

Сжатие, шифрование и не только

Меняем поведение и гарантии распределенной
базы данных

Сжатие, шифрование и не только

Меняем поведение и гарантии распределенной
базы данных

Сжатие, шифрование
и не только

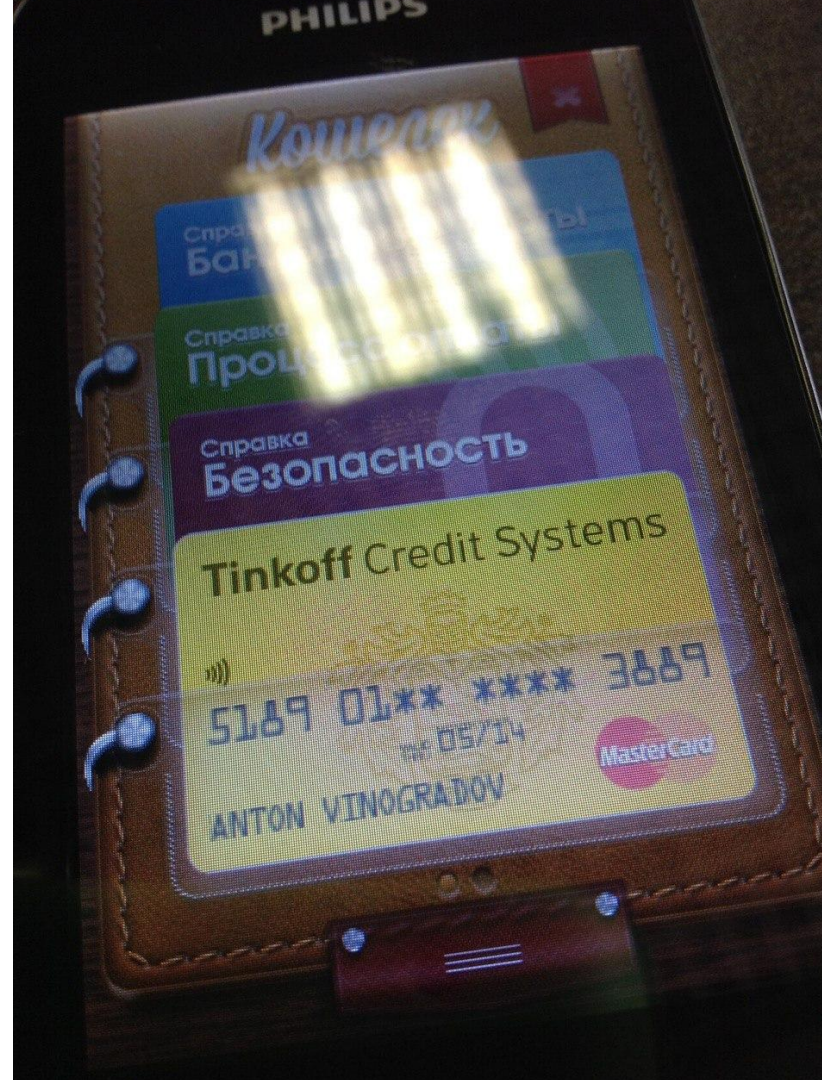
Меняем поведение и гарантии распределенной базы данных

Обо мне

- **Apache Ignite** (с 2015),
PMC member and Committer
- **Platform V DataGrid** (с 2018),
SberTech, Chief System Architect

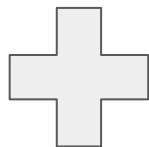
Обо мне

- **Apache Ignite** (с 2015),
PMC member and Committer
- **Platform V DataGrid** (с 2018),
SberTech, Chief System Architect
- **Кошелёк** (2012 - 2014),
Lead Java Developer



Добавление данных

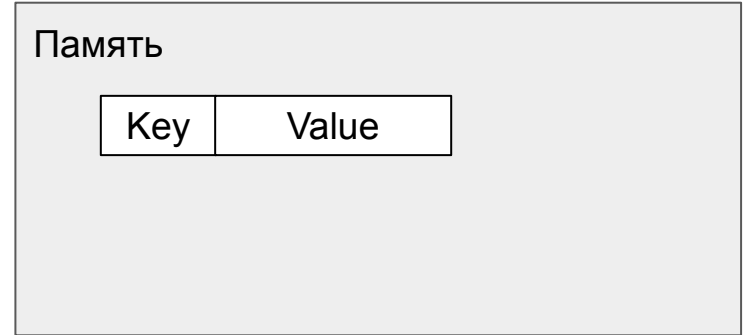
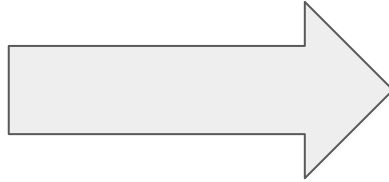
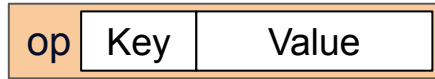
Key7	Value7
Key1	Value2
Key43	Value43



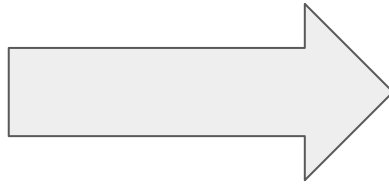
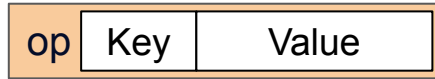
Key42	Value0
-------	--------



