



NETAPP TECHNICAL REPORT

# Flash Cache (PAM-II) and PAM: наилучшие методы использования

Paul Updike, NetApp

Январь 2010 | TR-3832

## **Коротко о главном:**

Performance Acceleration Module (PAM) и Flash Cache (ранее PAM II) это программное и аппаратное решение, которое позволяет вам улучшить показатели производительности контроллера системы хранения NetApp®. Это руководство в деталях описывает механизмы работы PAM и Flash Cache, и предоставляет необходимую информацию для полного понимания принципов их работы.

## Оглавление

1	Введение .....	3
1.1	Что такое Flash Cache и PAM .....	3
1.2	Как соотносятся производительность и емкость системы хранения? .....	3
1.3	Как это соотносится с экономией ресурсов .....	4
1.4	Как Flash Cache и PAM решают эту проблему .....	4
2	Как работают Flash Cache и PAM .....	4
2.1	Операция чтения с диска в Data ONTAP .....	4
2.2	Data ONTAP очищает место в системной памяти для данных.....	5
2.3	Сохранение нужных данных в кэширующем модуле .....	5
2.4	Чтение данных из кэширующего модуля.....	6
2.5	Ускорение чтения и максимальная производительность .....	6
2.6	Работа по каким протоколам ускоряется? .....	7
2.7	Рабочая нагрузка, которая не ускоряется.....	7
3	Режимы работы .....	7
3.1	Кэширование метаданных.....	7
3.2	Кэширование обычных пользовательских данных (по умолчанию) .....	8
3.3	Кэширование низкоприоритетных данных.....	9
3.4	Гибридный режим работы с FlexShare .....	10
4	Вопросы производительности.....	11
4.1	Период «прогрева» кэша (cache warm-up).....	11
4.2	Вопросы доступности данных .....	12
	Кластерный takeover .....	12
	Непрерывающие работу отказы .....	12
5	Взаимодействие с другими возможностями Data ONTAP .....	13
5.1	Взаимодействие с дедупликацией .....	13
5.2	Взаимодействие с FlexClone .....	13
5.3	Взаимодействие с FlexCache.....	13
6	Мониторинг нагрузки.....	13
7	Как наблюдать за производительностью Flash Cache или PAM? .....	13
8	Выводы .....	15

# 1 Введение

## 1.1 Что такое Flash Cache и PAM

Flash Cache (ранее носил имя PAM II) и собственно оригинальный Performance Acceleration Module (PAM) - это интеллектуальный кэш, соединяющий в себе программную и аппаратную компоненты внутри контроллера системы хранения NetApp. Они помогают ему работать быстрее, увеличивая производительность, и сокращая время доступа к данным без необходимости увеличивать количество дисковых «шпинделей» системы хранения. Flash Cache и PAM состоят из программной компоненты входящей в состав Data ONTAP® и аппаратной платы в слот расширения PCIe, на которой размещено по 16GB RAM (в случае PAM) или 256/512GB Flash (в случае Flash Cache) на каждом из таких модулей. Модули управляются специальной аппаратной программируемой логикой (FPGA - Field Programmable Gate Array). Несколько таких модулей могут быть скомбинированы в одной системе и работать как единое устройство. Такая технология позволяет достичь суб-миллисекундных величин задержек доступа к данным, которые раньше, при использовании жестких дисков, достигали 10 миллисекунд и более.

## 1.2 Как соотносятся производительность и емкость системы хранения?

Существуют различные по характеру и особенностям доступа к данным типы рабочих нагрузок системы хранения. Для большинства из них характерен случайный доступ к данным на чтение. Случайным доступом на чтение (*random reads*) принято называть процесс чтения не лежащих непрерывно на дисках системы хранения блоков данных.

Поскольку такое чтение производится над блоками, не расположенными рядом друг с другом, а более или менее равномерно разбросанными по пространству диска, то такой процесс чтения, как правило, занимает больше времени, чем в случае более локализованного, последовательного чтения. Этот тип доступа производит значительно больше операций поиска блока на диске и позиционирования на него головок дисков (*seeks*), что увеличивает время задержки доступа (*latency*) тем больше, чем более случаен по характеру процесс чтения. Нагрузка с большим процентным соотношением небольших, случайно считываемых блоков может быть наиболее сложной нагрузкой для любой системы хранения.

Традиционный путь обеспечить достаточную производительность в условиях случайной по характеру рабочей нагрузки, это увеличивать количество физических жестких дисков системы, так называемых «шпинделей» (*spindles*). «Больше дисков» означают больше головок чтения, и меньше шансов, что два чтения подряд придут на один и тот же диск, что вызовет значительную задержку в их обработке. Большое количество физических дисков позволяют системе хранения удовлетворять требованиям производительности при интенсивном, случайном по характеру чтении. Это не было ранее особенно заметной проблемой, поскольку значительное количество дисков на системе хранения, как правило, требовалось и для удовлетворения требованиям объемов дискового пространства. Таким образом, вы могли, покупая больше дисков, обеспечить и необходимую производительность, и нужное пространство хранения в один прием.

