->

home ->

lls=50

Home Tests:

httperf --server=localhost --port=4567 --uri=/test --numconns=100 --num-calls=50

Webserver: "thin"

EventMachine: a library for Ruby, C++, and Java programs. It provides eventdriven I/O using the Reactor pattern

single connection

Total: connections 1 requests 500 replies 500

	home	
Request rate	56.7 req/s (17.6 ms/req)	
Reply rate [replies/s]	min 41.6 avg 41.6 max 41.6	
Reply time [ms]	response 17.6 transfer 0.0	
Reply status:	1xx=0 2xx=500 3xx=0 4xx=0 5xx=0	
test-duration	8.824 s	

100 connections

Total: connections 100 requests 5000 replies 5000

	home
Request rate	48.5 req/s (20.6 ms/req)
Reply rate [replies/s]	min 16.2 avg 48.7 max 71.6
Reply time [ms]	response 20.6 transfer 0.0
Reply status:	1xx=0 2xx=5000 3xx=0 4xx=0 5xx=0
test-duration	103.160 s

2nd Experimental Setup

Replacing Apache+mod_php by Nginx+PHP-FPM

Nginx ...



is...

- a lightweight Web Server
- a Reverse Proxy
- an IMAP/POP3 proxy
- used by 14,988,610 domains today
- implmented by larges sites as WordPress, Github, SourceForge etc.

Nginx vs Apache

• Apache

- process based
- each connection requires a new thread
- high concurrency
- -> high memory usage
- -> CPU overhead (e.g. context switches)
- PHP is usally included in Apache Web Server as module (mod_php)

• Nginx

- fork of apache 1.3 with the multi-processing ripped out in favor of an event loop
- asynchronous model (event based)
- uses only one thread for all connections (master thread)
- PHP is used as seperate process over FastCGI (PHP-FPM)
 - Web Server and PHP-FPM are used as seperate applications
 - communication via TCP-connetions or Unix-sockets
 - -> little overhead due to communication costs

Event Loop

What is an event loop?

usually you write code like: var result = db.query("select.."); result.do_something();

but an event loop looks like: db.query("select..", function (result) { result.do_something()});

Motivation

Apache+mod_php compared to Nginx+php-fpm

(comparison made by Boštjan Škufca - http://blog.a2o.si))

5 Different scenarios

- *HelloWorld.php simple echo of* "Hello, World!" (13 bytes),
- HelloWorld.txt static file with "Hello, World!" (also 13 bytes)
- 100KB.txt static content
- 1MB.txt static content
- index.php more complex site with several DB-queries, HTML template parsing...

Tests with keepalive-feature [-k] and without keepalive (same socket can be used for several requests)

Benchmark Setup

Benchmark tests conducted using ApacheBench

- ab -n <u>NREQ</u> -c <u>NCONC</u> [-k] http://server.domain. com/bench/FileName NREQ is the number of requests:
 - HelloWorld.php: 500000
 - HelloWorld.txt: 500000
 - 100KB.txt: 500000
 - 1MB.txt: 50000
 - AppFront: 5000

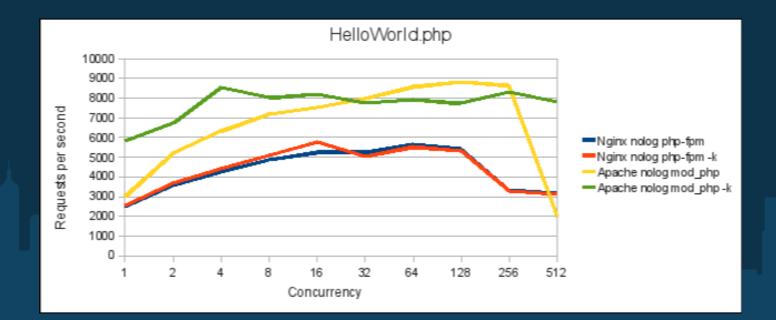
NCONC = number of concurrent requests

1,2,4,8,16,32,64,128,256,512

PHP-generated Hello World!