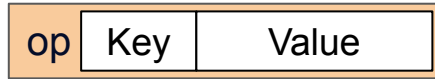
Запись в память



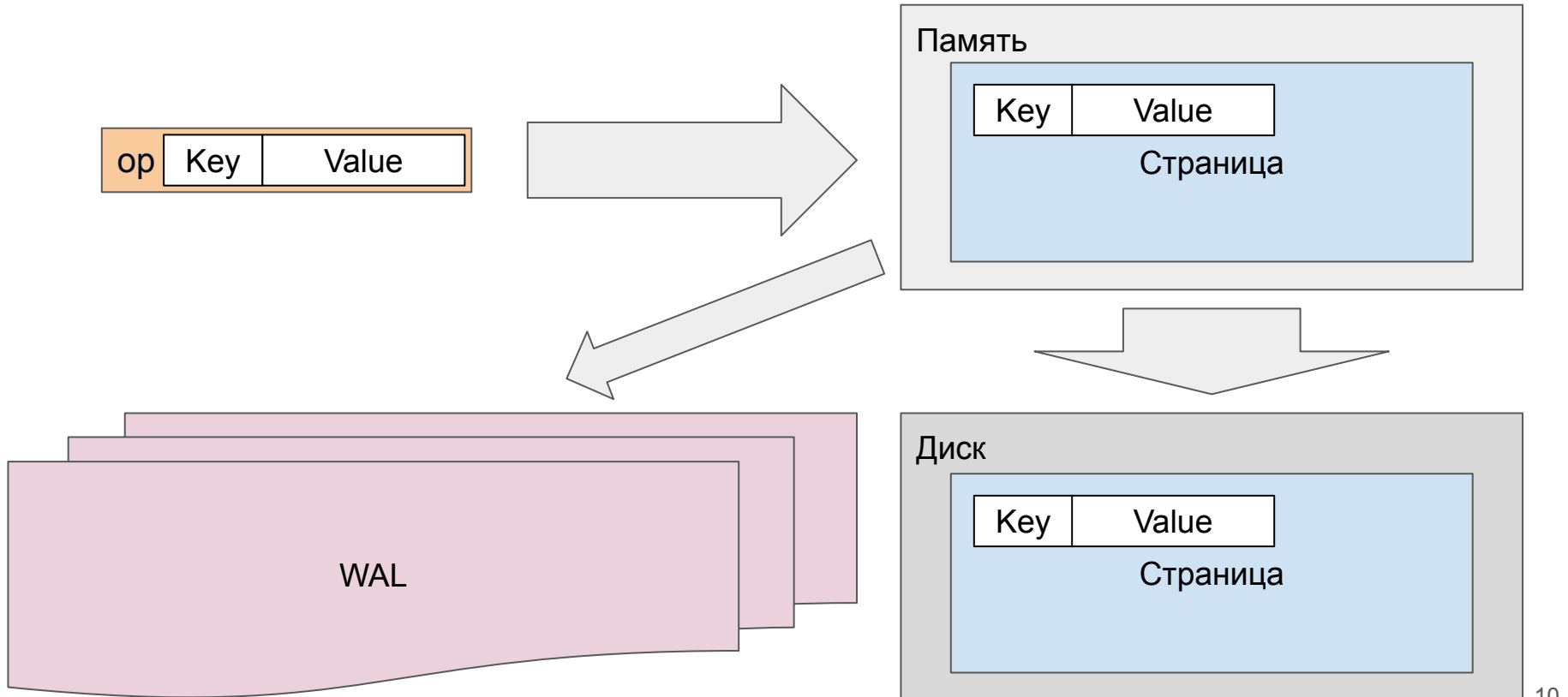
Запись в страницу



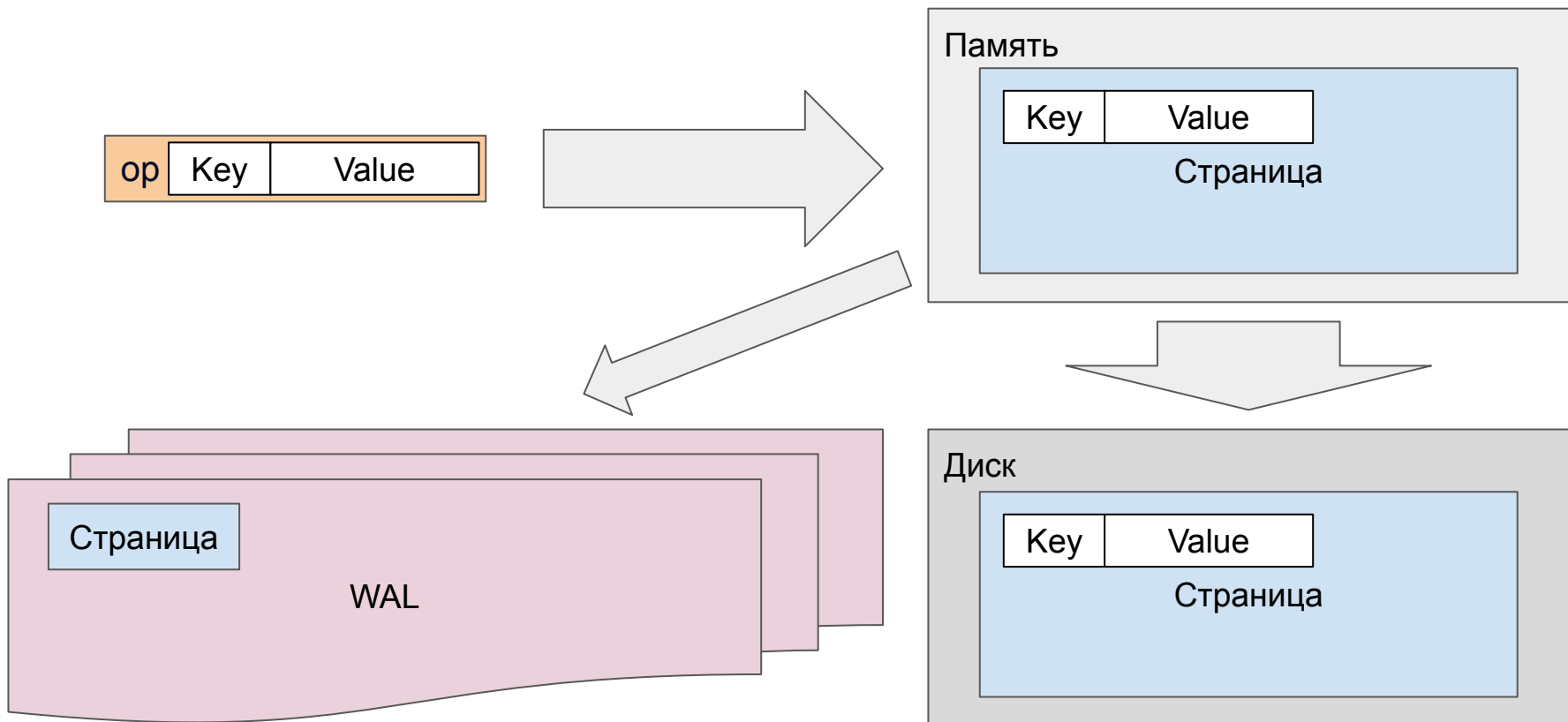
Запись на диск



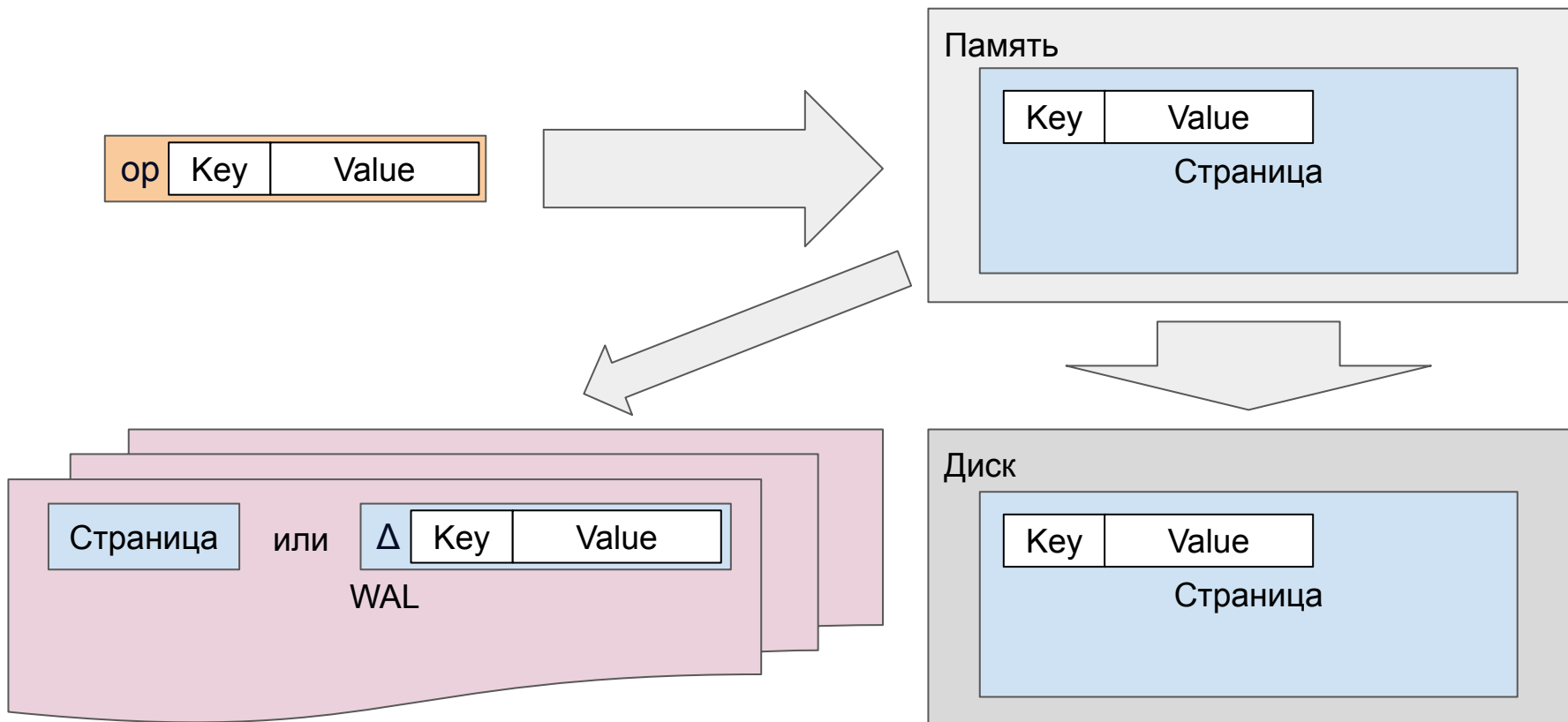
Запись в WAL



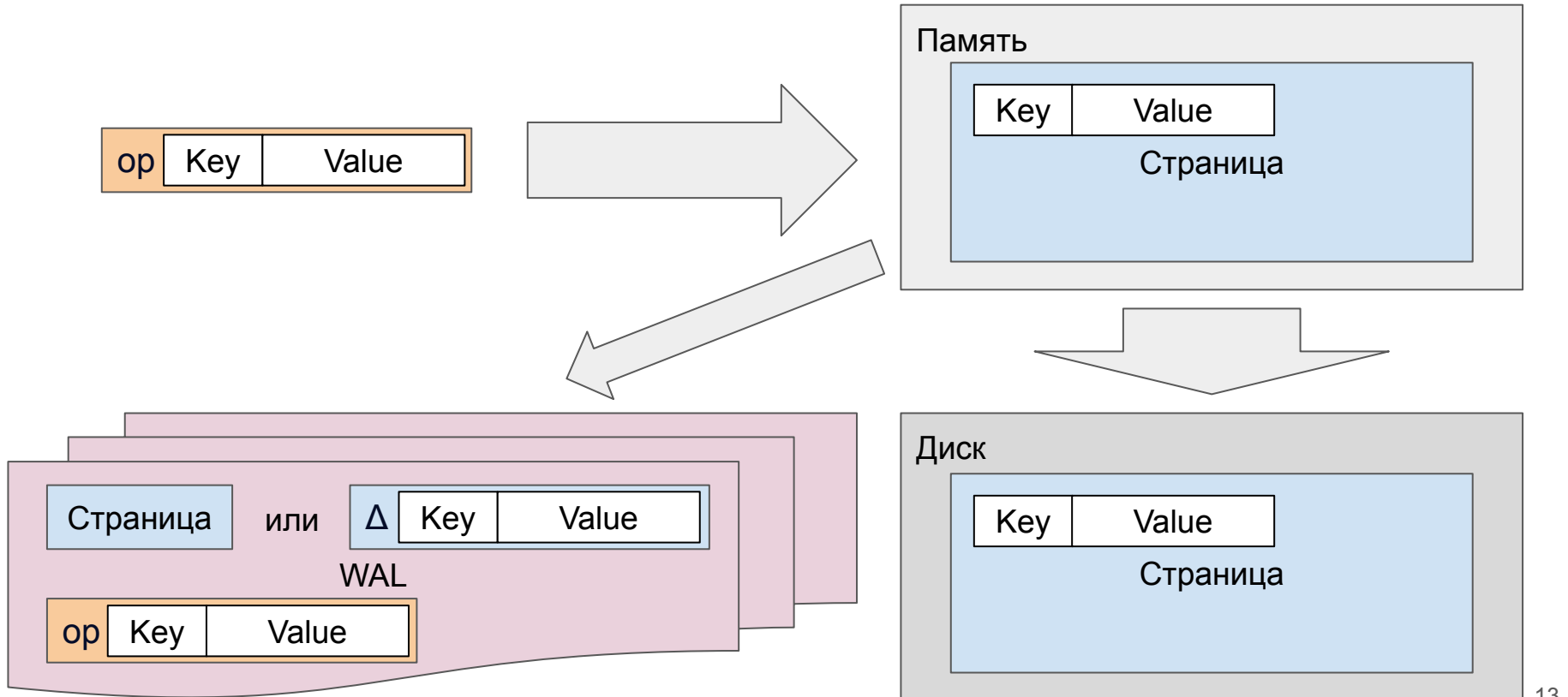
Гарантии записи на диск



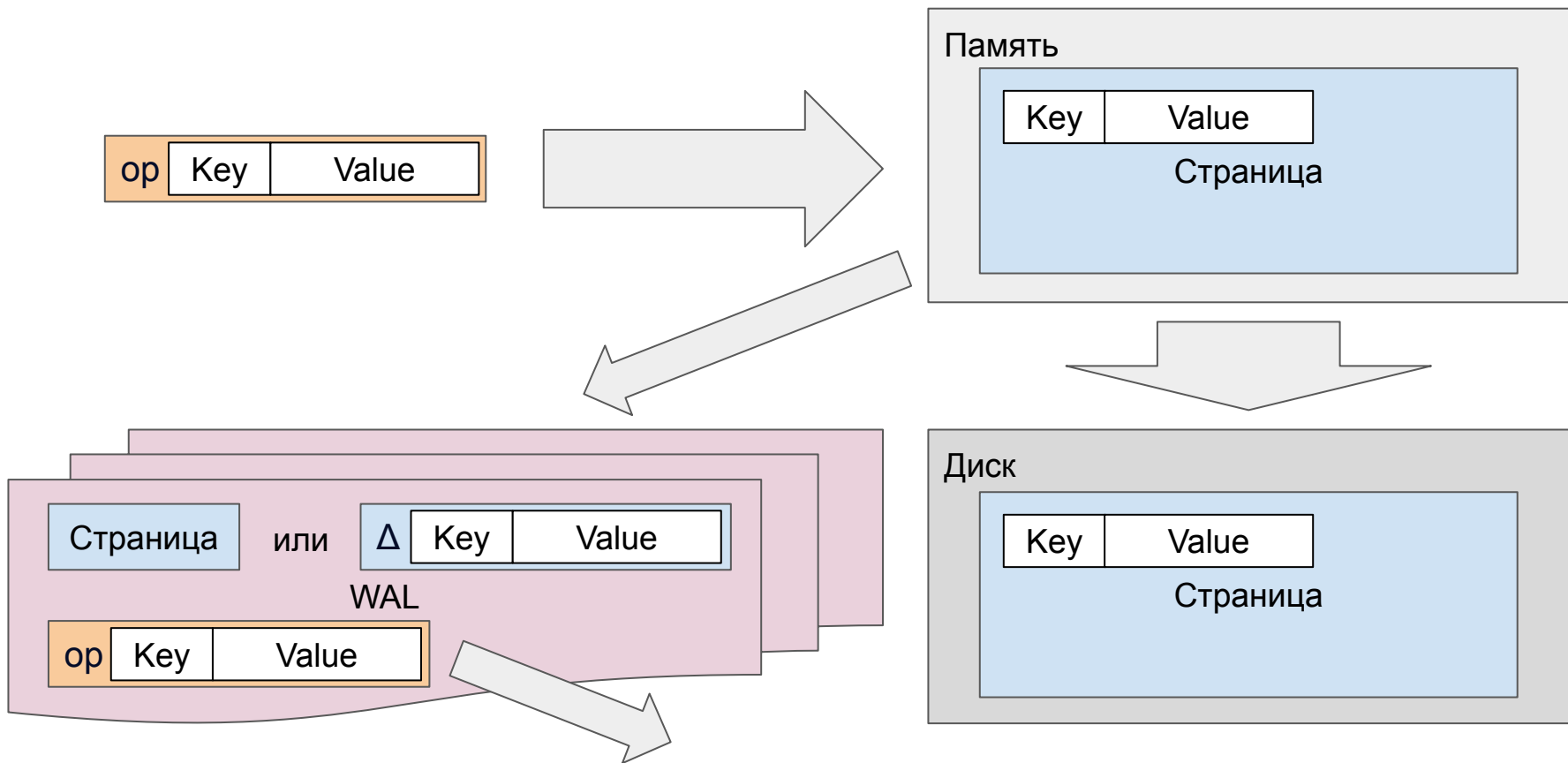
Гарантии записи на диск



