Сравнение производительности bcache, dm-cache, flashcache, gflash

май 2015

Данное тестирование систем кэширования с помощью флэш накопителей предназначено для сравнения производительности в режиме отложенной записи (далее **writeback**) для основных типов нагрузки при различных состояниях систем кэширования, а также предназначено для понимания их работы и целесообразности применения.

Участники тестирования:

bcache - включен в ядро **Linux** (www.kernel.org/doc/Documentation/bcache.txt);
dm-cache - включен в ядро **Linux** (www.kernel.org/doc/Documentation/device-mapper/cache.txt);
flashcache - модуль для ядра **Linux** (github.com/facebook/flashcache);
gflash - модуль geom для **FreeBSD** (github.com/geomflash/geomflash).

Платформа для тестирования:

MB - Intel® Server Board S1200BTL;
CPU - Intel® Xeon® Processor E3-1230 (8M Cache, 3.20 GHz);
RAM - 16GB;
HDD - WD Black 2TB.

Диски для системы кэширования:

cache - Samsung SSD 850 PRO 256GB (далее **ssd**);
data - WD Re 4TB (далее **hdd**).

Операционная система:

Linux ubuntu 3.16.0-31-generic для **bcache, dm-cache, flashcache;**
FreeBSD 10.1-RELEASE-p5 для **gflash.**

Программа тестирования:

fio-2.1.11 (ioengine=libaio) для **bcache, dm-cache, flashcache;**
fio-2.1.9 (ioengine=sync) для **gflash** (выбор sync связан с реализацией **fio** в FreeBSD).

Тестирование выполняется с помощью sh-скрипта и включает в себя:
– идентификацию операционной системы и дисков для системы кэширования;
– проверку состояния **ssd**;
– создание блочного устройства с кэшированием на **ssd**;
– вывод начальной статистики;
– последовательное выполнение специально подобранных заданий **fio** (всего 21 задание);
– каждое задание **fio** завершается выводом системной статистики;
– все данные тестирования записываются в файл для последующего анализа.

Для каждой системы кэширования создан свой sh-скрипт, учитывающий особенности создания кэша и отображения статистики в процессе работы. Все системы при создании настроены на кэширование запросов любого размера и типа нагрузки. Все остальные параметры оставлены по умолчанию.

Все файлы тестирования доступны на github.com/geomflash/geomflash/test_cache.

Перед началом анализа данных тестирования коротко опишем основные факторы, влияющие на производительность системы кэширования в режиме **writeback**:
– тип запроса ввода-вывода (чтение, запись);
– размер запроса ввода-вывода;
– тип нагрузки (последовательная, случайная, смешанная);

– состояние **ssd** - очищенный или заполненный (количество очищенных блоков - влияет на скорость записи флэш накопителя);
– количество незаписанных блоков (далее **dirty)** - определяет режим работы отложенной записи.

Для получения максимальной скорости записи перед началом тестирования **ssd** очищается и проводится перезагрузка системы.

Для определения состояния **ssd** проводится тест случайного чтения 4KB-блоками одним потоком. У заполненного **ssd** скорость чтения приблизительно равна заявленным 10000 iops (Samsung_SSD_850_PRO_Data_Sheet_rev_2_0.pdf). У очищенного **ssd** скорость чтения значительно превосходит 10000 iops (для **Linux** около 22000 iops, для **FreeBSD** около 30000 iops).

Далее создается блочное устройство с кэшированием на **ssd**, выдерживается пауза 10 секунд, снимается начальная статистика.

Для анализа из статистики блочного устройства с кэшированием на **ssd** используется только количество **dirty** блоков. Из статистики **ssd** и **hdd** количество считанной и записанной информации. Собранная информация оформляется в виде сравнительных таблиц. Анализ производительности производится относительно **ssd** или **hdd**, в зависимости от условий задания. Производительность **ssd** отличается в **Linux** и **FreeBSD**, поэтому сравнение с **ssd** производится для каждой операционной системы отдельно. В Linux **I/O Scheduler** для **ssd** и **hdd** оставлен по умолчанию **deadline** (для **ssd** сравнение с **noop** показало незначительные различия в производительности). Также производится абсолютное сравнение производительности между системами кэширования. Для понимания работы кэширования анализируется количество записанной и считанной информации с **ssd** и **hdd**.

Далее описывается цель заданий **fio** и анализ полученных результатов.

Оценка максимальной скорости чтения большими блоками

Исходное состояние: кэш пуст, **ssd** очищен, **dirty**=0.

Выполняется троекратное последовательное чтение 8GB блоками по 128KB.