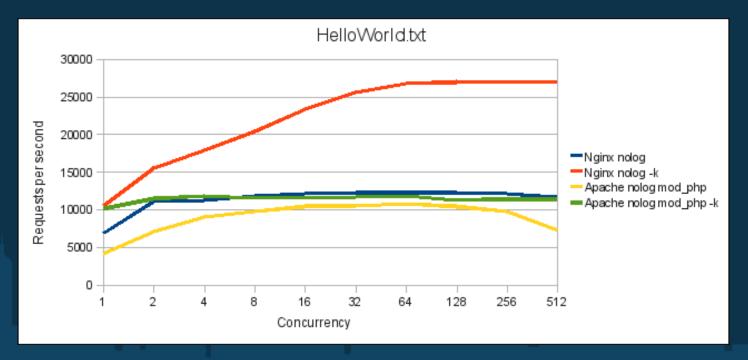
Apche is always faster than Nginx

 This demonstrates the overhead of the communication between Nginx and PHP-FPM



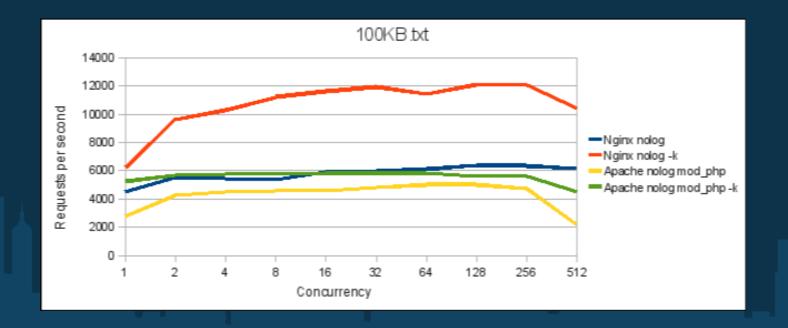
Static Hello World!

- Nginx with keepalive is more than twice as fast as Apache
- This demonstrates the overhead that is caused by creating TCP-connections



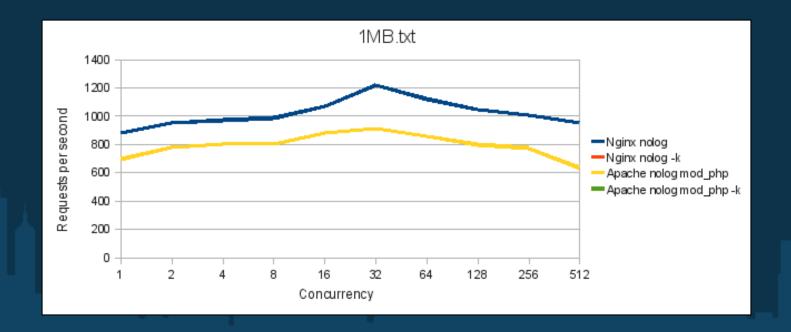
Static 100kb.txt File

This test should demonstrate a "real world " example of a static page request
Again, Nginx is twice as fast as Apache



1MB.txt File

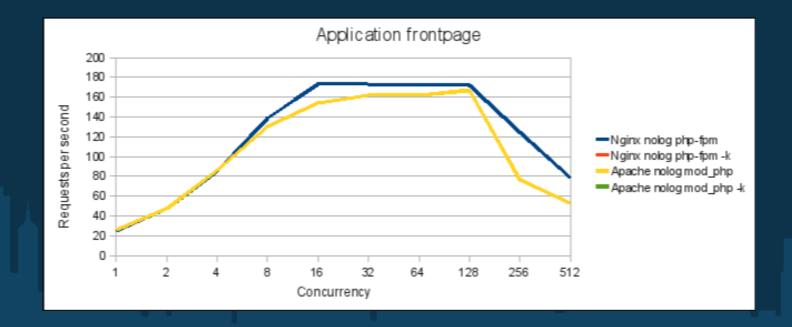
- This test demonstates a more complex file transfer
- Keepalive was not tested, because the file size is so large that TCP-connections aren't important
- Nginx is just slightly better



Application Frontpage (index. php)

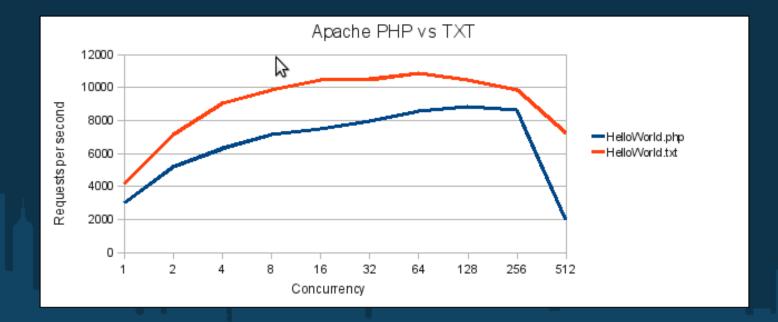
Again a "real world" example with a more complex PHP-site