За последние несколько лет, емкость жестких дисков значительно выросла. Например, менее чем за 10 лет производители представили жесткие диски с интерфейсом FibreChannel емкостью сперва 18, затем 36, 72, 144 и, наконец, 300 и 450 гигабайт. Одновременно с этим экспоненциальным ростом емкости, показатели скорости позиционирования головок на жестком диске улучшились едва ли на 20%. Результатом стало заметное несоответствие между приростом производительности жестких дисков, и их емкости. Таким образом, вы могли бы записать больше

данных на меньшем количестве более емких, чем раньше, дисков, но вам приходится приобретать больше дисков («шпинделей») ради того, чтобы удовлетворить растущим требованиям производительности.

В результате, для того, чтобы обеспечить необходимую производительность, вам приходится покупать дополнительную, лишнюю для ваших задач дисковую емкость «в нагрузку» к дисковой производительности. А так как диски продолжают расти в емкости, ситуация только ухудшается. В то же самое время, забота об экономии ресурсов подталкивает пользователя снижать объемы потребляемой электроэнергии, ресурсов охлаждения, объемов и площадей, занимаемых в датацентре, как и вообще затрат на оборудование.

### **1.3 Как это соотносится с экономией ресурсов**

По мере того, как область IT-сферы в мире бизнеса продолжает расти, она сталкивается со все более строгими требованиями по расходу трех основных ресурсов датацентра: электроэнергией, занимаемым пространством и охлаждением. Как мы рассмотрели в главе 1.2, большинство задач уже не требует так много дисков как ранее, для удовлетворения требованиям объема хранения, но по-прежнему требует их значительное число для удовлетворения требованиям производительности.

Большое количество дисков создает значительные объемы непродуктивно используемого пространства, а также уровни потребления электричества и ресурсов охлаждения. Эта ситуация подтверждает необходимость поиска решения для того, чтобы удовлетворить требования уровней производительности системы и объемов ее хранения отдельно.

### **1.4 Как Flash Cache и PAM решают эту проблему**

Flash Cache (PAM II) и PAM заменяют операции чтения с диска на доступ к расширенному объему кэша, содержащемуся в одном или нескольких установленных в контроллер системы хранения аппаратных модулях. Производительность увеличивается пропорционально тому, сколько операций чтения с диска заменяется на операции чтения из кэша. В этом документе мы рассмотрим различные варианты рабочих нагрузок, и того, как они ускоряются при использовании Flash Cache и PAM, а также о том, как выбрать и сконфигурировать наилучший для вашей задачи используемый режим, и как проанализировать эффект от работы модулей.

## **2 Как работают Flash Cache и PAM**

Flash Cache и PAM добавляют новые возможности в Data ONTAP. С их использованием изменяется поведение системы хранения в алгоритме чтения данных по запросу пользовательской хост-системы. Давайте рассмотрим то, как данные проходят по системе хранения, и как меняется ситуация с добавлением Performance Acceleration Module.

### **2.1 Операция чтения с диска в Data ONTAP**

До появления в Data ONTAP устройств Flash Cache и PAM, когда клиенту или хосту требовались данные, и запрошенные данные в этот момент не находились в памяти системного кэша, то начиналась процедура считывания этих данных с дисков.



Рис. 1) Процесс чтения до появления Flash Cache или PAM.

## 2.2 Data ONTAP очищает место в системной памяти для данных

Когда в памяти нужно больше свободного места, Data ONTAP анализирует уже хранимые в ней блоки, и ищет данные с наименьшим приоритетом, которые можно очистить и освободить место под более важные данные. В зависимости от объемов нагрузки, данные сохраняются в системной памяти от секунд до часов; но в любом случае она должна очищаться.



Рис. 2) Очистка памяти до появления Flash Cache или PAM.

## 2.3 Сохранение нужных данных в кэширующем модуле

При добавлении в систему Performance Acceleration Module, данные, которые раньше просто очищались, перемещаются в память этого модуля. Данные, как и ранее, всегда прочитываются в системный кэш в основной памяти, и после этого, когда наступает время ее очистки для более срочных данных, переносятся в память модуля.