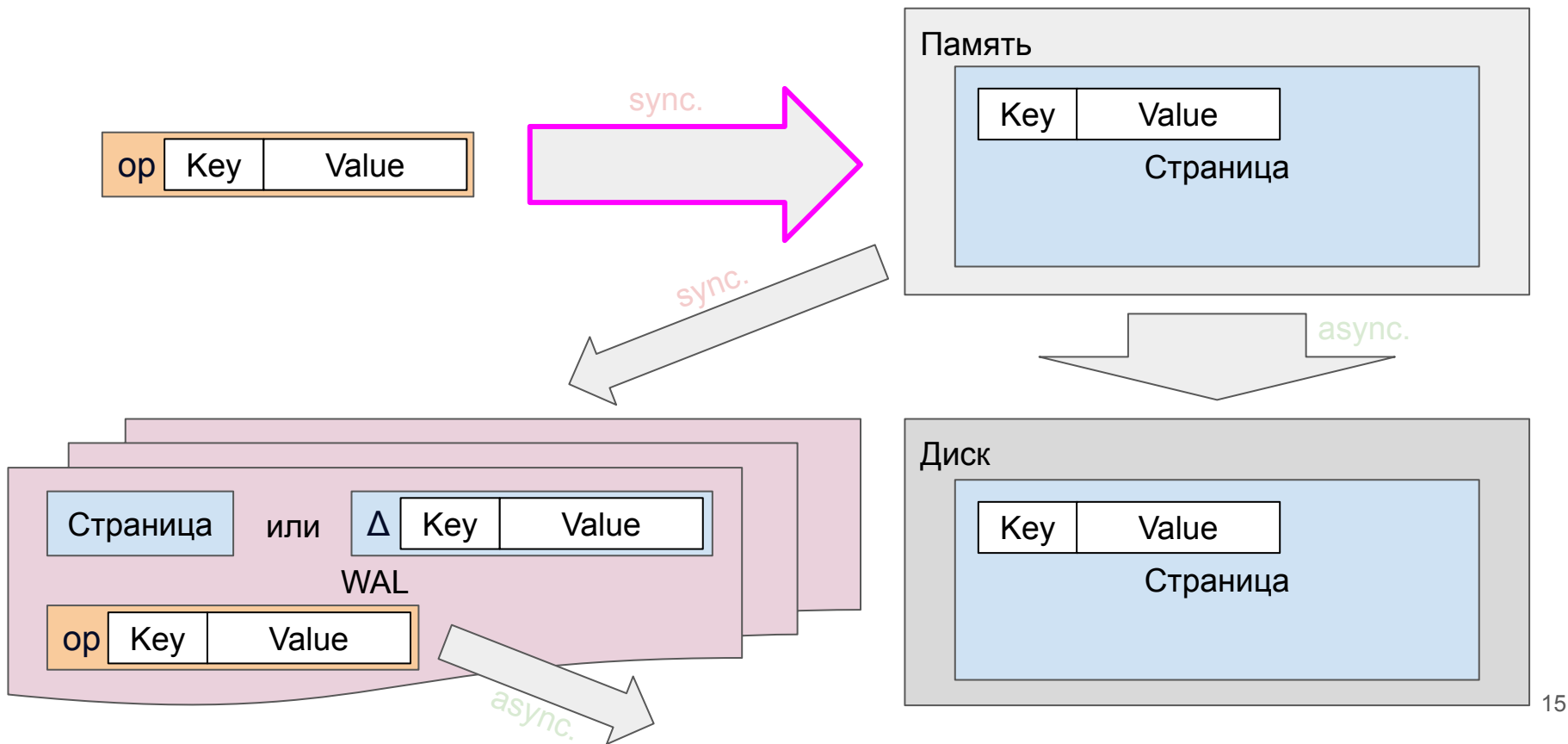
Гарантии восстановления кластера



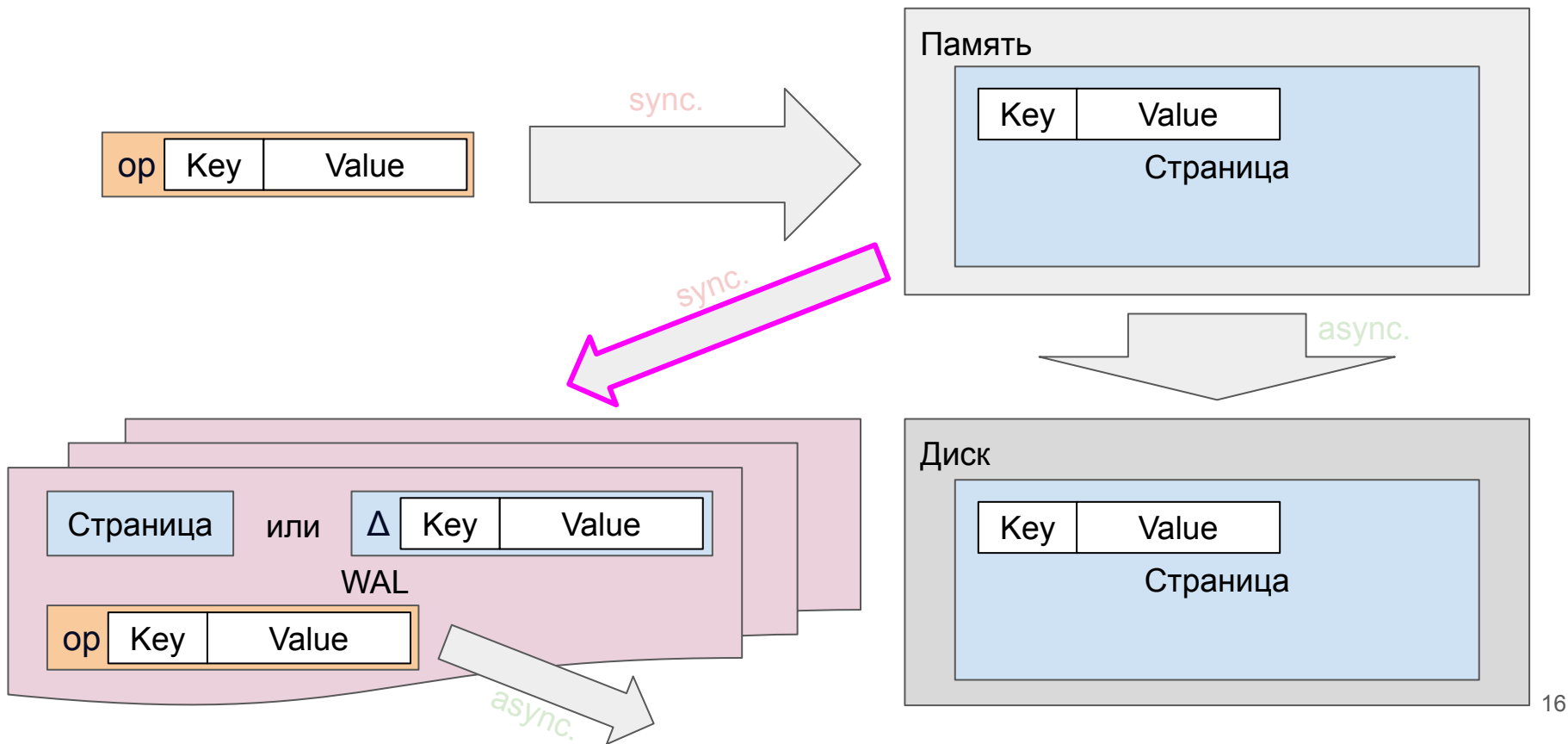
Межкластерная репликация



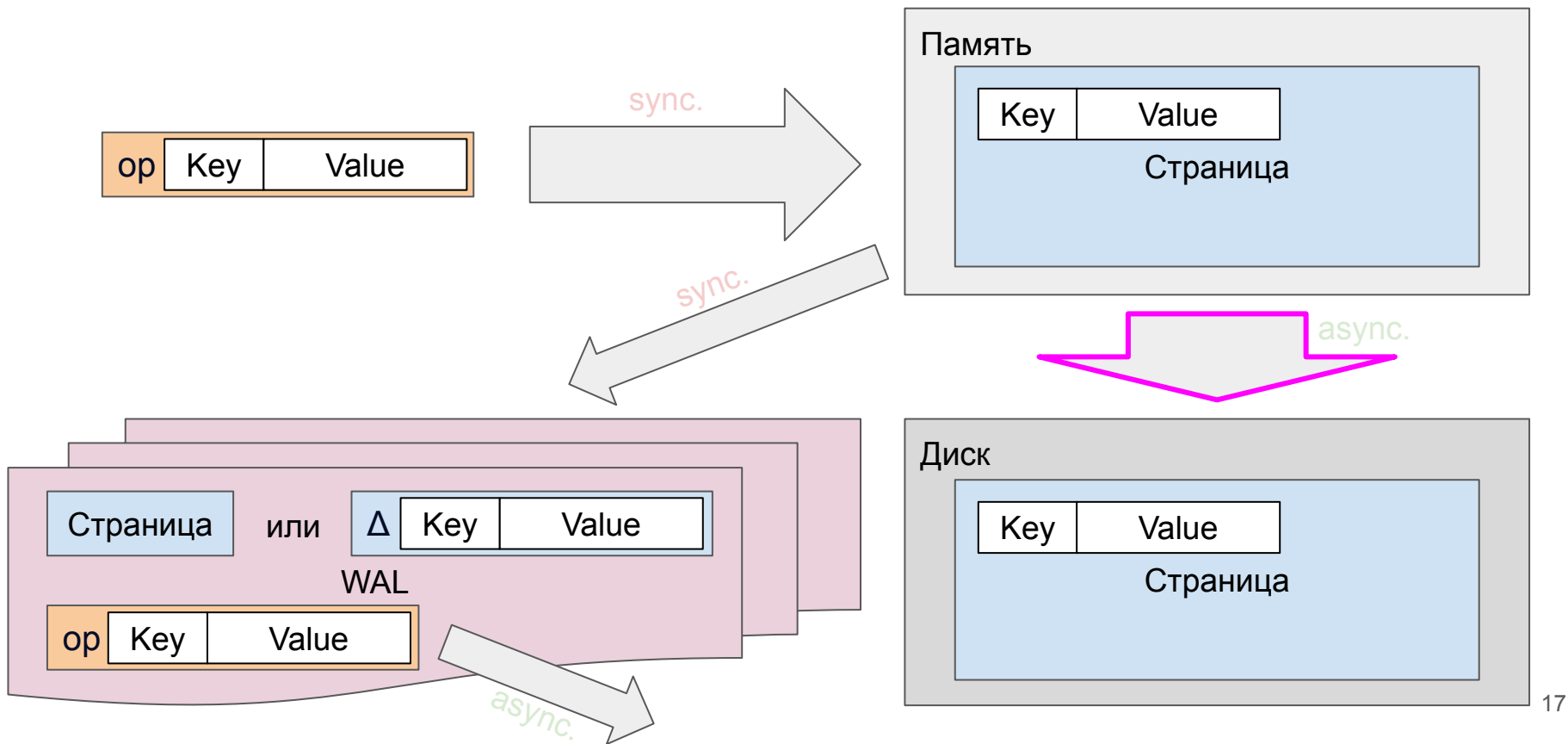
Записываем в память



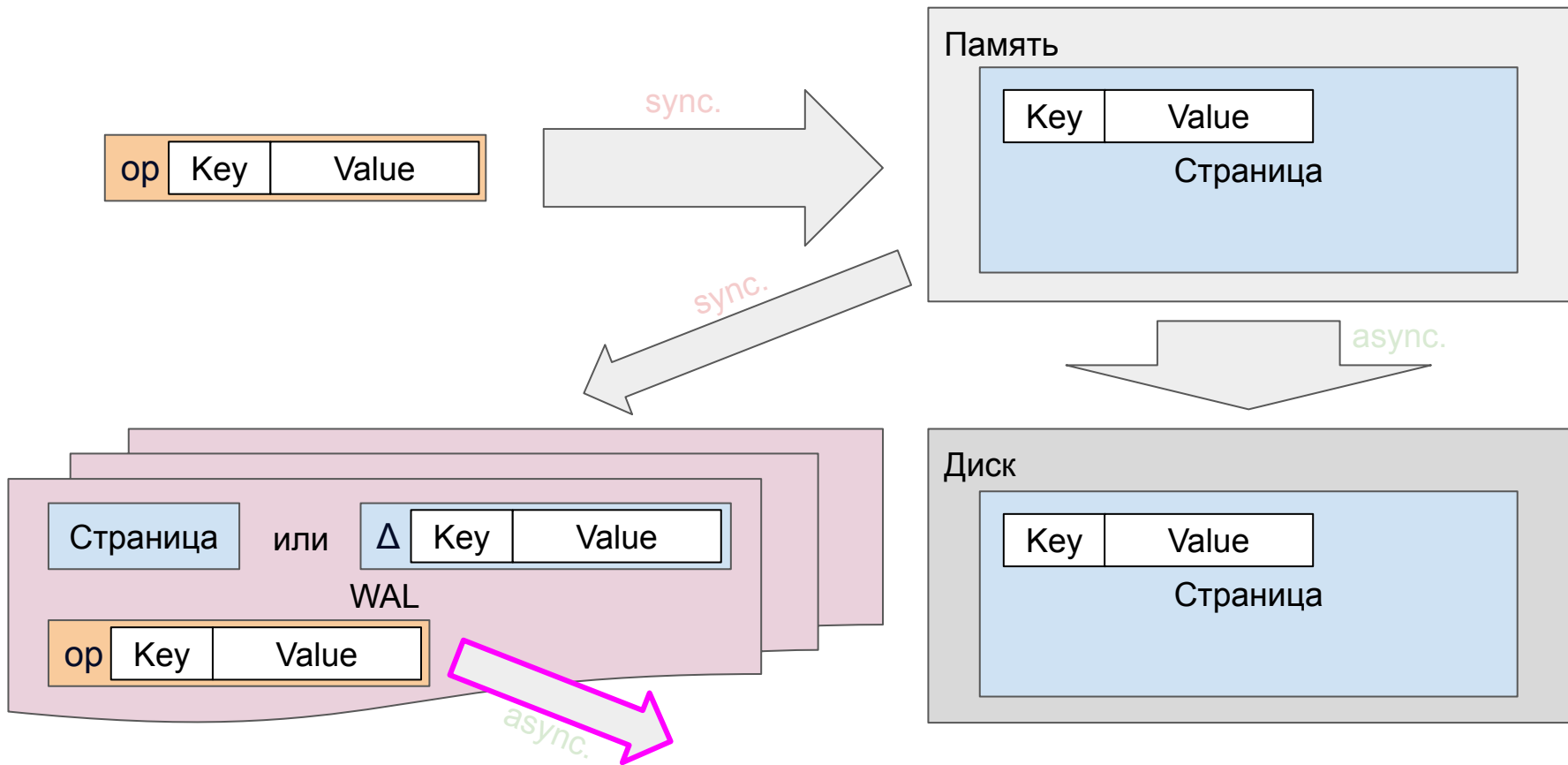
Гарантируем запись на диск



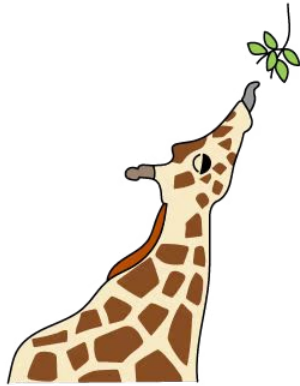
Записываем на диск



Отдаем внешним системам



Трансформация!