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
1. --rw=read --bs=128k --iodepth=1 --size=8g --offset=1g							
hdd	173630	1356					
bcache	173236	1353	0	8407740	8388608	0	0
dm-cache	173620	1356	0	4	8388608	0	0
flashcache	132811	1037	0	8388608	8388608	0	0
gflash	173616	1356	248	28	8388632	0	0
2. --rw=read --bs=128k --iodepth=1 --size=8g --offset=1g							
bcache	350548	2738	8175488	213200	213120	0	0
dm-cache	31746	248	8388856	9175044	8388608	0	0
flashcache	269245	2103	8388608	0	0	0	0
gflash	173656	1356	272	8454144	8388608	0	0
3. --rw=read --bs=128k --iodepth=1 --size=8g --offset=1g							
bcache	363789	2842	8388608	12	0	0	0
dm-cache	309829	2420	8388608	4	0	0	0
flashcache	269956	2109	8388608	0	0	0	0
ssd_linux	448373	3502					
gflash	390458	3050	8388880	0	0	0	0
ssd_FreeBSD	497811	3889					

Три последовательных чтения выбраны, потому что кэширование чтения может происходить по первому чтению (**bcache, flashcache**), по второму чтению - (**gflash**) либо настраиваться (**dm-cache** - настроен кэшировать после второго чтения).

Первое чтение (задание №1):

bcache - читает с **hdd** и записывает (кэширует) блоки на **ssd**, скорость чтения равна максимальной скорости чтения с **hdd**;

dm-cache - читает с **hdd** и помечает прочитанные блоки (увеличивает счетчики чтений блоков), на **ssd** запись (кэширование) не производится, скорость чтения равна максимальной скорости чтения с **hdd** *(незначительная запись 4KB на **ssd** наверно связана с особенностями работы данной системы кэширования и в дальнейшем не будет приниматься во внимание)*;

flashcache - читает с **hdd** и записывает (кэширует) блоки на **ssd**, скорость чтения на 23% ниже максимальной скорости чтения с **hdd**;

gflash - читает с **hdd**, на **ssd** запись (кэширование) не производится, скорость чтения равна максимальной скорости чтения с **hdd** *(дополнительное чтение с **hdd** 248KB и запись (кэширование) 28KB на **ssd** связано с особенностями функционирования **FreeBSD** и в дальнейшем будет считаться накладными расходами данной операционной системы и не будет приниматься во внимание).*

Второе чтение (задание №2):

bcache - читает с **ssd** (97%) и с **hdd** (3%). Записывает (кэширует) прочитанные с **hdd** 3% информации на **ssd** *(непонятна причина, почему при первом чтении в кэш не попал весь объём считанных с **hdd** данных)*. Скорость чтения на 22% ниже максимальной скорости чтения с **sdd**;

dm-cache - читает с **hdd** 8GB и читает с **ssd** 8GB. Записывает (кэширует) на **ssd** 8,75GB. Скорость чтения на 82% ниже максимальной скорости чтения с **hdd** *(такое странное поведение и соответственно крайне низкую скорость чтения можно попытаться объяснить только такой последовательностью действий: при поступлении запроса на чтение 128KB, если читаемый блок подлежит кэшированию, то производится чтение с **hdd** блока размером 256KB (размер блока кэша), запись данного блока на **ssd** и уже затем чтение 128KB блока с **ssd**, если 128KB блок находится уже в кэше, то чтение производится с **ssd**, возможно я ошибаюсь, но иного объяснения у меня не получается)*;

flashcache - читает с **ssd**, скорость чтения на 40% ниже максимальной скорости чтения с **ssd**;

gflash - читает с **hdd** и записывает (кэширует) блоки на **ssd**, скорость чтения равна максимальной скорости чтения с **hdd**.

Третье чтение (задание №3):

Все системы кэширования производят чтение только из кэша (**ssd**).

Итоги:

Скорость чтения во время кэширования по сравнению с максимальной скоростью чтения с **hdd**:

gflash	100%;
bcache	100%;
flashcache	76%;
dm-cache	18%.

Скорость чтения из кэша по сравнению с максимальной скоростью чтения с **ssd**:

bcache	81%;
gflash	78%;
dm-cache	69%;
flashcache	60%.

Абсолютная скорость чтения из кэша:

gflash 100%;
bcache 93%;
dm-cache 79%;
flashcache 69%.

Объём записанных метаданных по сравнению с объемом кэшируемых данных составляет:

flashcache 0,0% *(удивительный результат, по факту метаданные в кэш не записываются)*;
gflash 0,8%;
bcache 2,8%;
dm-cache 9,4%.

Выводы: чтение большими блоками, исходя из результатов тестирования, нужно оценивать в двух фазах - скорость чтения при записи в кэш (кэширование) и чтение из кэша.

gflash и **bcache** кэширование - очень хорошо, чтение из кэша - хорошо;
flashcache кэширование - удовлетворительно, чтение из кэша - удовлетворительно;
dm-cache кэширование - плохо *(возникло предположение, что такая низкая производительность из-за большого размера блока кэша в 256KB. Проверено - при уменьшении размера блока кэша до 128KB производительность становится ещё ниже)*, чтение из кэша - немного лучше чем у **flashcache**.