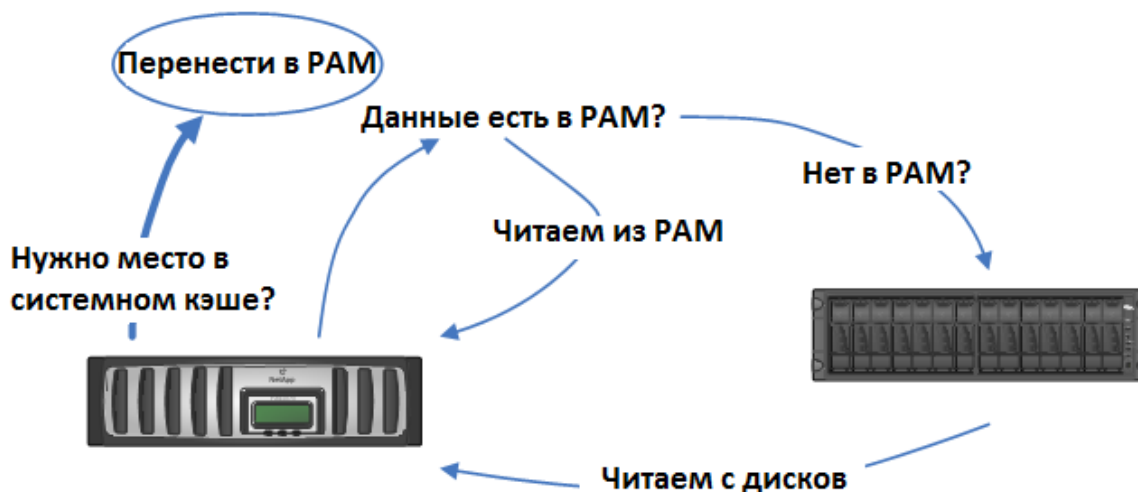


Рис. 3) Данные сохранены в Flash Cache или РАМ.

## 2.4 Чтение данных из кэширующего модуля

Теперь данные записаны и сохраняются в модуле, Data ONTAP может получить их оттуда, когда эти данные потребуются в следующий раз. Когда это случится, доступ к этим данным может быть получен гораздо быстрее, чем при считывании их непосредственно с дисков. Таким образом и достигается ускорение работы.

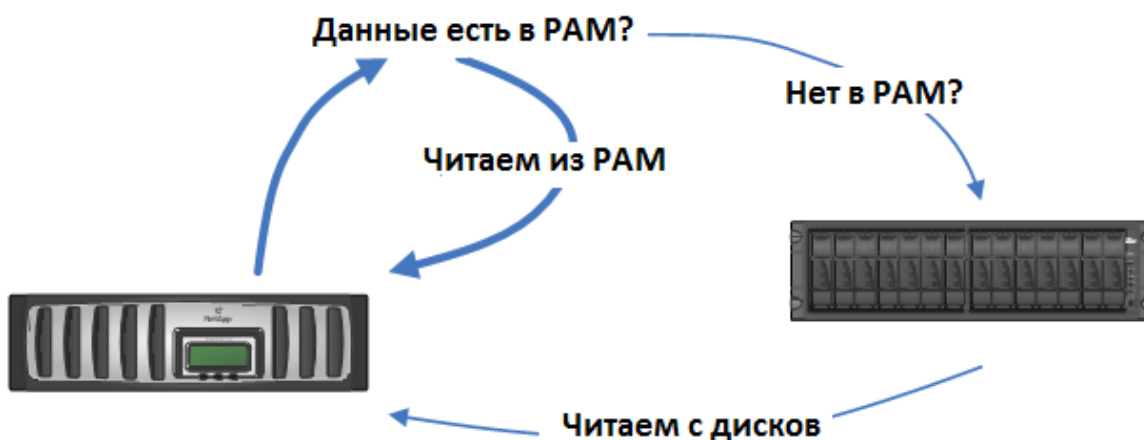


Рис. 4) Чтение из Flash Cache или РАМ в 10 раз быстрее, чем с диска.

## 2.5 Ускорение чтения и максимальная производительность

Важная особенность работы Flash Cache и РАМ состоит в том, что это кэширование только операций чтения.

Flash Cache и РАМ разработаны для ускорения производительности только операций чтения. Операции записи на диск в Data ONTAP уже максимально оптимизированы, и использование кэша на запись для них уже не ускорит их сколь-нибудь заметно. Однако существует возможность несколько улучшить производительность на запись, так как использование кэша на чтение частично разгружает диски от операций, данные которых попали в кэш, и не завершаются обращением к диску, поэтому некоторое количество дискового времени высвобождается и может быть потрачено на улучшение характеристик записи.

## 2.6 Работа по каким протоколам ускоряется?

Так как Flash Cache и PAM работают уровнем ниже всех протоколов доступа к данным, то они ускоряют доступ к данным по любым используемым в системе хранения протоколам. Как будет рассмотрено в следующей главе, степень ускорения работы зависит от характера нагрузки, а не от используемого протокола.

## 2.7 Рабочая нагрузка, которая не ускоряется

Некоторые типы рабочей нагрузки могут не испытывать заметного ускорения, например:

- Если нагрузка состоит в основном из операций записи
- Если нагрузка состоит в основном из длинных последовательных чтений

В обоих этих случаях эффект от использования модулей будет сравнительно невелик. Записи не ускоряются непосредственно, хотя и может наблюдаться некоторый небольшой положительный эффект, связанный с тем, что для выполнения операций записи теперь необходимо меньше операций чтения с дисков. Аналогично, длинные последовательные чтения, скорее всего, не будут использовать повторно уже прочитанные и сохраненные в кэше модуля данные. Поэтому для таких типов нагрузки эффективность работы модулей будет сравнительно невелика.

Хотя на любой из этих типов нагрузок возможно появление положительного эффекта от использования Flash Cache и PAM, он не будет так значителен, как на более подходящих для их использования.