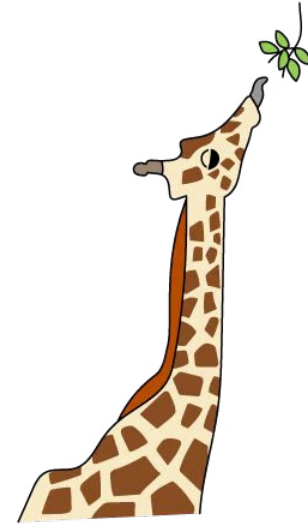
THE ORIGINAL
SHORT-NECKED
ANCESTOR



GIRAFFE KEEPS
STRETCHING ITS
NECK TO REACH
HIGHER LEVELS

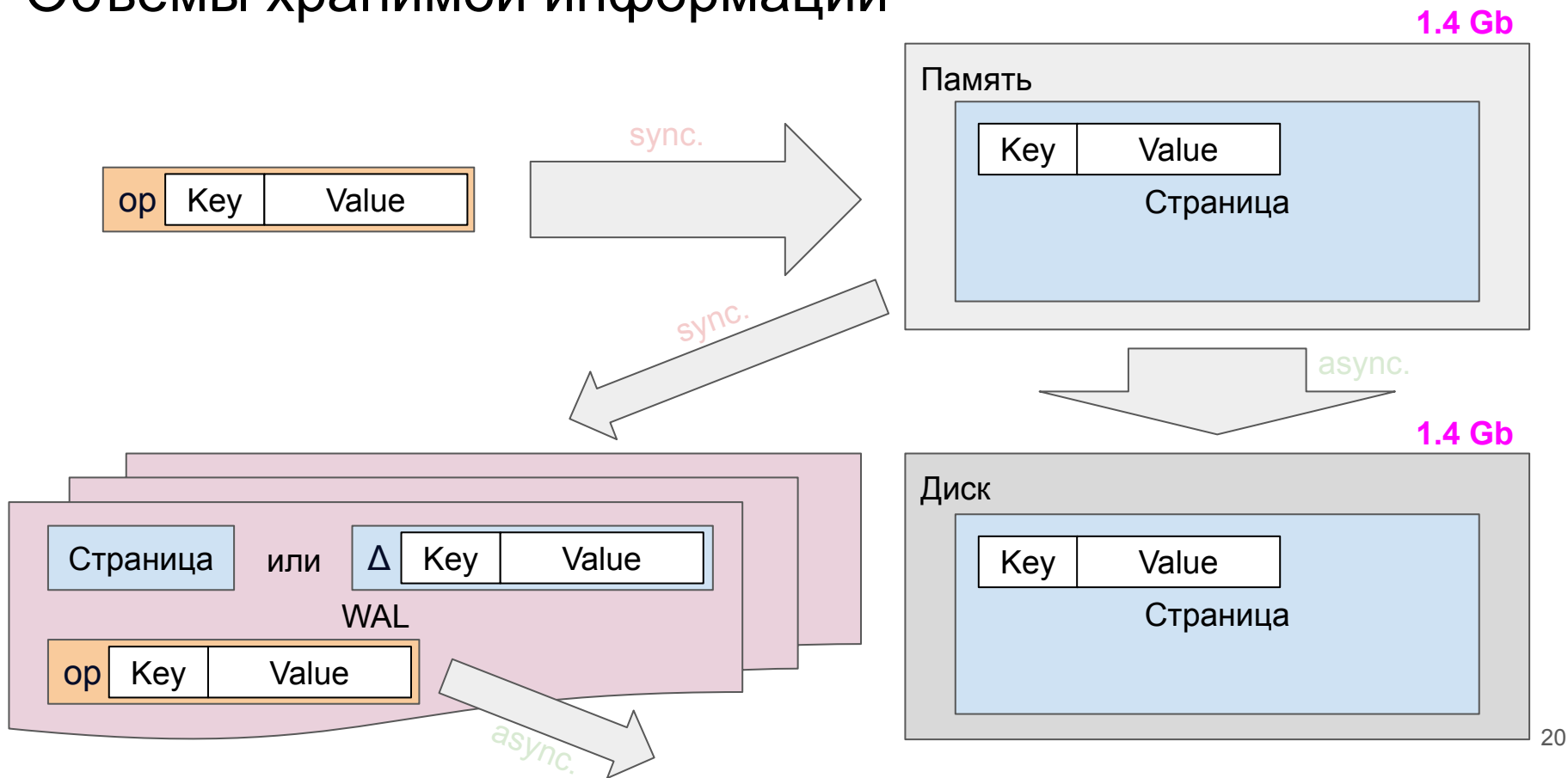


AND CONTINUES
STRETCHING UNTIL
NECK BECOMES
PROGRESSIVELY
LONGER

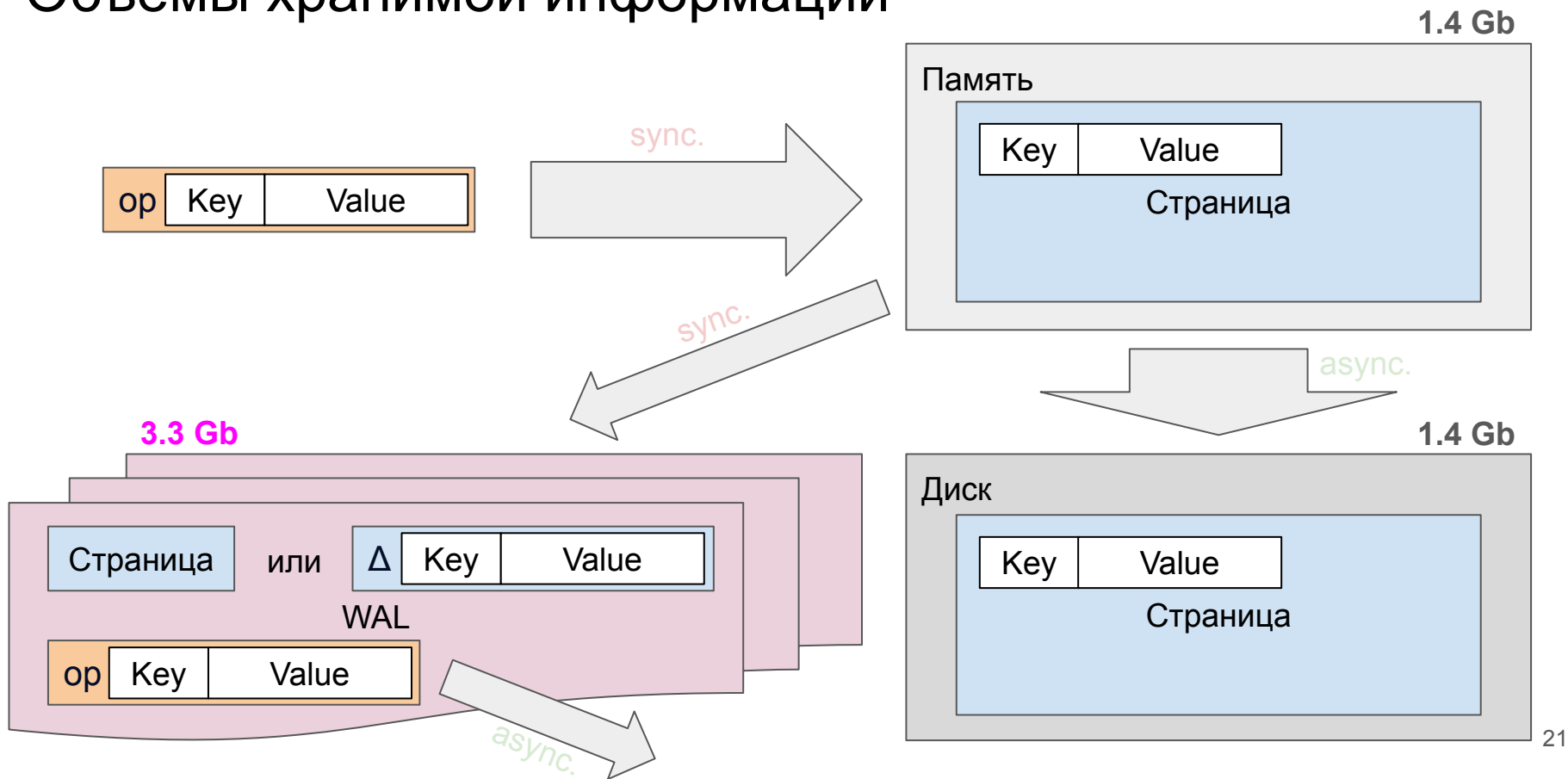


LONG-NECKED
DESCENDANT
AFTER MANY
GENERATIONS

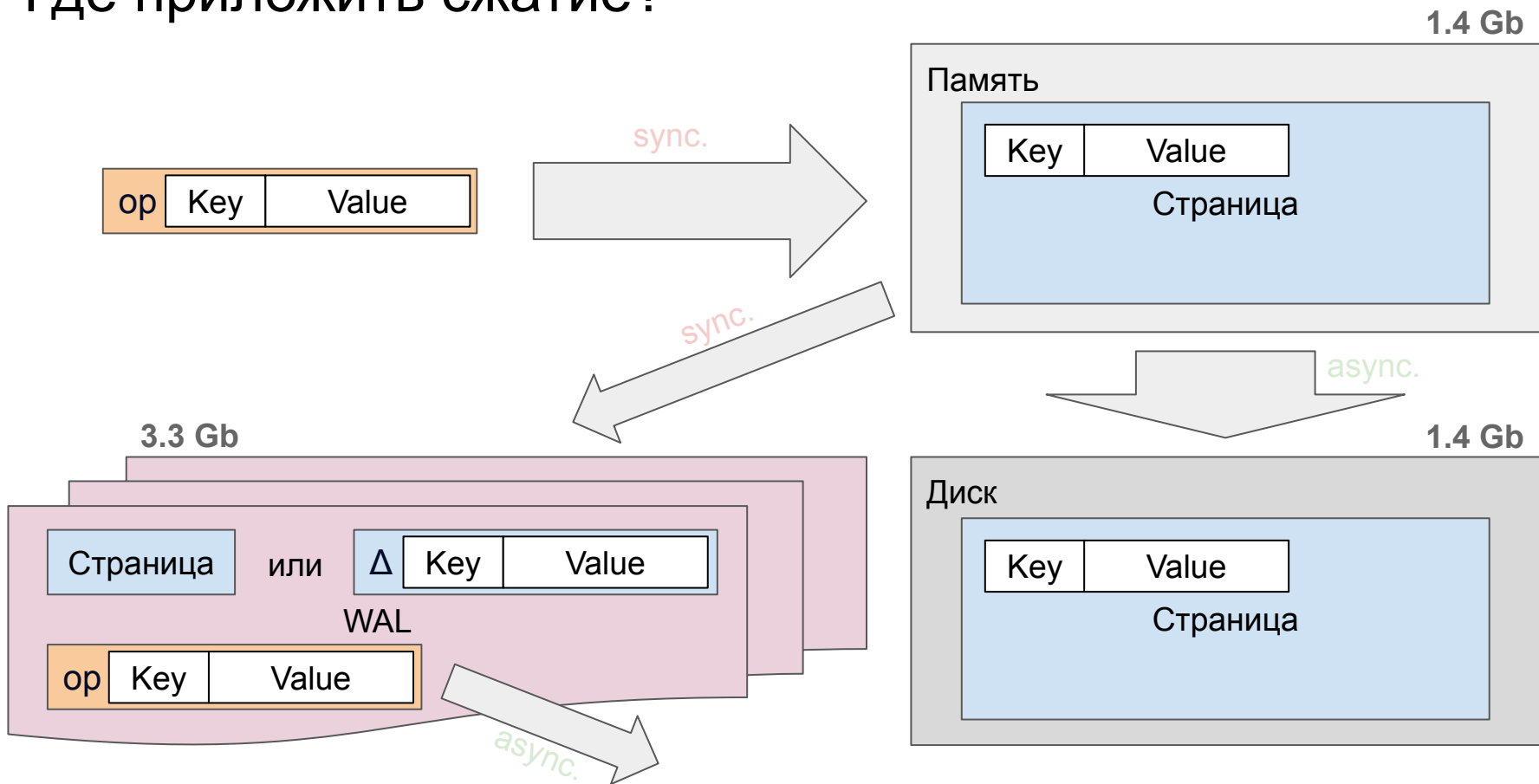
Объемы хранимой информации



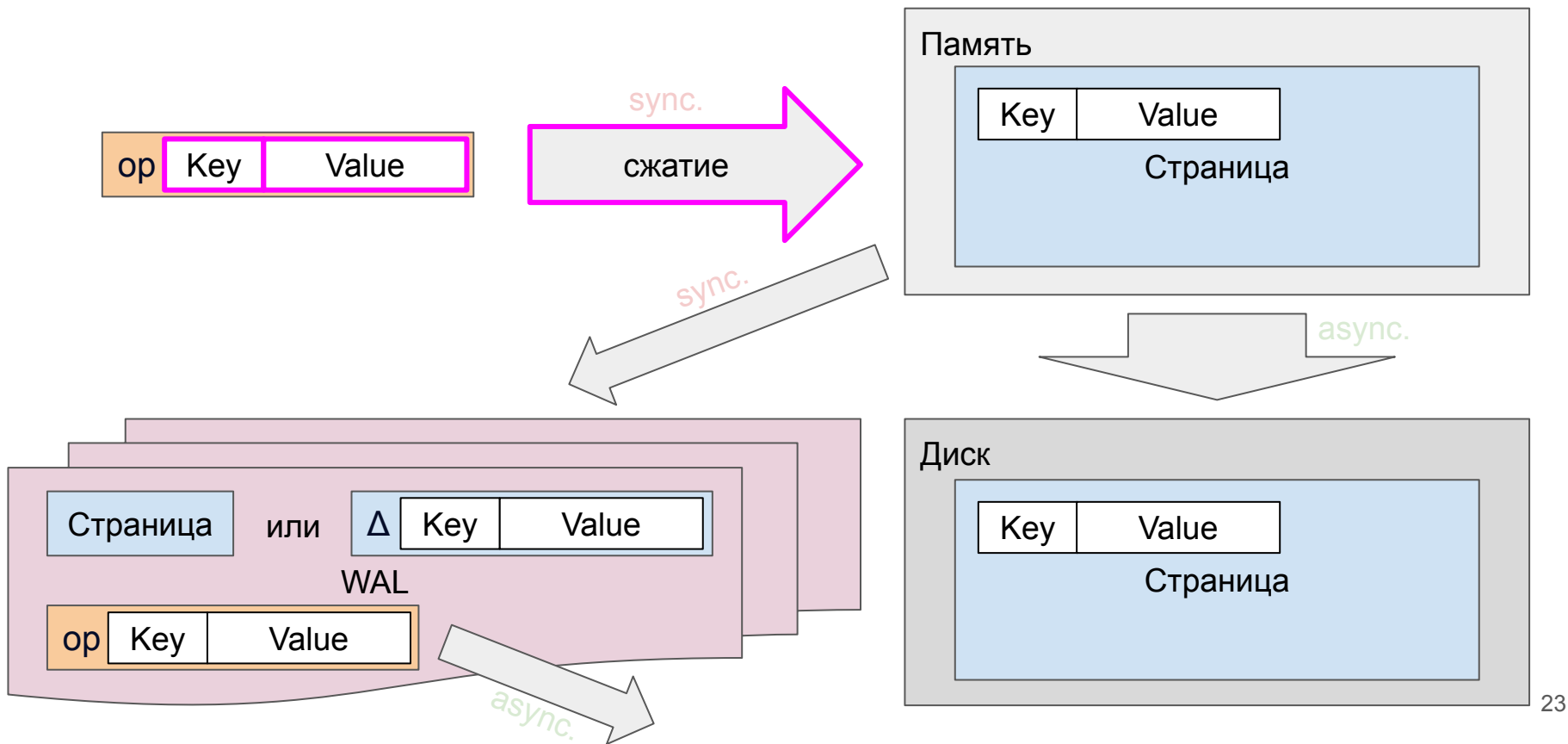
Объемы хранимой информации



Где приложить сжатие?



Сжимаем данные на входе!



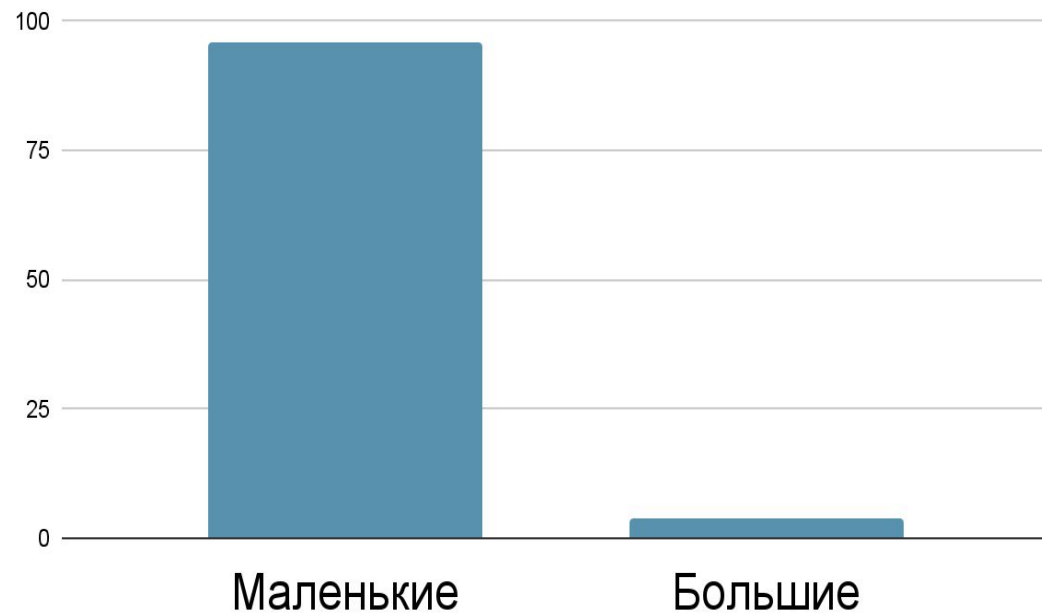
Эффективность сжатия

Key	Value
-----	-------

Эффективность сжатия

32	180
----	-----

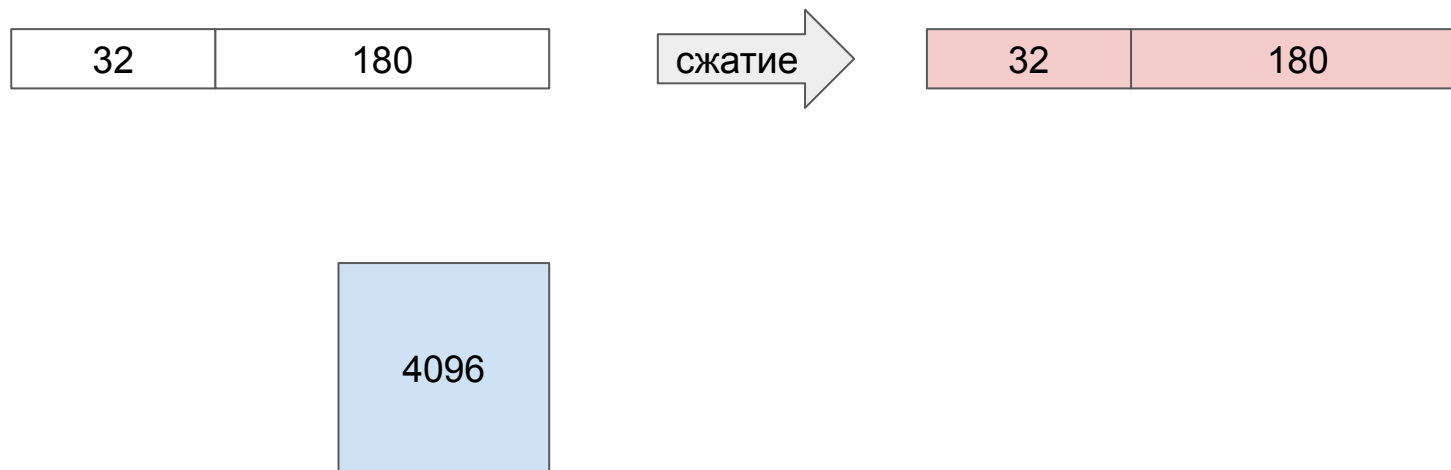
Соотношение числа малых и больших данных