Оценка максимальной скорости чтения малыми блоками из кэша

Исходное состояние: кэш заполнен данными на 8GB, **dirty**=0.
Выполняется случайное чтение 4KB-блоками из кэшированной области размером 8GB.

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
4. --rw=randread --bs=4k --iodepth=1 --size=1g --offset=1g --filesize=9g							
bcache	33569	8392	1048576	0	0	0	0
dm-cache	33061	8265	1048576	4	0	0	0
flashcache	34153	8538	1048576	0	0	0	0
ssd_Linux	35877	8969					
gflash	37560	9390	1048848	0	0	0	0
ssd FreeBSD	39343	9835					
5. --rw=randread --bs=4k --iodepth=32 --size=4g --offset=1g --filesize=9g							
bcache	397263	99315	4194304	0	0	0	0
dm-cache	387644	96910	4194304	4	0	0	0
flashcache	399953	99988	4194304	0	0	0	0
ssd_Linux	397301	99325					
gflash	398244	99560	4198792	0	0	0	0
ssd FreeBSD	397866	99466					

Итоги:

При очереди ***QD=1 (задание №4)*** скорость чтения из кэша по сравнению с максимальной скоростью чтения с **ssd** составляет:

gflash 95%;

flashcache 95%;
bcache 94%;
dm-cache 92%

Абсолютная скорость чтения из кэша:

gflash 100%;
flashcache 91%;
bcache 89%;
dm-cache 88%.

При очереди ***QD=32 (задание №5)*** скорость чтения из кэша по сравнению с максимальной скоростью чтения с **ssd** составляет:

flashcache 100%;
gflash 100%;
bcache 100%;
dm-cache 98%.

Абсолютная скорость чтения из кэша:

flashcache 100%;
gflash 100%;
bcache 99%;
dm-cache 97%.

Выводы: скорость чтения малыми блоками из кэша у всех систем кэширования почти равна максимальной скорости чтения с **ssd**. Можно только отметить небольшое отставание **dm-cache**.

Следующие задания будут связаны с работой кэша в режиме **writeback**, поэтому для понимания необходимо кратко описать алгоритмы работы и различия в архитектуре систем кэширования.

В режиме **writeback** данные записываются сначала в кэш (**ssd**) и затем переносятся в основное хранилище (**hdd**). При записи в кэш записываются данные и метаданные. Метаданные необходимы для возобновления работы кэша после перезагрузки системы. Для режима **writeback** важную роль играет максимально возможный объем незаписанных (**dirty**) данных на **hdd**. До достижения этого порога кэш находится в режиме заполнения. При полном заполнении переходит в форсированный режим. В режиме заполнения алгоритм отложенной записи определяет, как данные будут переноситься на **hdd**. Основная задача данного алгоритма способствовать максимальной производительности, поэтому настройка может производиться по различным критериям (по временному интервалу, в зависимости от текущей нагрузки на блочное устройство или другим критериям). В режиме заполнения скорость записи может приближаться к максимальной скорости записи **ssd**. В форсированном режиме необходимо постоянно записывать **dirty** блоки для освобождения места в кэше. Поэтому в форсированном режиме максимальная скорость записи ограничивается максимальной скоростью записи **hdd**.

В режиме **writeback** выход со строя кэша с большой вероятностью влечет за собой потерю всех данных. В **bcache**, **dm-cache** и **flashcache** надежность функционирования **writeback** кэша может обеспечиваться путем создания **raid1** из **ssd** дисков. В **gflash** имеется встроенный механизм объединения в **raid1** для **writeback** кэша выделенного объема оперативной памяти (**ram**) и такого же объема на **ssd**. Преимущество данного метода заключается в отсутствии чтения с **ssd** во время выполнения отложенной записи и увеличении скорости чтения при наличии данных в **ram**. Недостатком является малый объем **writeback** кэша (ограничен объемом свободной оперативной памяти в системе) и дополнительным копированием в оперативную память при кэшировании.

Для **gflash** в данном тестировании размер **writeback** кэша определен в 4GB, для остальных систем размер **writeback** кэша и настройки оставлены по умолчанию.

Оценка максимальной скорости записи большими блоками

Исходное состояние: кэш заполнен данными на 8GB, **dirty**=0.

Выполняется последовательная запись 4GB блоками по 128KB в некэшированную область. Объем записи в 4GB выбран, исходя из минимального размера **writeback** кэша у **gflash**.