Впрочем, если размер активного набора данных невелик, и он помещается целиком в Flash Cache или PAM то это может быть хорошим исключением из написанного выше. Более подробное рассмотрение такого варианта приводится в главе 3.3.

## 3 Режимы работы

Существует три основных режима работы Flash Cache и PAM. Они обеспечивают возможность настройки поведения механизма кэширования в модуле, чтобы наилучшим образом соответствовать требованиям решаемой системой хранения рабочей задачи.

### 3.1 Кэширование метаданных

• **Metadata mode:** Первый способ улучшить производительность с помощью Flash Cache или PAM работает для задач с большим объемом считываемых метаданных. Некоторые из задач требуют доступа к метаданным, прежде чем будет обслужено обращение к данным приложения. В таких условиях, помещение метаданных в кэш чтения позволяет организовать к ним быстрый доступ с незначительными задержками; ускорение обращения к метаданным естественным образом ускоряет и обращение к собственно данным. Это режим более применим для PAM, так как Flash Cache (PAM II), с гораздо большим объемом памяти, может более эффективно использоваться и в рассмотренных далее режимах с сохранением не только метаданных, но и собственно пользовательских данных.

Для использования этого режима должны быть выставлены следующие системные опции:

```
flexscale.enable          on
flexscale.lopri_blocks    off
flexscale.normal_data_blocks off
```

## Кэширование метаданных

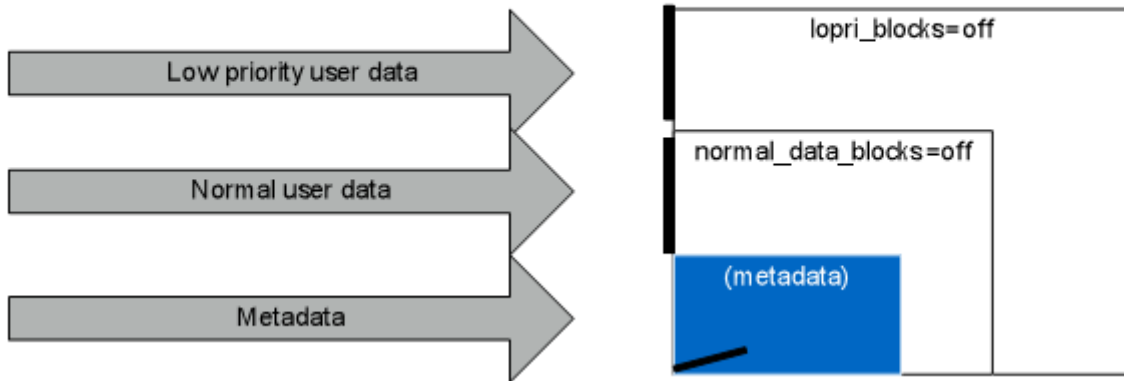


Рис. 5) Только метаданные; обычные пользовательские данные, а также низкоприоритетные данные не попадают в память модуля.

### 3.2 Кэширование обычных пользовательских данных (по умолчанию)

- **Default mode:** Наиболее простой способ увеличения производительности состоит в непосредственном кэшировании данных, к которым производится доступ на дисках. В случае, если запрошенные данные уже находятся в кэше, вместо того, чтобы читать данные с диска, что сопряжено со сравнительно большими задержками доступа, данные прочитываются из памяти Acceleration Module. Этот режим установлен по умолчанию. Установки опций для этого режима выглядят следующим образом:

```
flexscale.enable          on
flexscale.lopri_blocks    off
flexscale.normal_data_blocks on
```

## Кэширование обычных пользовательских данных

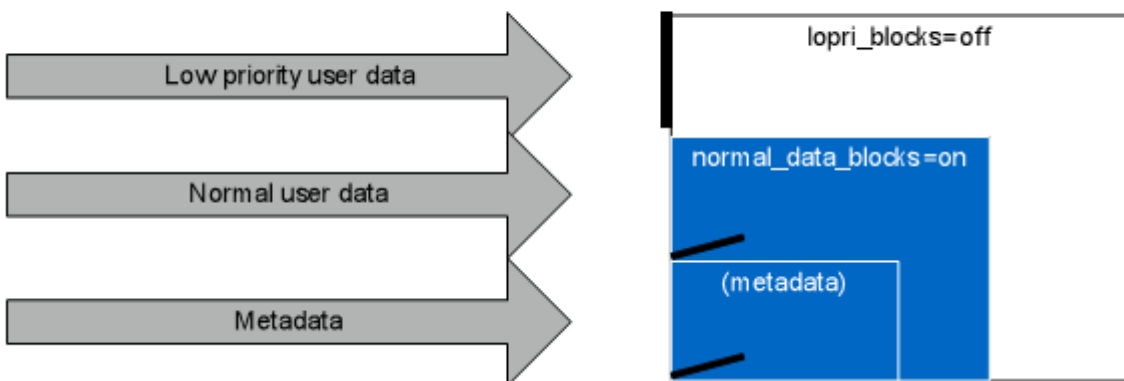


Figure 6) Кэшируются как metadata, так и обычные данные пользователя; low-priority user data не попадают в модуль.