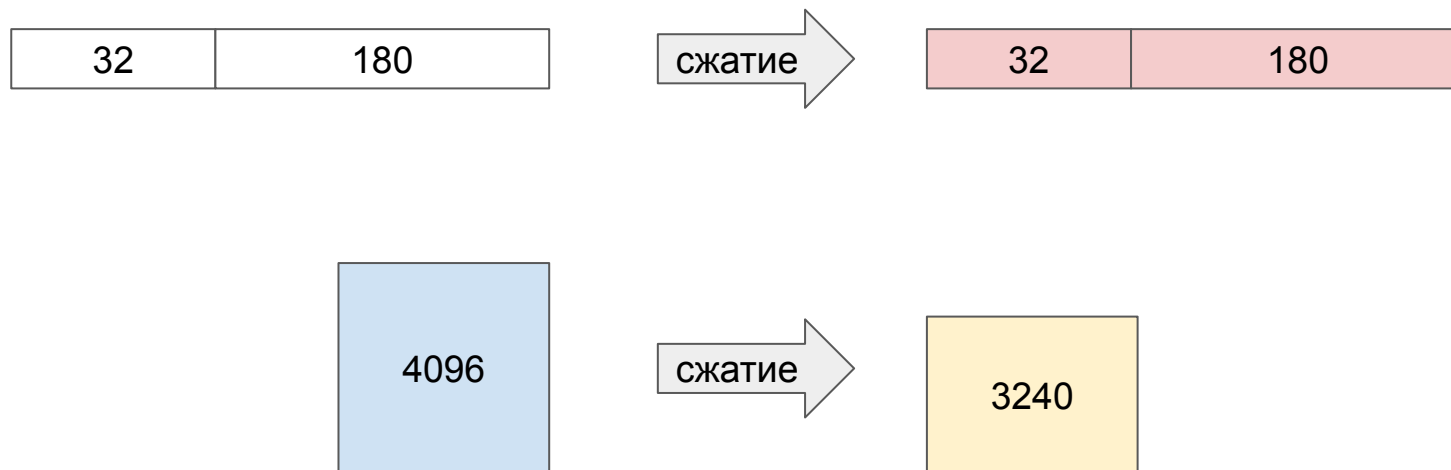
Эффективность сжатия данных малого размера



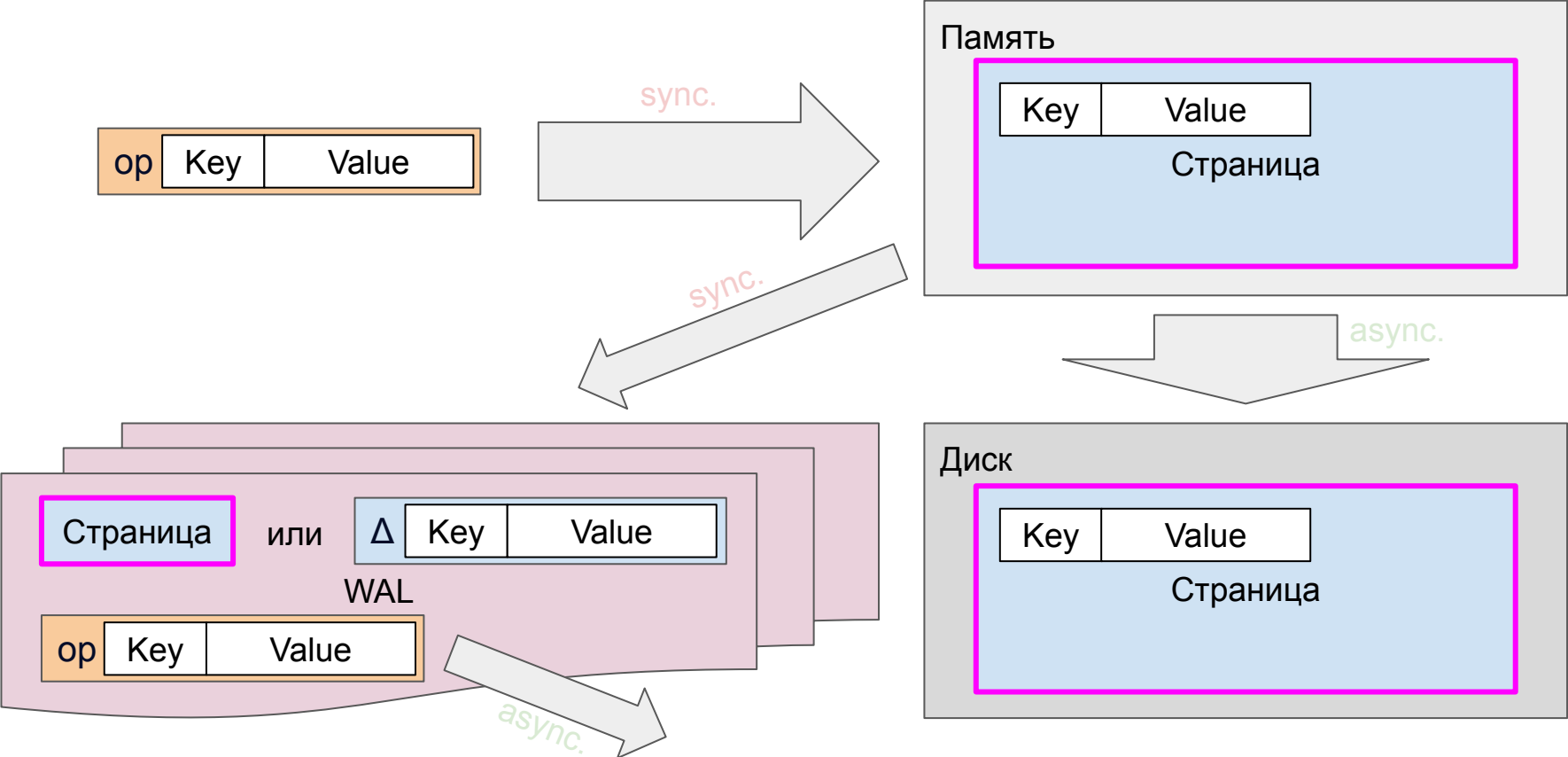
Эффективность сжатия страниц



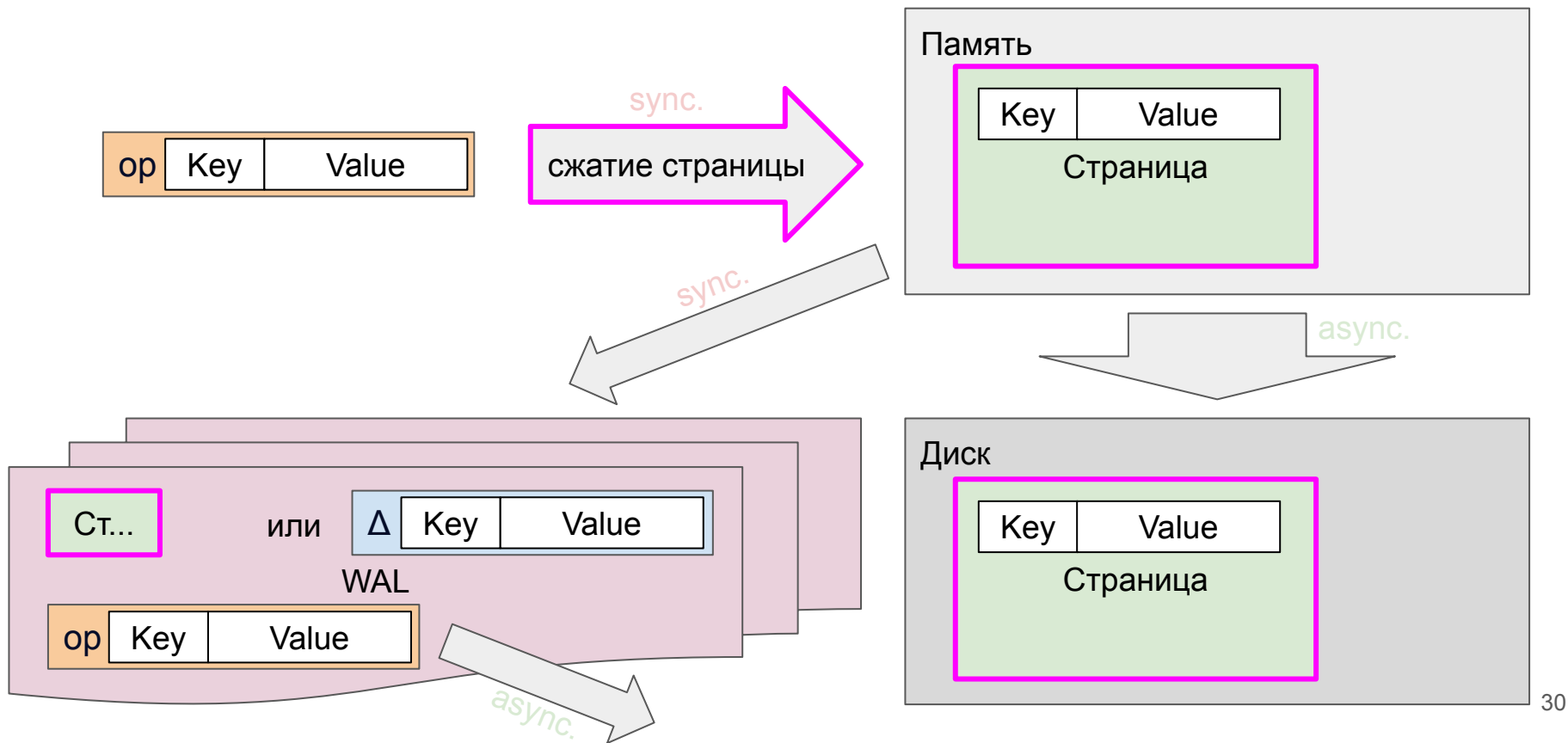
Эффективность сжатия страниц



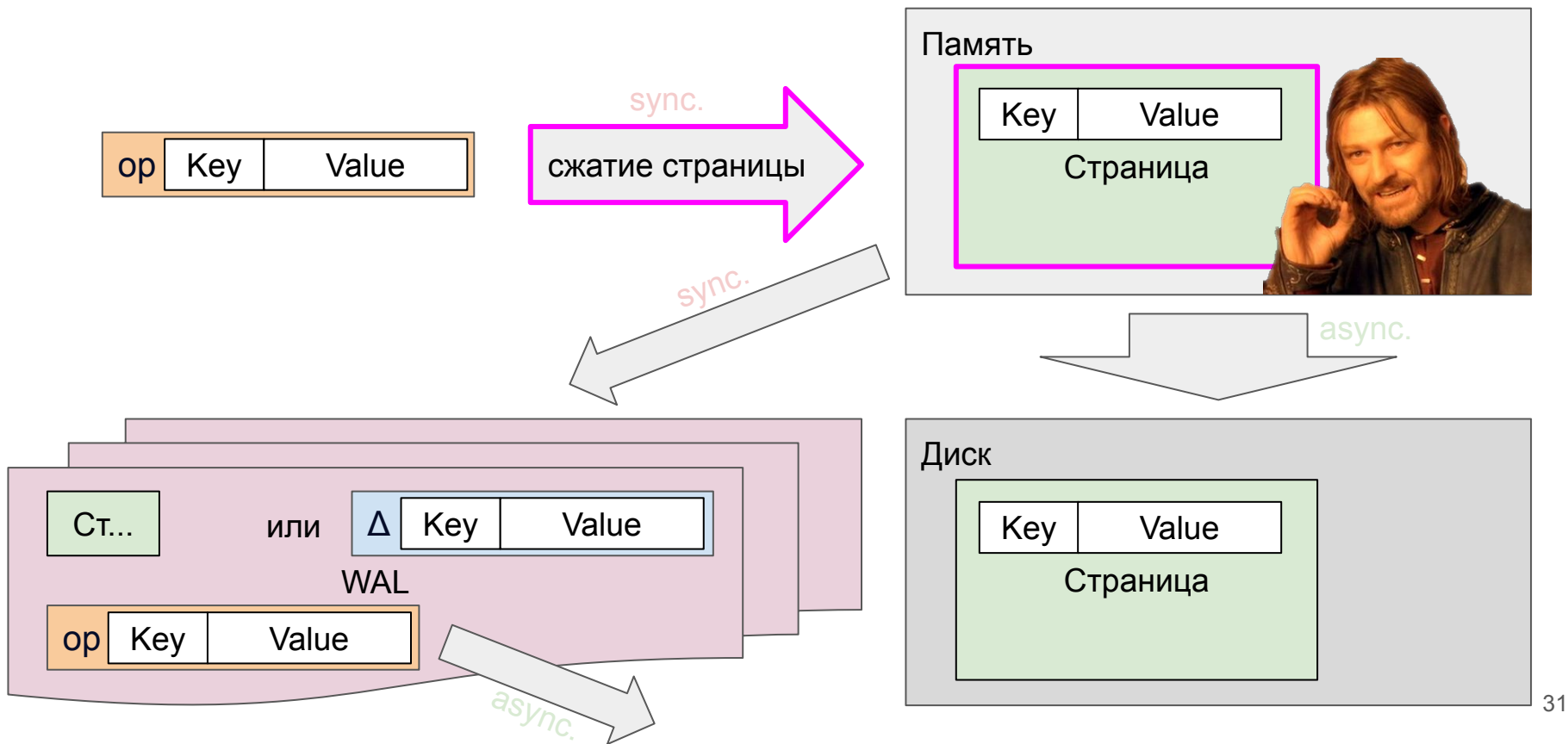
Будем сжимать здесь!



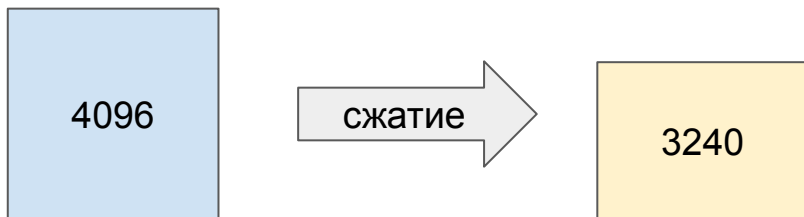
Сжимаем страницы везде



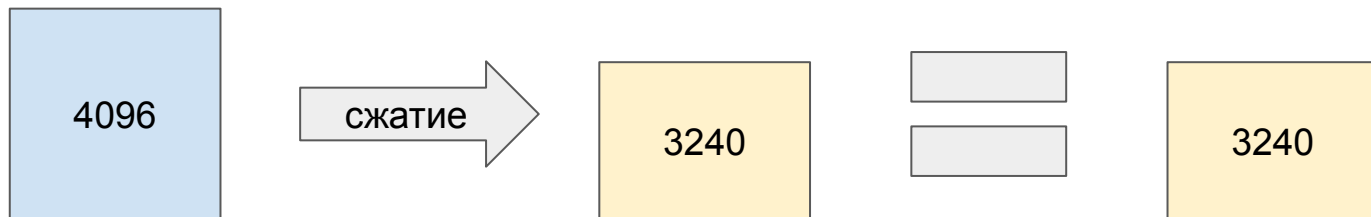
Сжимаем страницы?