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
6. --rw=write --bs=128k --iodepth=1 --size=4g --offset=25g							
bcache	320714	2505	1108	4202388	456	132	4096
dm-cache	22481	175	8390548	8782668	4195136	8388608	0
flashcache	266711	2083	21824	4326012	0	19968	4077
ssd_Linux	417515	3261					
gflash	457793	3576	272	4227072	0	39680	4056
ssd FreeBSD	462386	3612					

Скорость записи на **ssd** зависит от количества свободных (незаписанных) блоков. В нашем случае на **ssd** записано немного больше 8GB, поэтому сравнение происходит с максимальной скоростью записи на очищенный **ssd**.

Итоги:

Скорость записи по сравнению с максимальной скоростью записи на очищеный **ssd** составляет:

gflash 99%;
bcache 77%;
flashcache 64%;
dm-cache 5%.

Абсолютная скорость записи:

gflash 100%;
bcache 70%;
flashcache 58%;
dm-cache 5%.

Объем записанных метаданных по сравнению с объемом записываемых данных составляет:

bcache 0,2%;
gflash 0,8%;
flashcache 3,1%;
dm-cache 109,4%.

Выводы:

gflash **-** очень хорошо;
bcache **-** хорошо;
flashcache - удовлетворительно;
dm-cache **-** очень плохо *(происходит малопонятный обмен блоками между **hdd** и **ssd**, в итоге имеем значительные объёмы чтения и записи на дисках и отсутствие **dirty** блоков, то есть по факту **dm-cache** не является **writeback** кэшем).*

Оценка скорости записи большими блоками при выполнении фоновой отложенной записи

Исходное состояние: кэш заполнен данными на 12GB, **dirty** блоков около 4GB *(**dm-cache** в расчет не принимается)***.**

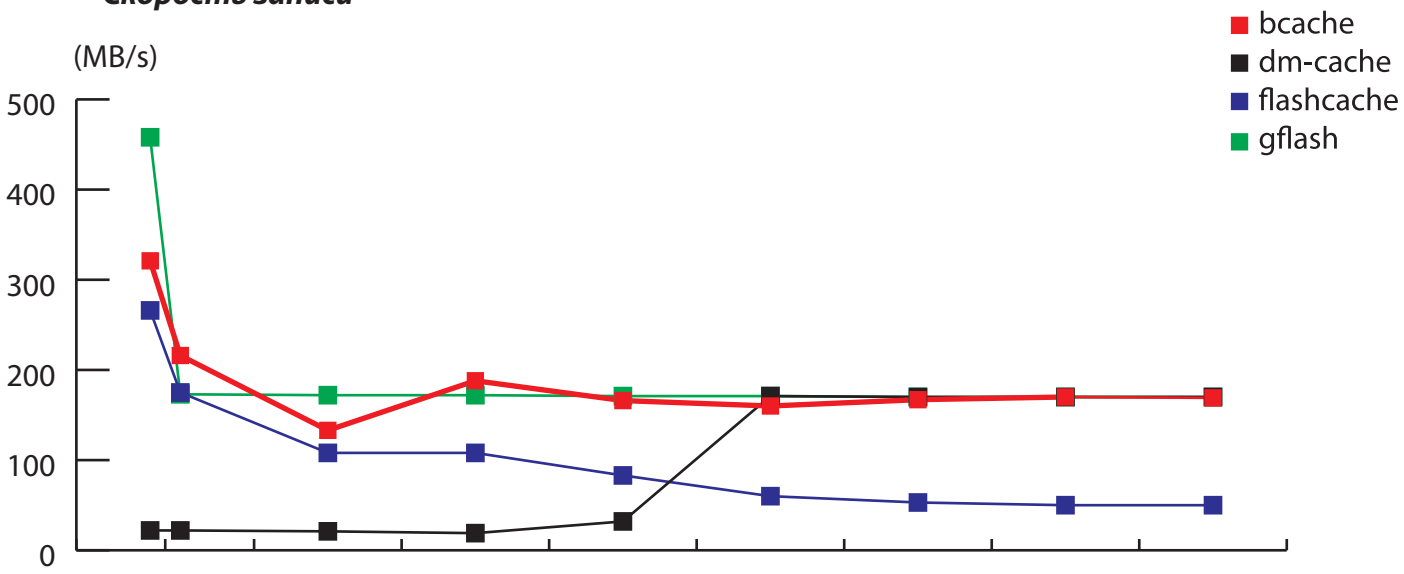
Выполняется последовательная запись 512GB (восемь раз по 64GB) блоками по 128KB. Объём записи в 512GB выбран для двукратной перезаписи кэша (**ssd)**. Цель данного теста - заполнить весь объем **writeback** и заставить кэш форсированно переносить (записывать) данные на **hdd.** Также данный тест позволит оценить работу системы кэширования с флэш накопителем (особенность флэш памяти - падение скорости при перезаписи).