### 3.3 Кэширование низкоприоритетных данных

- **Low-priority blocks mode:** Третий способ использования кэширования в Flash Cache и PAM это сохранение в их памяти данных, которые обычно не сохраняются в системном кэше вовсе. Они называются *low-priority buffers* и бывают таких типов:

**Записываемые данные:** В обычном режиме записи буферизуются в RAM и логируются в NVRAM, затем, когда эти блоки данных оказываются сброшены на диск, они очищаются в NVRAM и остаются с наиболее низким приоритетом в системной RAM, чтобы излишне не загромождать ее. Другими словами, недавно записанные данные являются первыми кандидатами на вытеснение из кэша системной памяти контроллера. В ряде случаев использования, для недавно записанных данных может потребоваться чтение сразу после записи. Для такого типа нагрузки Performance Acceleration Module может улучшить производительность, кэшируя записанные блоки в памяти PAM или Flash Cache, вместо обычной системной процедуры сброса блоков на диск, очистки кэша, и нового чтения тех же данных с диска.

**Длинные последовательные считанные блоки:** Длинные, последовательно считанные с дисков блоки могут заполнить всю системную память большим объемом данных, доступ к которым, в большинстве случаев, производится только раз. По умолчанию, Data ONTAP не сохраняет такие данные в кэше системной памяти, предпочитая занимать его пространство под данные, повторный доступ к которым может возникнуть с большей вероятностью. Большой объем данных, которым располагает Performance Acceleration Module, например такой, как Flash Cache (PAM II), позволяет сохранять длинные последовательно считанные фрагменты данных в своей памяти без значительного негативного влияния на кэширование других данных. Если для этих блоков возникнет необходимость повторного считывания, то вы увидите заметный прирост производительности при использовании Performance Acceleration Module в этом режиме.

Системные опции для этого режима должны быть установлены в такой комбинации:

```
flexscale.enable          on
flexscale.lopri_blocks    on
flexscale.normal_data_blocks on
```

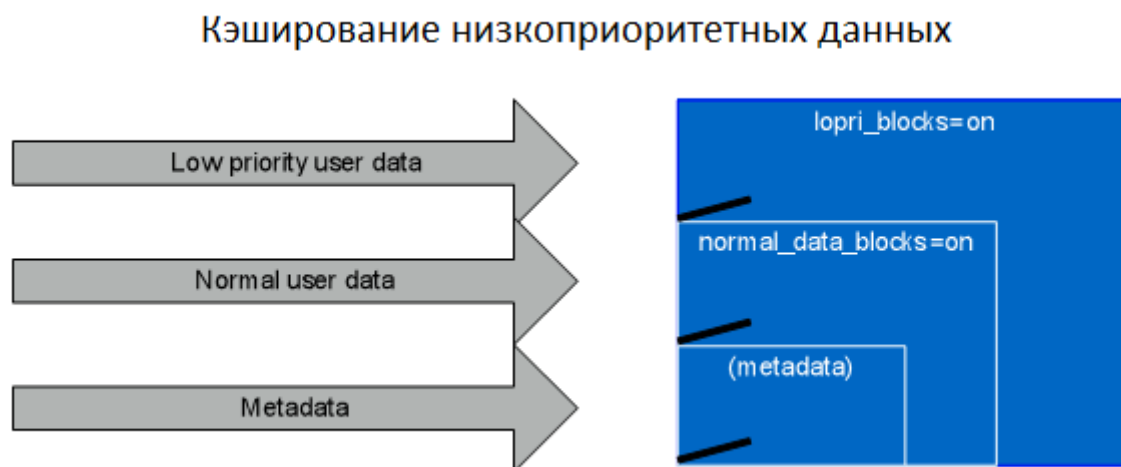


Figure 7) В памяти модуля размещаются метаданные, обычные пользовательские данные, и низкоприоритетные данные.

### 3.4 Гибридный режим работы с FlexShare

FlexShare® обеспечивает функциональность *quality-of-service* в контроллере системы хранения NetApp. В комбинации с интеллектуальным кэшированием в Performance Acceleration Module он позволяет вам устанавливать политики кэширования на определенные тома, добавлять или убирать опции из общесистемных настроек с помощью возможностей FlexScale™.

Например, вы определили, что наилучший для вашей задачи режим кэширования это *metadata only*, но у вас есть один том, для которого вы хотите использовать режим кэширования *normal data*. Установите общесистемную опцию FlexScale в *normal*. Затем, в командной строке, включите FlexShare обычным образом, или только для кэширующих компонентов.