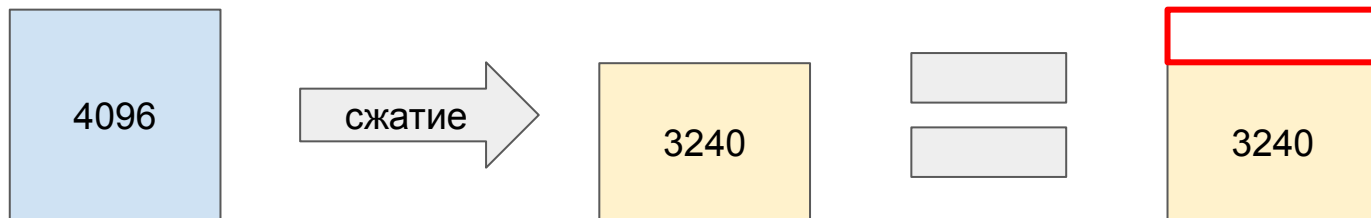
Сжимаем страницы



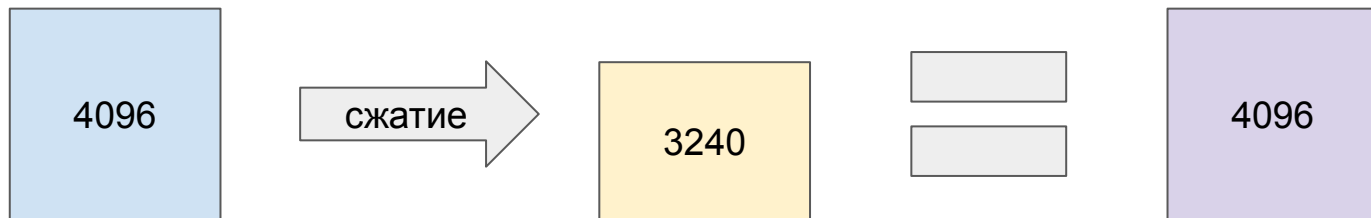
Сжимаем страницы (Ожидания)



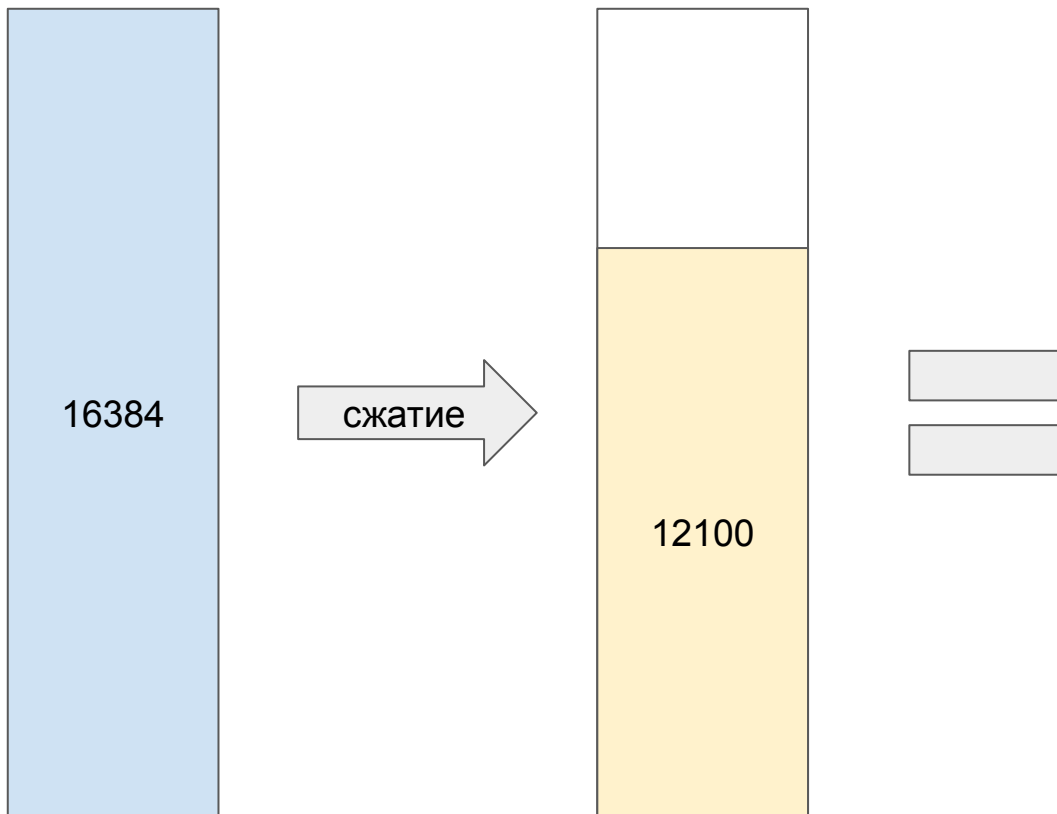
Сжимаем страницы (Реальность)



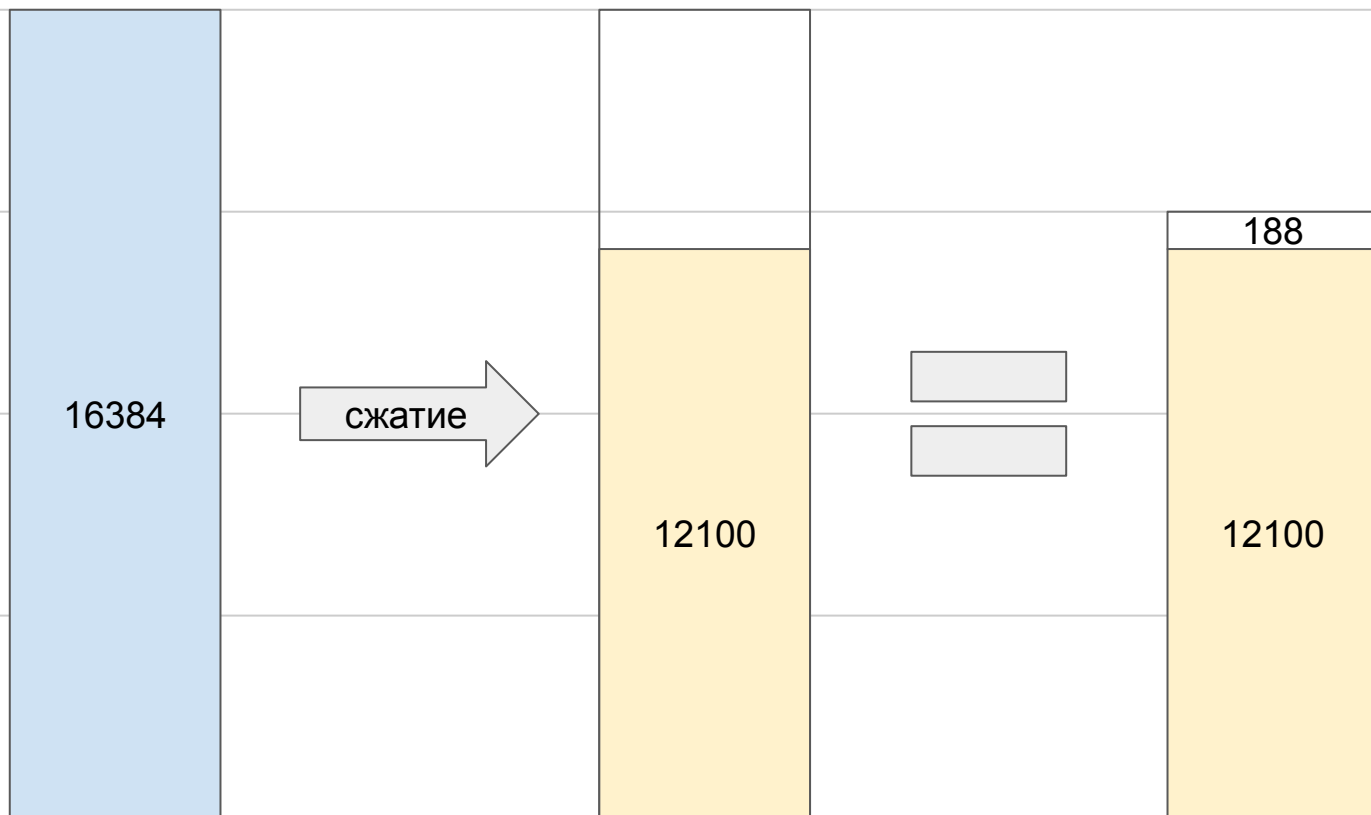
Сжимаем страницы (Блок 4Кб)



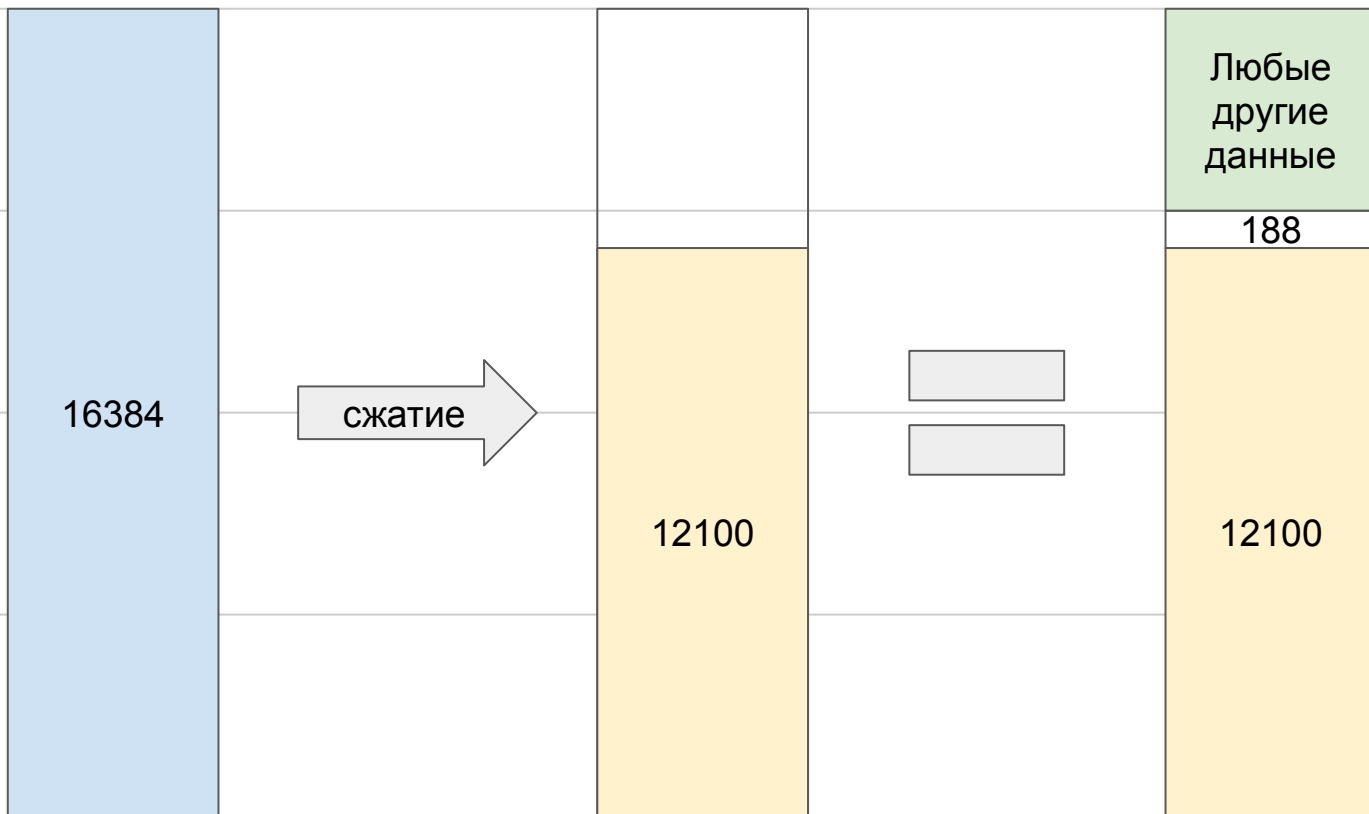
Сжимаем БОЛЬШИЕ страницы



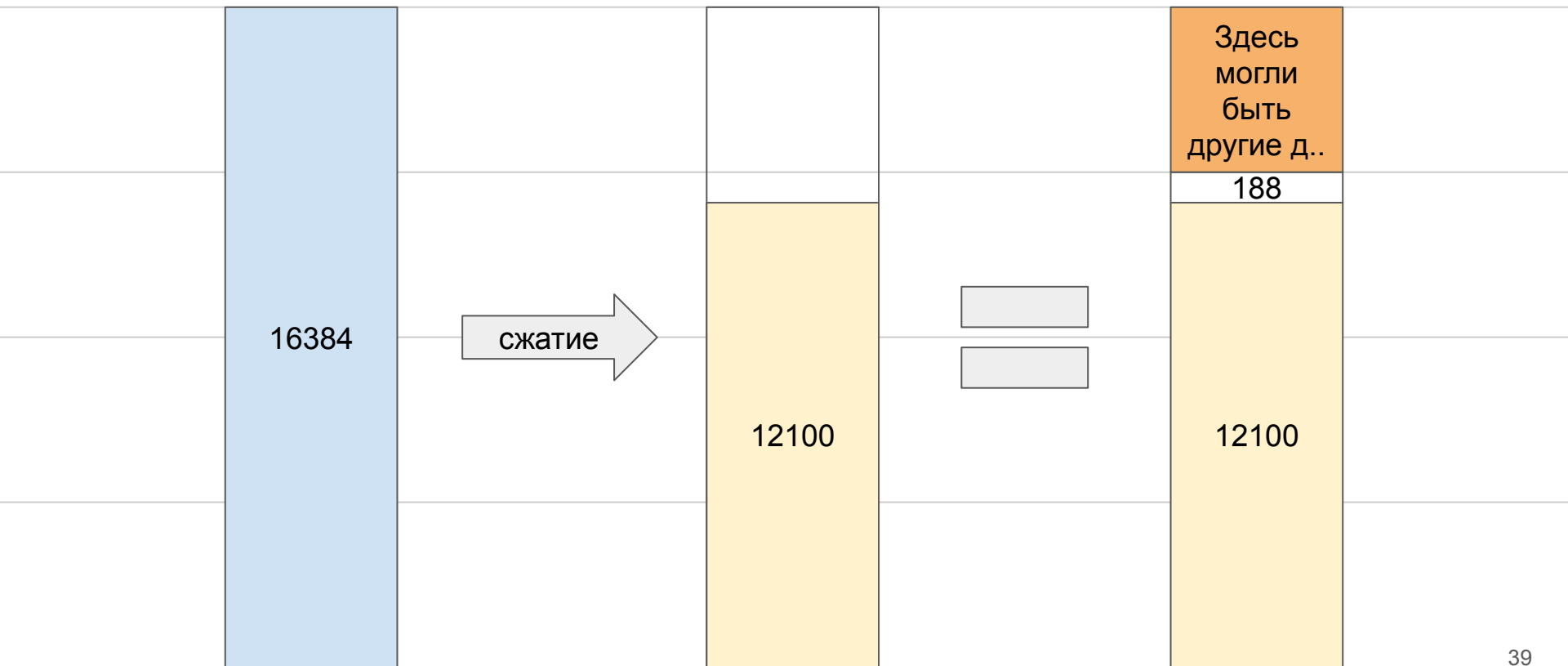
Сжимаем БОЛЬШИЕ страницы (Блок 4Кb)