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
7. --rw=write --bs=128k --iodepth=1 --size=64g --offset=29g							
hdd	173641	1356					
bcache	216387	1690	34064672	67237920	0	34052096	36352
dm-cache	21639	169	134221596	140509188	67108928	134217728	0
flashcache	174976	1366	14368776	69792812	0	14368128	55581
gflash	172792	1349	272	67633152	0	67086080	4076
8. --rw=write --bs=128k --iodepth=1 --size=64g --offset=93g							
bcache	133232	1040	86754592	67247740	0	86732800	17203
dm-cache	21265	166	134221596	140509188	67108928	134217728	0
flashcache	108076	844	24731956	69789072	0	24981052	96722
gflash	171675	1341	272	67633160	0	67120000	4068
--rw=write --bs=128k --iodepth=1 --size=64g --offset=157g							
bcache	188410	1471	56192800	67241640	0	56189440	27853
dm-cache	19071	148	134221596	140509188	67108928	134217728	0
flashcache	108492	847	25199136	66649276	184	28430328	134494
gflash	171650	1341	272	67633152	0	67126400	4048
9. --rw=write --bs=128k --iodepth=1 --size=64g --offset=221g							
bcache	166331	1299	68243232	67243924	0	68269568	26726
dm-cache	32026	250	72259420	75643540	36128320	103237120	0
flashcache	82571	645	29937052	58286996	52	41412848	159588
gflash	171335	1338	176	67633196	96	67096832	4060
10. --rw=write --bs=128k --iodepth=1 --size=64g --offset=285g							
bcache	160191	1251	70820640	67243436	0	70798848	23142
dm-cache	171205	1337	1796	4	64	67108864	0
flashcache	60388	471	35052888	48899116	20	55912760	170521
gflash	170901	1335	248	67633188	24	67094528	4076
11. --rw=write --bs=128k --iodepth=1 --size=64g --offset=349g							
bcache	167171	1306	67188000	67243268	0	67207168	23040
dm-cache	170635	1333	1796	4	64	67108864	0
flashcache	53127	415	38197788	42715372	508	65210528	172375
gflash	170242	1330	272	67633152	0	67116544	4068
12. --rw=write --bs=128k --iodepth=1 --size=64g --offset=413g							
bcache	170296	1330	66103584	67242644	0	66087424	23962

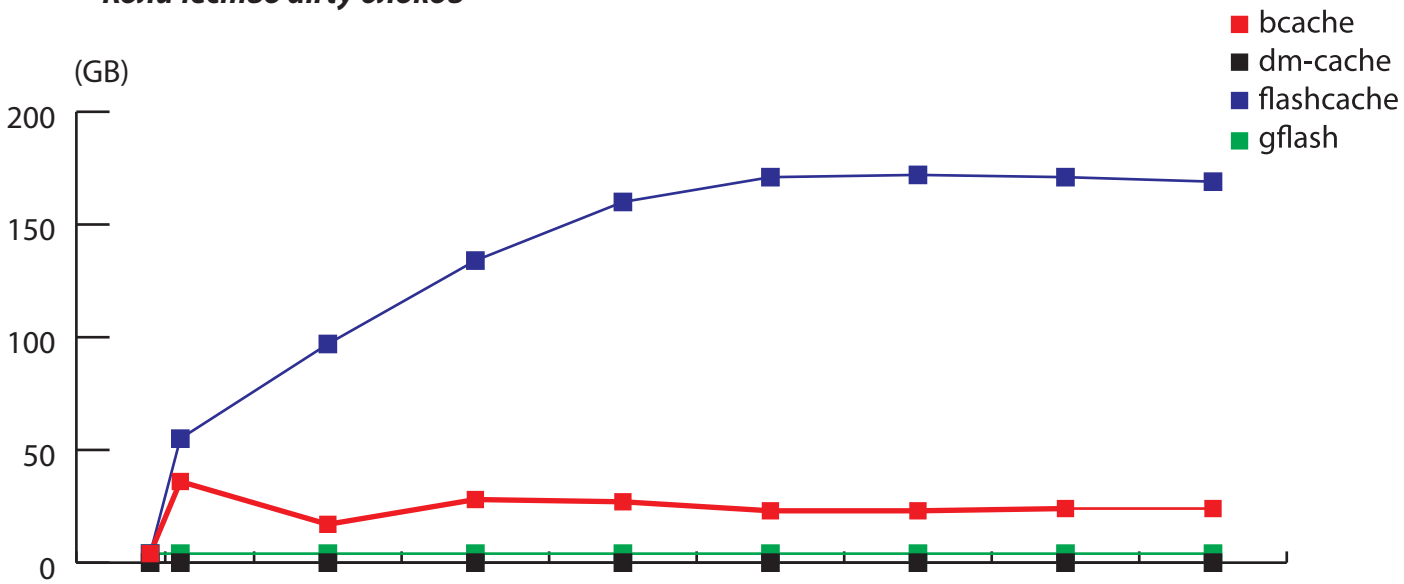
dm-cache	170702	1333	1796	4	64	67108864	0
flashcache	50063	391	39768036	40511104	172	69003140	170525
gflash	170334	1330	272	67633156	0	67099136	4076
13. --rw=write --bs=128k --iodepth=1 --size=64g --offset=477g							
bcache	168797	1318	66743072	67244372	0	66739712	24371
dm-cache	170140	1329	1796	4	64	67108864	0
flashcache	49962	390	39314076	40219264	8	68795084	168878
gflash	169826	1326	176	67633196	96	67128576	4056
hdd	170110	1328					

Для наглядности представим полученные данные в виде графиков.