Для того, чтобы включить FlexShare введите в командной строке:

```
> priority on
```

Если вы хотите управлять через FlexShare только кэшированием (если вы хотите использовать все возможности FlexShare, не только для задач кэширования, то пропустите следующий шаг), то также ведите команду:

```
> priority set enabled_components=cache
```

Когда указанный компонент включен, у вас будет две опции для работы с режимами кэширования PAM:

```
> priority set volume Vol1 cache=keep
```

Эта установка меняет поведение кэша на более агрессивное сохранение в кэше данных, относящихся к Vol1. Когда опция включена, и модуль находится в *normal* или *low-priority mode*, она не дает эффекта. Но, как показано на следующем рисунке, когда система работает в режиме *metadata mode* и том имеет значение «*cache=keep*», то поведение кэша меняется таким образом, что кэш продолжает хранить метаданные для всех томов, но, дополнительно для Vol1, он будет сохранять и обычные данные.

#### Кэширование метаданных, и кэширование всех данных для одного тома Vol1

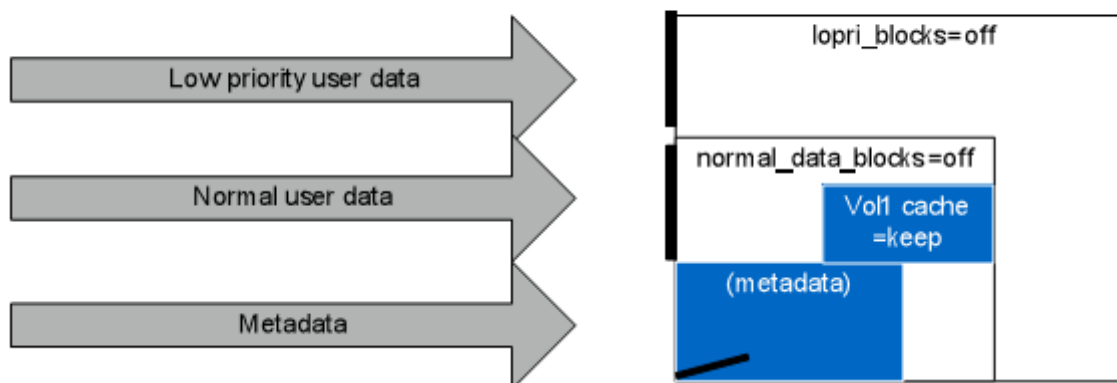


Рис. 8) Даже несмотря на то, что общие настройки контроллера установлены в *metadata mode*, настройка FlexShare для Vol1 «*cache=keep*» позволяет использовать для него *normal mode*.

Аналогично, если общесистемный режим для модуля установлен в *normal* или *low-priority mode* и нам необходимо, например для тома Vol2, снизить этот уровень до *metadata mode*, то следует воспользоваться опцией «*cache=reuse*»:

> priority set volume Vol2 cache=reuse.

### Полное кэширование для всех томов, за исключением тома Vol2 (только метаданные)

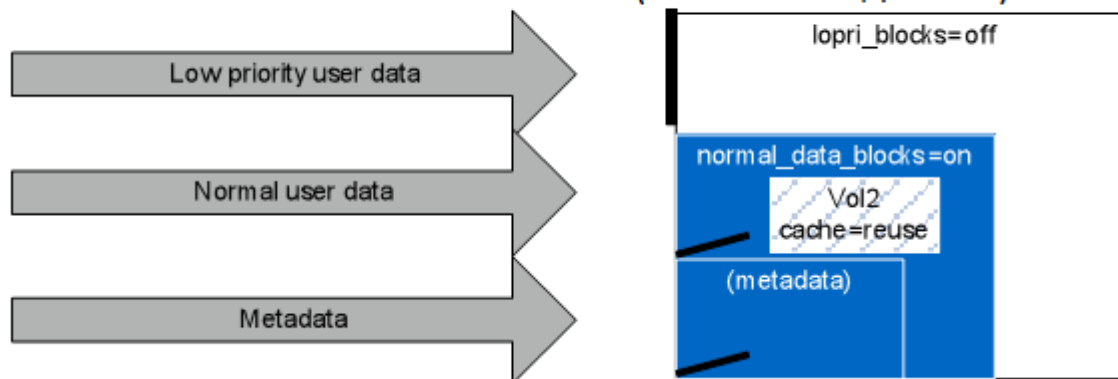


Рис. 9) Даже несмотря на то, что общие настройки контроллера установлены в *normal mode*, настройка FlexShare для Vol2 «cache=reuse» позволяет использовать для него *metadata mode*.

Глава в **System Administration Guide** под названием **System Performance and Resources** содержит подробности об администрировании FlexShare. Также смотрите для подробностей документ **TR-3459** в котором рассматривается FlexShare. Документ можно найти в разделе Library на вебсайте NetApp.