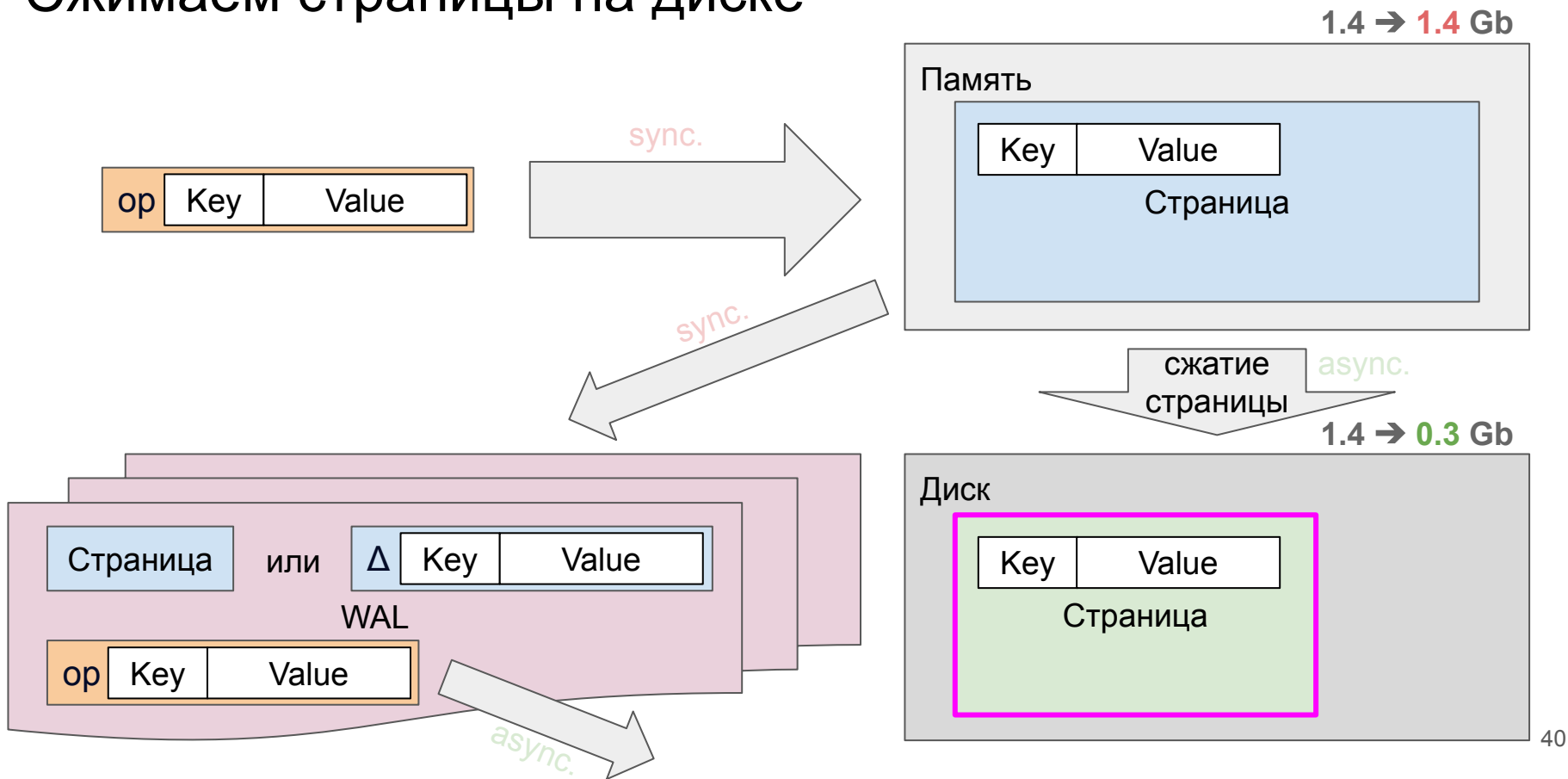
Сжимаем БОЛЬШИЕ страницы на диске



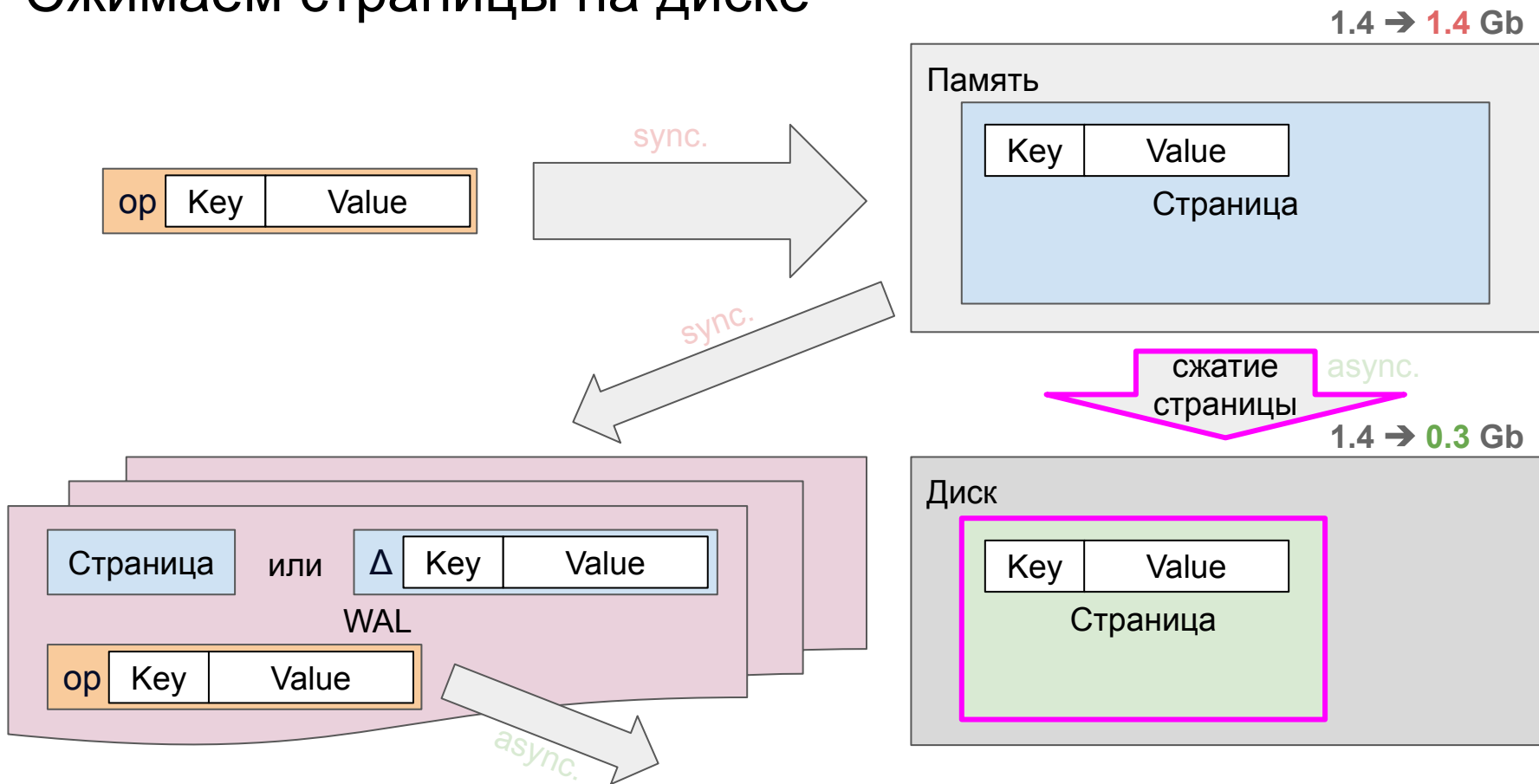
Сжимаем БОЛЬШИЕ страницы в памяти



Сжимаем страницы на диске

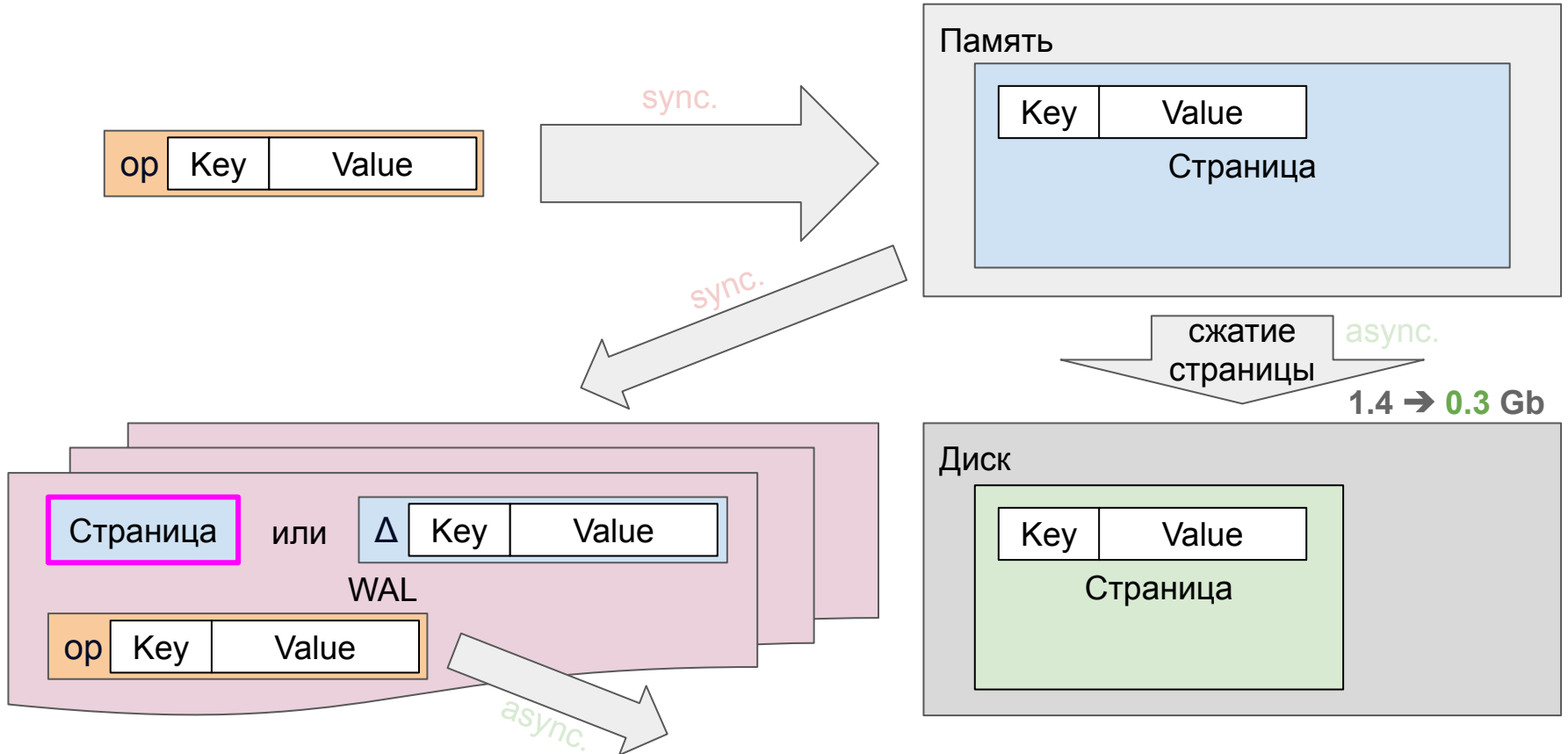


Сжимаем страницы на диске



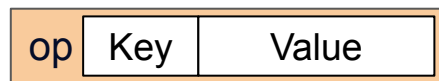
Сжимаем страницы только в хранилище :(

1.4 → 1.4 Gb

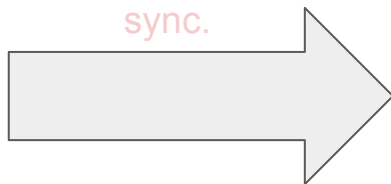


Сжимаем страницы только в хранилище :(

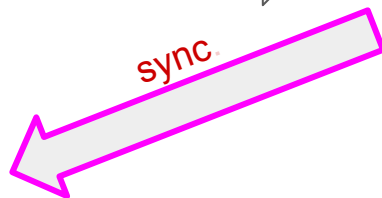
1.4 → 1.4 Gb



sync.

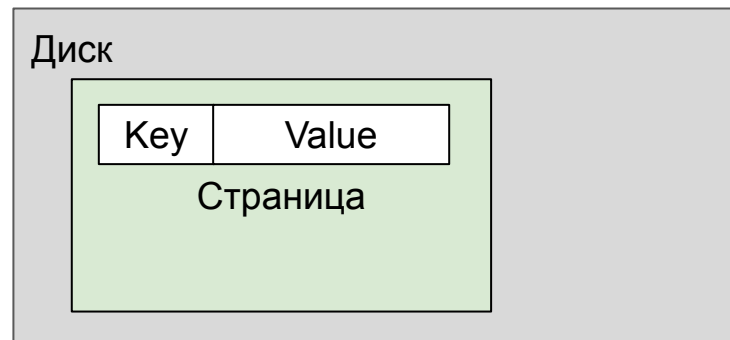
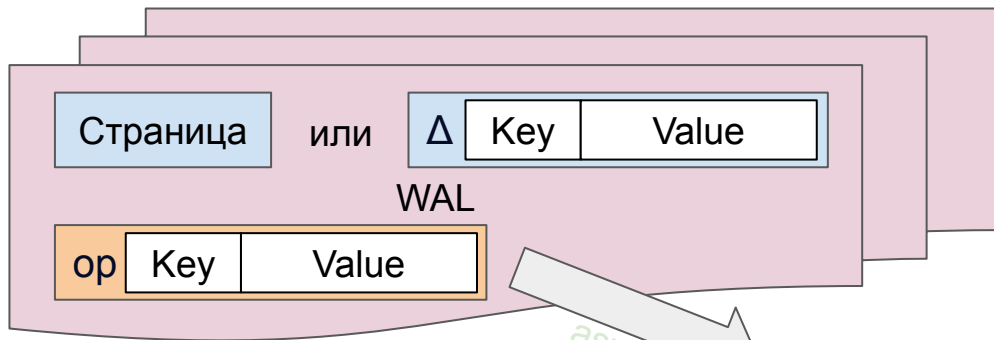


sync.