Nginx is just slightly better



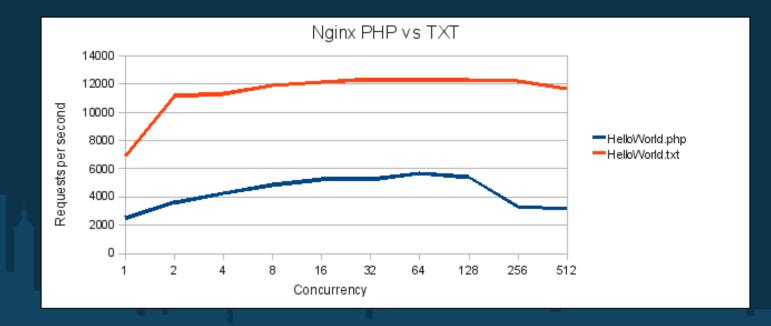
Apache PHP vs TXT

 Dynamically generated content and static content are nearly equally fast

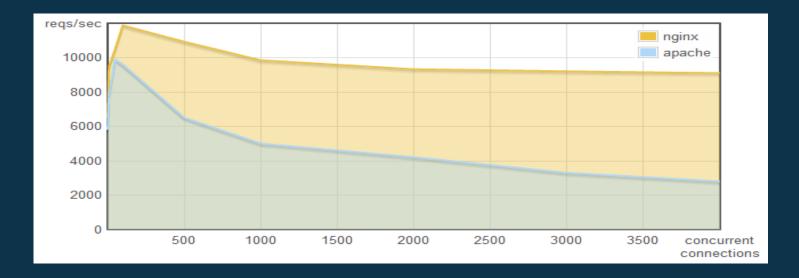


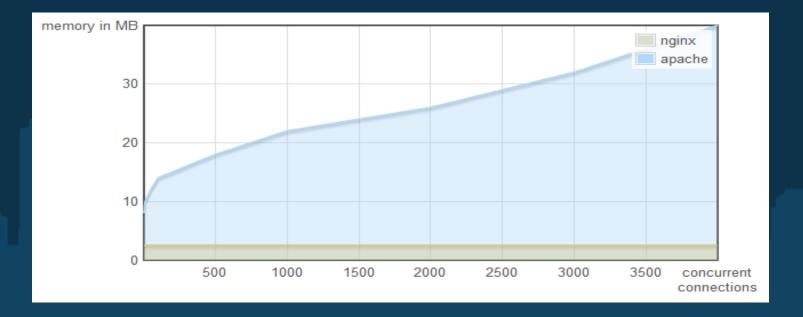
Nginx PHP vs TXT

Nginx serves static content twice as fast as dynamic content



Additional Comparison





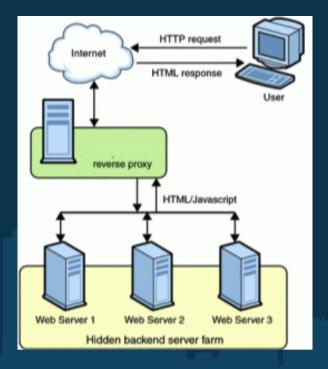
Conclusion

• When is it worth to use Nginx?

- If you have limited hardware resources (e.g. on VPS)
- $\,\circ\,$ If you have a lot of static content

Further Alternative

- Nginx works as reverse proxy
- static content is passed by Nginx (e.g. Pictures)
- dynamic Content will be forwarded to an Apache behind the proxy
- advantages:
 - static content will be returned very fast
 slow user connections do no longer hold resources, because the "blocking" connection is now between Nginx and Apache; not the user and Apache



our problems

mysql dumps: create dump + copy dump + insert dump = hours

- loadtesting:
 - client == server
- --> no testing for high concurrency, no isolation of variables.
 - client too slow
 - different configurations
 - keepalives
 - Nginx workers/processes vs apache threads/clients
 - 0...