Скорость записи



Количество dirty блоков



Выводы:

gflash - очень хорошо *(скорость ожидаемо упала до максимальной скорости записи **hdd** и стабильно продержалась до конца теста. Объём **dirty** блоков около 4GB)*;

bcache - хорошо *(вначале небольшие колебания относительно максимальной скорости записи **hdd**. Объём **dirty** блоков достигал 36GB. Затем скорость стабилизировалась и продержалась до конца теста. Объём **dirty** блоков стабилизировался около 23GB)*;

flashcache - плохо *(скорость все время падала. Когда кэш заполнился, система кэширования перестала справляться с записью в кэш и стала писать часть данных прямо на **hdd**. Есть подозрение, что не учитываются свойства флэш памяти при записи. В конце задания скорость упала до 29% от максимальной скорости записи **hdd**. Объём **dirty** блоков всё время рос и достиг к концу задания 168GB)*;

dm-cache - очень плохо *(до полного заполнения кэша опять происходит малопонятный обмен блоками между **hdd** и **ssd**. Очень низкая скорость записи. Значительные объёмы чтения и записи на дисках. Традиционное отсутствие **dirty** блоков. После заполнения кэша, запись производилась только на **hdd** с соответствующей скоростью).*

Оценка скорости кэширования чтения большими блоками при выполнении фоновой отложенной записи

Исходное состояние: кэш заполнен, выполняется отложенная запись, объём **dirty** блоков можно увидеть в предыдущей таблице (***задание №13***)**.**

Выполняется троекратное последовательное чтение 8GB блоками по 128KB аналогичное ***заданиям №1, 2, 3.***

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
14. --rw=read --bs=128k --iodepth=1 --size=8g --offset=1g							
bcache	52760	412	14792832	8409104	8388480	14811648	9933
dm-cache	299925	2343	8388612	4	0	0	0
flashcache	28316	221	9260700	4885200	8105984	8978076	160111
gflash	173566	1355	248	24	8388632	85376	3972
15. --rw=read --bs=128k --iodepth=1 --size=8g --offset=1g							
bcache	347830	2717	8398080	3328	3328	12800	9932
dm-cache	310471	2425	8388608	4	0	0	0
flashcache	38952	304	10382228	1319492	3702052	5695672	154548
gflash	173566	1355	272	8454144	8388608	87040	3888
16. --rw=read --bs=128k --iodepth=1 --size=8g --offset=1g							
bcache	354998	2773	8400896	16	0	12288	9830
dm-cache	309748	2419	8388608	4	0	0	0
flashcache	37299	291	11028304	1065632	2628860	5268556	149403
gflash	380246	2970	8388880	0	0	3805824	172

Итоги:

Скорость чтения во время кэширования ***задание №14*** по сравнению с ***заданием №1*** (для **gflash** ***задание №15*** по сравнению с ***заданием №2***) составляет:

gflash 100% *(отложенная запись в режиме заполнения)*;

bcache 30% *(с большой вероятностью можно предположить, что отложенная запись сначала была в форсированном режиме, затем в режиме заполнения)*;

flashcache 21% *(отложенная запись в форсированном режиме)*;

dm-cache не оценивается (*кэш в предыдущем задании во время записи не обновлялся и поэтому чтение производилось с **ssd**).*

Скорость чтения из кэша ***задание №16*** по сравнению с ***заданием №3***:

dm-cache 100% *(отложенная запись не выполнялась)*;
bcache 98% *(отложенная запись в режиме заполнения)*;
gflash 97% *(отложенная запись в режиме заполнения)*;
flashcache 14% *(отложенная запись в форсированном режиме).*

Абсолютная скорость чтения из кэша:

gflash 100%;
bcache 93%;
dm-cache 81%;
flashcache 10%.

Выводы:

В данном тестировании из-за различных размеров и алгоритмов работы **writeback** кэша системы кэширования находились в разных режимах работы и оцениваются достаточно условно.

gflash кэширование и чтение из кэша - очень хорошо;
bcache кэширование - удовлетворительно, чтение из кэша - хорошо;
flashcache кэширование и чтение из кэша - плохо;
dm-cache в данных тестах не оценивается.