## 4 Вопросы производительности

### 4.1 Период «прогрева» кэша (cache warm-up)

С объемами, равными 256GB или 512GB (а также кратными им при установке нескольких модулей), заполнение кэша данными в модулях Flash Cache (РАМ II) может занять довольно значительное время. Для того, чтобы понять разницу между заполненным, или, как иногда говорят, «прогретым» (warming up) кэшем, и пустым кэшем, следует разобраться с тем, как именно система его использует. После начальной загрузки контроллера, или после кластерной операции *giveback*, модули Flash Cache не содержат данных, которые могли бы быть использованы для ускорения работы. В результате, все чтения производятся непосредственно с дисковой подсистемы. По мере того, как данные проходят через контроллер, кэш данных наполняется. В какой-то момент времени система достигает точки, в которой кэш наполнен доверху, в этом случае возможно два варианта: или новые данные поступают в него только в случае замены ими «инвалидированных» (более не являющихся действительными и актуальными) данных, например которые были перезаписаны на дисках, или же новые поступающие данные заменяют (вытесняют) валидные, но наиболее давно лежащие в кэше блоки данных.

Если мы примем второй сценарий, то кэш будет работать со 100% эффективностью в момент, когда он целиком заполнится. Для оценки того, как много займет времени процесс заполнения, мы можем воспользоваться простой формулой:

**(размер кэша) / (пропускная способность чтения в секунду) = время заполнения кэша**

Например:

**512GB/100MB в секунду = 5120 секунд = 85 минут**

На первый взгляд этот сценарий может выглядеть сложным, но мы упустили из виду поведение кэша и кэшированных данных с течением времени. Когда кэш наполняется и при этом используется системой, то чаще попадают в повторное использование блоки данных, недавно оказавшиеся в кэше. Когда кэш окончательно заполнен, то относительный эффект большего объема данных снижается. По этой причине эффект кэширования наиболее значителен на первых 10% заполненного объема, и имеет наименьший эффект на последних 10% заполнения кэша. Конечно, мы хотим, чтобы кэш был достаточно велик, чтобы в него поместились максимально возможные из используемых нами наборов данных, но вы начинаете пожинать плоды использования кэша с первого повторно запрошенного с дисков блока, который уже попал в кэш.

Когда мы оцениваем время «прогрева» кэша, нам следует принять во внимание следующие параметры:

- Каков поток данных на чтение для рабочей нагрузки?
- Каков объем рабочего набора данных?
- Как много блоков данных из рабочего объема считывается повторно?

Все эти факторы вместе влияют на время «прогрева» кэша. Величина потока считываемых данных, как описано выше, напрямую влияет на объем данных, которые попадают в кэш. Рабочий объем данных может быть больше или меньше, чем действующий объем закэшированных данных; на практике многие задачи имеют в своем рабочем объеме данных так называемые *hot spots*, или «горячие точки», с очень активными участками сравнительно небольших объемов. Ну и, наконец, если рабочий объем данных имеет обширные фрагменты повторно считываемых участков, то процесс «прогрева» даст более заметный и быстрый эффект, чем если бы эти данные перечитывались реже.

## 4.2 Вопросы доступности данных

### Кластерный takeover

Когда вы планируете использовать Flash Cache или RAM в кластерной системе, помните о следующих особенностях:

1. NetApp рекомендует симметричное использование модулей в контроллерах HA-кластера. Если на одном из контроллеров установлено три модуля, то мы рекомендуем на его кластерном партнере установить такое же количество. Это обеспечит достаточную производительность в случае кластерного *takeover*.
2. В случае *takeover*, содержимое кэша будет реинициализировано и на контроллере начнется новый период «прогрева» кэша (*cache warm-up period*). Это же произойдет и в случае обратного переключения (*giveback*).

### Непрерывающие работу отказы

Flash Cache и RAM созданы с возможностью в случае ошибки или отказа не останавливать работу системы хранения их использующих. В случае аппаратной проблемы или отказа модуль переводится в *off-line* и контроллер системы хранения продолжает работать без их использования. Это позволяет сохранять доступность данных в случае отказа или аппаратной ошибки. Однако для физической замены неисправных модулей может потребоваться запланированный *downtime*.

## 5 Взаимодействие с другими возможностями Data ONTAP

Flash Cache и PAM работают в Data ONTAP на уровне ниже программной функциональности других возможностей OS. По этой причине модули могут ускорять их работу тем же образом, как и для доступа к данным по протоколам NAS или SAN.

### 5.1 Взаимодействие с дедупликацией

Дедуплицированные блоки могут кэшироваться в модулях PAM и Flash Cache как и любые другие блоки. Только одна копия дедуплицированных блоков будет попадать в кэш, точно также как они хранятся на диске.

### 5.2 Взаимодействие с FlexClone

Flash Cache и PAM полностью совместимы с *flexible clones*. Если доступ к данным клона происходит в соответствии с паттерном доступа рассмотренном в главе 2.5, то такой доступ будет ускорен точно также, как и с обычными данными. Как и в случае дедупликации, только одна копия блока данных тома FlexClone будет находиться в модуле, вне зависимости от того, сколько клонов в системе совместно используют этот блок.

### 5.3 Взаимодействие с FlexCache

Flash Cache и PAM работают с технологией NetApp FlexCache®. Работая на уровне ниже уровня протоколов, модули могут хранить блоки для томов FlexCache также как и для любых других томов.

## 6 Мониторинг нагрузки

## 7 Как наблюдать за производительностью Flash Cache или PAM?

Flash Cache и PAM используют архитектуру счетчиков производительности Data ONTAP. Имя счетчика, отвечающего за кэш чтения:

```
ext_cache_obj
```

Для просмотра информации об этом объекте используйте команду:

```
>stats show ext_cache_obj
ext_cache_obj:ec0:type:IOMEM-FLASH
ext_cache_obj:ec0:blocks:402653184
ext_cache_obj:ec0:size:1560
ext_cache_obj:ec0:usage:1%
ext_cache_obj:ec0:accesses:0
ext_cache_obj:ec0:disk_reads_replaced:0/s
ext_cache_obj:ec0:hit:0/s
ext_cache_obj:ec0:hit_normal_lev0:0/s
ext_cache_obj:ec0:hit_metadata_file:0/s
ext_cache_obj:ec0:hit_directory:0/s
ext_cache_obj:ec0:hit_indirect:0/s
ext_cache_obj:ec0:total_metadata_hits:0/s
ext_cache_obj:ec0:miss:0/s
ext_cache_obj:ec0:miss_metadata_file:0/s
ext_cache_obj:ec0:miss_directory:0/s
ext_cache_obj:ec0:miss_indirect:0/s
ext_cache_obj:ec0:hit_percent:0%
ext_cache_obj:ec0:inserts:0/s
ext_cache_obj:ec0:inserts_normal_lev0:0/s
```

```

ext_cache_obj:ec0:inserts_metadata_file:0/s
ext_cache_obj:ec0:inserts_directory:0/s
ext_cache_obj:ec0:inserts_indirect:0/s
ext_cache_obj:ec0:evicts:0/s
ext_cache_obj:ec0:evicts_ref:0/s
ext_cache_obj:ec0:readio_solitary:0/s
ext_cache_obj:ec0:readio_chains:0/s
ext_cache_obj:ec0:readio_blocks:0/s
ext_cache_obj:ec0:readio_max_in_flight:236
ext_cache_obj:ec0:readio_avg_chainlength:0
ext_cache_obj:ec0:readio_avg_latency:0ms
ext_cache_obj:ec0:writeio_solitary:0/s
ext_cache_obj:ec0:writeio_chains:0/s
ext_cache_obj:ec0:writeio_blocks:0/s
ext_cache_obj:ec0:writeio_max_in_flight:67
ext_cache_obj:ec0:writeio_avg_chainlength:0
ext_cache_obj:ec0:writeio_avg_latency:0ms
ext_cache_obj:ec0:invalidates:0/s

```

Вывод дает вам «мгновенный снимок» состояния счетчиков данных. Для наблюдения за динамикой NetApp рекомендует использовать итеративный вывод с пресетом для счетчиков Flash Cache/PAM. Это покажет вам вывод результата каждые 5 секунд, для посекундных показателей. Команда выглядит, например, так:

```
>stats show -p flexscale-access
```

Вывод будет иметь следующую форму:

```

Cache                               Reads      Writes      Disk Reads
Usage                               Chain      Chain      Replaced
%      /s      /s      /s      %      /s      /s      /s      /s      /s      /s      /s
0      0      0      0      0      0      0      0      0      0      0      0

```

Ниже приведены определения счетчиков:

Счетчик	Значение
<b>Cache Usage</b>	Сколько места занято в кэше модуля (модулей)
<b>Hit</b>	Число 4кВ-блоков диска, которые считаны из кэша в секунду
<b>Meta</b>	Число 4кВ-блоков метаданных, которые считаны из кэша в секунду
<b>Miss</b>	Число 4кВ-блоков диска, которых не оказалось в кэше
<b>Hit %</b>	Соотношение суммарного hit/miss
<b>Evict</b>	Число 4кВ-блоков диска, которые были вытеснены из кэша в секунду
<b>Inval</b>	Число 4кВ-блоков диска, которые были помечены как <i>invalidated</i> в секунду
<b>Insert</b>	Число 4кВ-блоков диска, которые были вставлены ( <i>inserted</i> ) в кэш в секунду
<b>Reads Chain</b>	Число считанных цепочек вода-вывода в секунду
<b>Reads Blocks</b>	Число считанных 4кВ-блоков диска в секунду
<b>Writes Chain</b>	Число записанных цепочек вода-вывода в секунду
<b>Writes Blocks</b>	Число записанных 4кВ-блоков диска в секунду
<b>Disk Reads Replaced</b>	Число операций чтения в секунду, которые вместо чтения с диска были произведены из кэша

Основные показатели в выводе счетчиков это **hit rate** и число **disk reads replaced**.

## **8 Выводы**

Flash Cache (РАМ II) и РАМ обеспечивают ускорение производительности для широкого спектра рабочих нагрузок системы хранения. Процесс конфигурирования для большинства сценариев использования прост и понятен, и предоставляет достаточно гибкости для тонкой настройки при использовании его совместно с FlexShare. Оба рассмотренных продукта обладают способностью ускорять работу системы хранения по всем доступным протоколам, и обеспечивают преимущества при использовании большинства возможностей Data ONTAP.