Оценка скорости чтения малыми блоками из кэша при выполнении фоновой отложенной записи

Исходное состояние: кэш заполнен, выполняется отложенная запись, объём **dirty** блоков можно увидеть в предыдущей таблице (***задание №16***)**.**

Выполняется случайное чтение 4KB-блоками из кэшированной области размером 8GB аналогичное ***заданиям №4, 5*.**

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
17. --rw=randread --bs=4k --iodepth=1 --size=1g --offset=1g --filesize=9g							
bcache	33493	8373	1064448	4	0	15872	9830
dm-cache	33090	8272	1048576	4	0	0	0
flashcache	1572	394	10338232	394920	214808	9504464	140121
gflash	37532	9383	1048848	0	0	179200	0
18.--rw=randread --bs=4k --iodepth=32 --size=4g --offset=1g --filesize=9g							
bcache	397942	99485	4199936	0	0	5632	9830
dm-cache	387393	96848	4194304	4	0	0	0
flashcache	11114	2778	7836672	689344	679432	403464	139728
gflash	398282	99570	4198792	0	0	0	0

Итоги:

Задание №17 (QD=1)

Скорость чтения из кэша по сравнению с ***заданием №4*** составляет:

gflash 100% *(отложенная запись в режиме заполнения)*;
bcache 100% *(отложенная запись в режиме заполнения)*;
dm-cache 100% *(отложенная запись не выполнялась)*;
flashcache 4% *(отложенная запись в форсированном режиме)*.

Абсолютная скорость чтения из кэша:

gflash 100%;
bcache 89%;
dm-cache 88%;
flashcache 4%.

Задание №18 (QD=32)

Скорость чтения из кэша по сравнению ***с заданием №5*** составляет:

gflash 100% *(отложенная запись не выполнялась)*;
bcache 100% *(отложенная запись в режиме заполнения)*;
dm-cache 98% *(отложенная запись не выполнялась)*;
flashcache 3% *(отложенная запись в форсированном режиме)*.

Абсолютная скорость чтения из кэша:

gflash 100%;
bcache 100%;
dm-cache 97%;
flashcache 3%.

Выводы:

У систем кэширования во время выполнения заданий наблюдаются различные режимы работы отложенной записи. Поэтому выводы достаточно условны:

gflash производительность чтения практически не зависит от выполнения отложенной записи в режиме заполнения;

bcache в режиме форсированной записи - удовлетворительно, в режиме заполнения - хорошо;

flashcache в режиме форсированной записи - очень плохо. Из-за большого размера и низкой скорости работы **writeback** кэша, в режим заполнения система так и не вернулась;

dm-cache в данных тестах не оценивается.

Оценка производительности при нагрузке, характерной для баз данных

Исходное состояние: кэш заполнен, состояние систем кэширования можно увидеть в предыдущей таблице (***задание №18***)**.**

Выполняется случайное чтение (67%) и запись (33%) общим объемом 4GB блоками по 8KB в кэшированной области размером 8GB QD=32.

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
19. --rw=randrw --rwmixread=67--bs=8k --iodepth=32 --size=4g --offset=1g --filesize=9g							
bcache	147042 72303	18380 9037	2824304	1405412	0	11312	11162
dm-cache	4733 2328	591 290	47046608	1382580	32	44216832	17
flashcache	13355 6568	1669 820	6732832	2399156	178372	4097620	137076
ssd_Linux full	103892 51085	12986 6385					
ssd_Linux clean	302434 148712	37804 18589					
gflash	298031 145389	37253 18173	2608032	1547140	0	104472	1145
ssd_FreeBSD full	114332 55775	14291 6971					
ssd_FreeBSD clean	222904 108740	27863 13592					

Итоги:

Системы кэширования выполняют данное задание в различных режимах работы. Поэтому оцениваются достаточно условно, но результаты данного задания позволяют сделать некоторые интересные выводы. В таблице представлены скорости заполненного (**full**) и очищенного (**clean**) **ssd**.

Абсолютная производительность:

gflash 100% *(отложенная запись в режиме заполнения. В данном тесте впервые не удалось избежать чтения из оперативной памяти, было прочитано 210MB, что составляет около 8% от объёма чтения. Сколько это дало в общий прирост производительности оценить сложно, так как основным сдерживающим фактором в данном тесте является скорость записи. Скорость записи явно превосходит скорость записи заполненного **ssd**, что явно указывает на предварительную очистку флэш памяти перед записью)*;

bcache 49% *(отложенная запись в режиме заполнения. Скорость записи немного превосходит скорость записи заполненного **ssd**, но в два раза уступает скорости очищенного, поэтому однозначно оценить алгоритм записи во флэш память затруднительно)*;

flashcache 4% *(отложенная запись в форсированном режиме. Производительность низкая. Запись во флэш память явно производится без предварительной очистки)*;

dm-cache 2% *(снова очень низкая производительность. Огромные объёмы чтения и записи. Появились первые **dirty** блоки. Это дает право предположить, что **dirty** блоки появляются, если запись прозводится в блоки находящиеся в кэше).*

Выводы:

gflash - очень хорошо;
bcache - хорошо;
flashcache - плохо;
dm-cache - очень плохо.

Оценка скорости записи малыми блоками при выполнении фоновой отложенной записи

Исходное состояние: кэш заполнен, выполняется отложенная запись, объём **dirty** блоков можно увидеть в предыдущей таблице (***задание №19*****).**

Выполняется случайная запись 4KB-блоками в некэшированную область размером 8GB.

	KB/s	iops	cache (ssd)		data (hdd)		dirty (MB)
			read	write	read	write	
20. --rw=randwrite --bs=4k --iodepth=1 --size=1g --offset=9g --filesize=17g							
bcache	43185	10796	14112	1071980	0	12288	12486
dm-cache	709	177	50362060	10602452	8265744	50640176	0
flashcache	2193	548	12944296	1919684	0	13207248	125203
ssd_Linux full	57481	14370					
ssd_Linux clean	109215	27303					
gflash	196916	49228	272	1310720	0	101992	2069
ssd_FreeBSD full	91157	22798					
ssd_FreeBSD clean	115190	28797					
21. --rw=randwrite --bs=4k --iodepth=32 --size=4g --offset=17g --filesize=25g							
bcache	91855	22963	25376	4289216	0	23552	16282
dm-cache	1120	279	251508184	13270664	8388624	251739940	29
flashcache	5816	1453	15286156	8002280	0	16725404	125863
ssd_Linux full	58194	14548					
ssd_Linux clean	363017	90754					
gflash	54292	13573	4488	5242880	0	1248180	4093
ssd_FreeBSD full	59773	14943					
ssd_FreeBSD clean	148608	37151					

Итоги:

Задание №20 (QD=1)

Абсолютная скорость записи:

gflash 100% *(отложенная запись в режиме заполнения. Скорость записи превосходит даже скорость записи очищенного* ***ssd****, но это не ошибка. На момент окончания задания количество записанной в кэш информации (данные + метаданные) составляет 125% от количества записываемых данных, значит запись сразу производится на* ***ssd****. Разгадка данного феномена кроется в архитектуре построения* ***gflash*** *и задекларированной возможной потере до четырех последних запросов на запись при аварийном завершении работы. То есть* ***gflash*** *умеет "разгонять" скорость записи на малых очередях)*;

bcache 22% *(отложенная запись в режиме заполнения. Обращает на себя внимание очень малое количество записанных метаданных - около 2%. Это явно говорит о группировке в*

метаданных информации о большом количестве записываемых блоков и сразу возникает вопрос о надёжности при аварийном завершении работы);

flashcache 1% *(отложенная запись в форсированном режиме)*;

dm-cache 0,4% *(**dirty**=0).*

Задание №21 (QD=32)

Абсолютная скорость записи:

bcache 100% *(отложенная запись в режиме заполнения)*;

gflash 59% *(довольно показательный пример - вначале отложенная запись в режиме заполнения (в момент старта задания* **writeback** кэш заполнен приблизительно на *50%) скорость записи высокая. Затем при полном заполнении кэша начинается форсированный режим и скорость записи ограничивается скоростью случайной записи на **hdd**. В результате имеем довольно низкую среднюю скорость)*;

flashcache 6% *(отложенная запись в форсированном режиме)*;

dm-cache 1% *(сложно что-либо комментировать).*

Выводы:

Системы кэширования работают в различных режимах работы отложенной записи, но можно сделать некоторые выводы:

bcache и **gflash** традиционно показывают довольно высокую производительность;

flashcache из-за форсированного режима **writeback** кэша имеет низкую скорость записи;

dm-cache традиционно удивляет очень низкой скоростью записи.

На этом тест завершен.

Объем **dity** блоков на момент окончания теста можно сравнить в последней таблице. Это тот объём данных, который еще предстоит записать системам кэширования в основное хранилище.

Общие выводы

Данный тест не может претендовать на абсолютную объективность, так как существует еще много нерассмотренных нагрузок и режимов работы, но позволяет оценить сильные и слабые стороны систем кэширования.

gflash практически во всех тестах показывает лучшие результаты.

bcache незначительно уступает **gflash**. Это лучшая по прозводительности система кэширования из протестированных на платформе **Linux**. Если с надежностью работы все в порядке, то однозначный выбор для данной платформы.

flashcache сильной стороной является чтение малыми блоками из кэша при отсутствии фоновой отложенной записи, во всех остальных случаях производительность оставляет желать лучшего.

dm-cache самая несбалансированная система кэширования (с одной стороны имеет достаточно неплохую скорость чтения из кэша, с другой очень плохую производительность при помещении данных в кэш). Также не является в классическом понимании **writeback** кэшем и отличается странным (непредсказуемым) поведением.