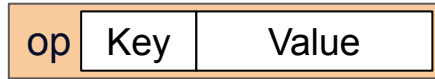
сжатие
страницы *async.*

1.4 → 0.3 Gb

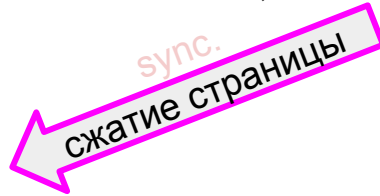
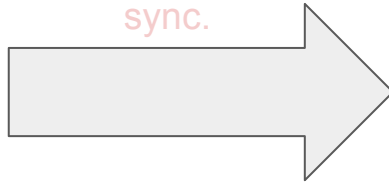


Сжимаем страницы в хранилище и WAL

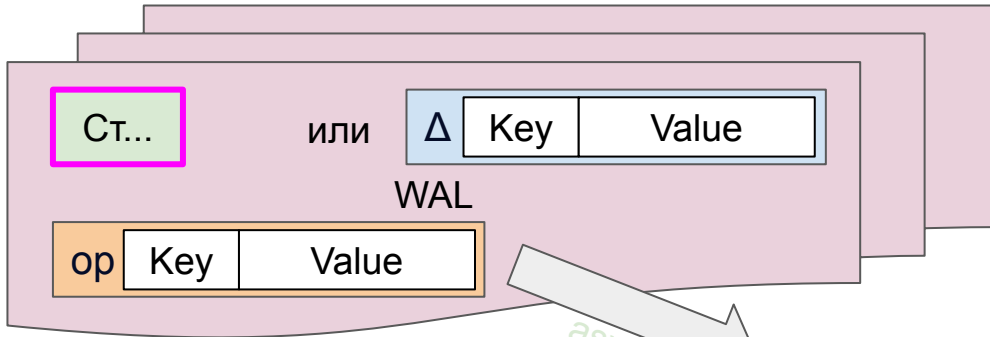
1.4 → 1.4 Gb



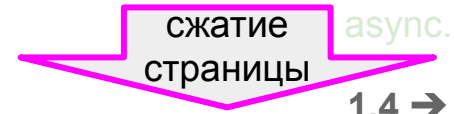
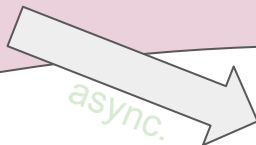
sync.



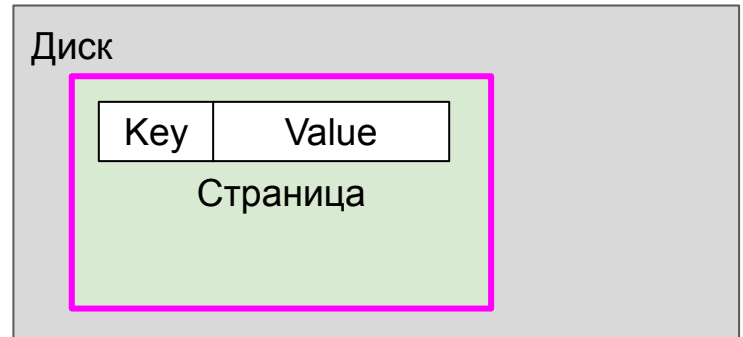
3.3 → 2.7 Gb



async.

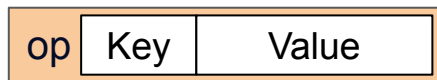


1.4 → 0.3 Gb

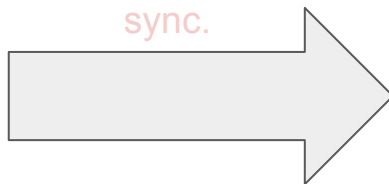


4 → 16 Kb

1.4 → 1.4 Gb



sync.

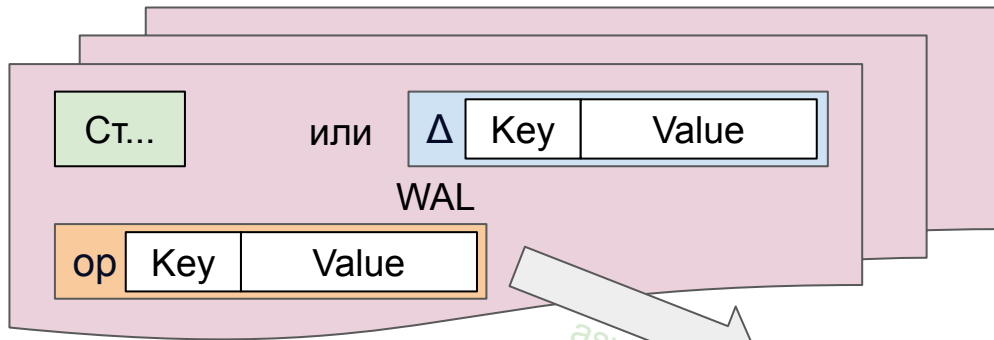


sync.

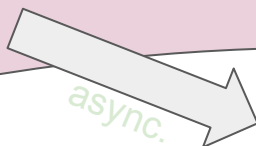
сжатие стр. **16Kb**



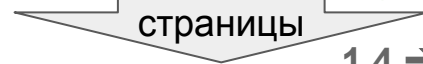
3.3 → 2.7 Gb



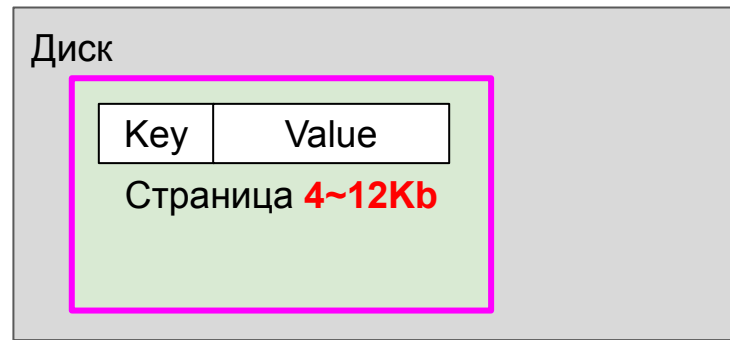
async.



сжатие страниц *async.*

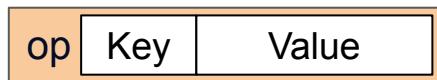


1.4 → 0.3 Gb

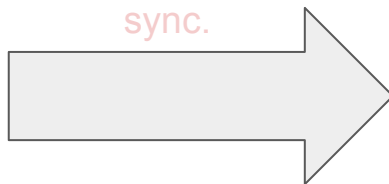


WAL не требует кратности блоку ФС

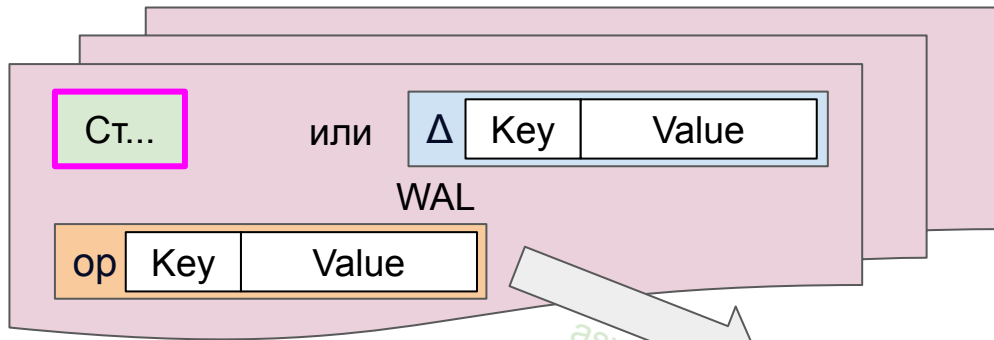
1.4 → 1.4 Gb



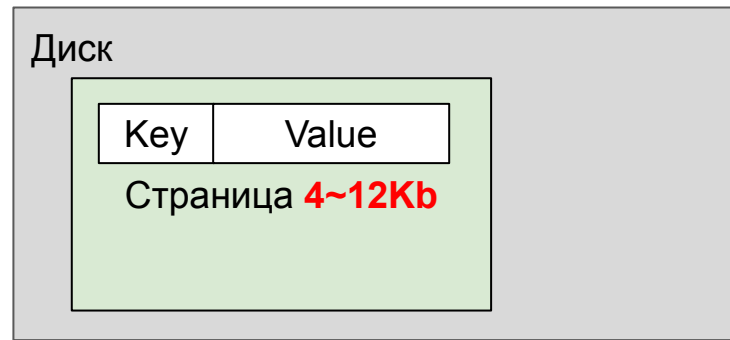
sync.



3.3 → 2.7 Gb

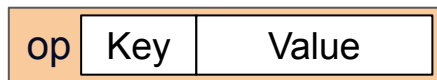


1.4 → 0.3 Gb

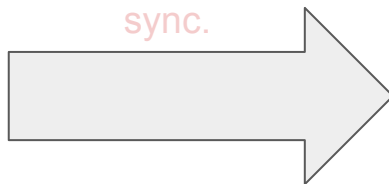


1.4 → 0.3 = Успех!

1.4 → 1.4 Gb



sync.

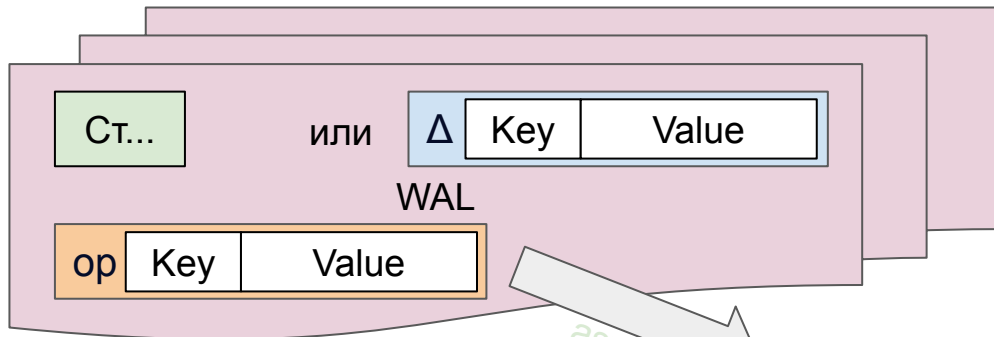


sync.

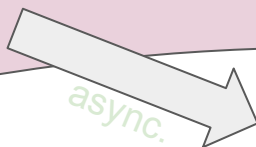
сжатие стр. 16Kb



3.3 → 2.7 Gb



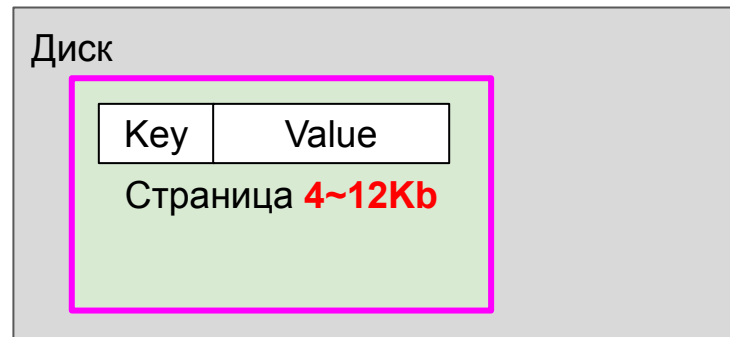
async.



сжатие страниц async.



1.4 → 0.3 Gb



Итого

	Без сжатия	Сжатие на диске
Память (Gb)	1.4	1.4
Диск (Gb)	1.4	<u>0.3</u>
WAL (Gb)	3.3	2.7
Latency, fast SSD (ms)	0.20	0.21
Latency, slow SSD (ms)	0.37	0.37

Итого

	Без сжатия	Сжатие на диске
Память (Gb)	1.4	1.4
Диск (Gb)	1.4	0.3
WAL (Gb)	3.3	<u>2.7</u>
Latency, fast SSD (ms)	0.20	0.21
Latency, slow SSD (ms)	0.37	0.37

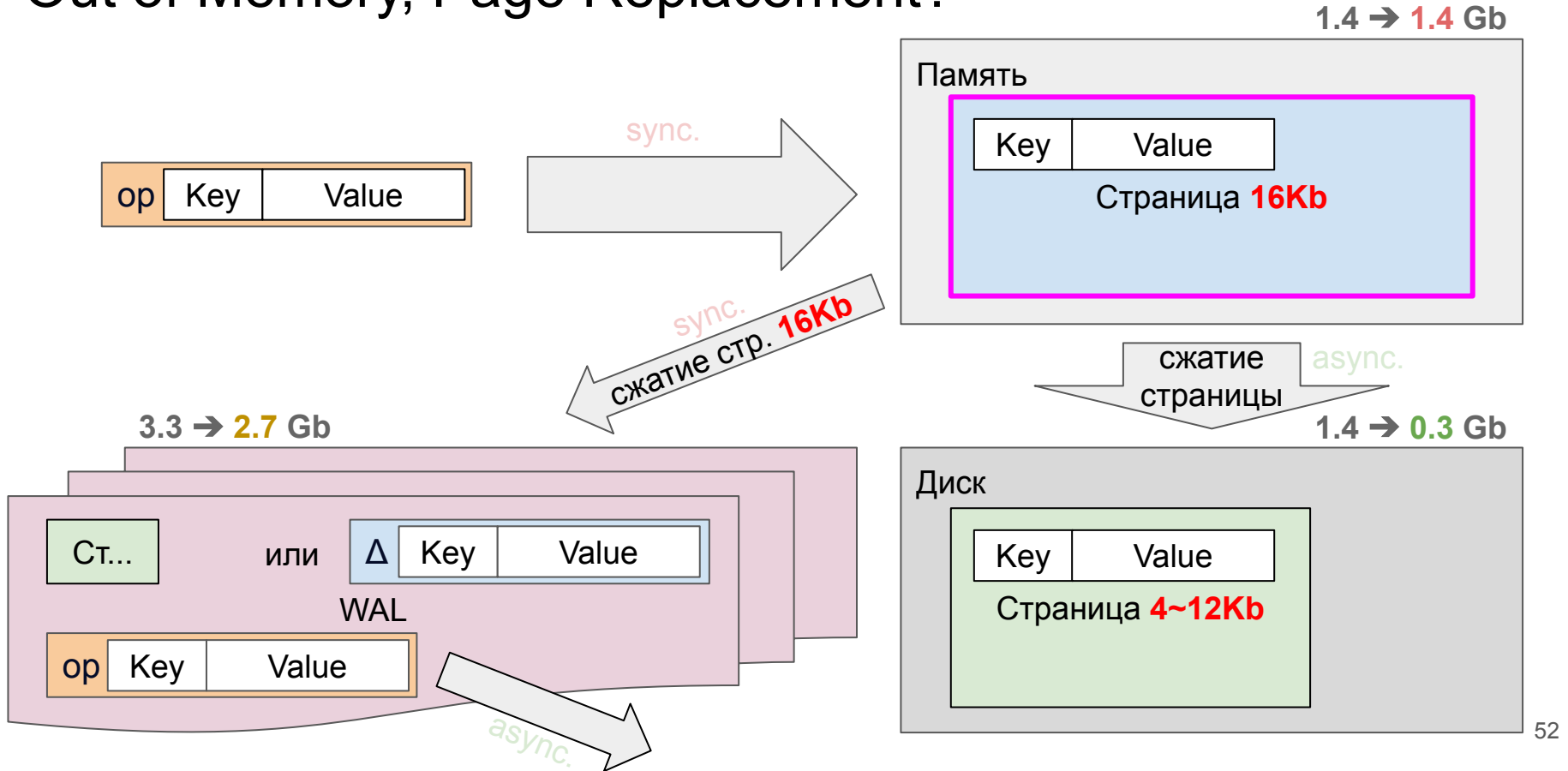
Итого

	Без сжатия	Сжатие на диске
Память (Gb)	1.4	1.4
Диск (Gb)	1.4	0.3
WAL (Gb)	3.3	2.7
Latency, fast SSD (ms)	0.20	<u>0.21</u>
Latency, slow SSD (ms)	0.37	<u>0.37</u>

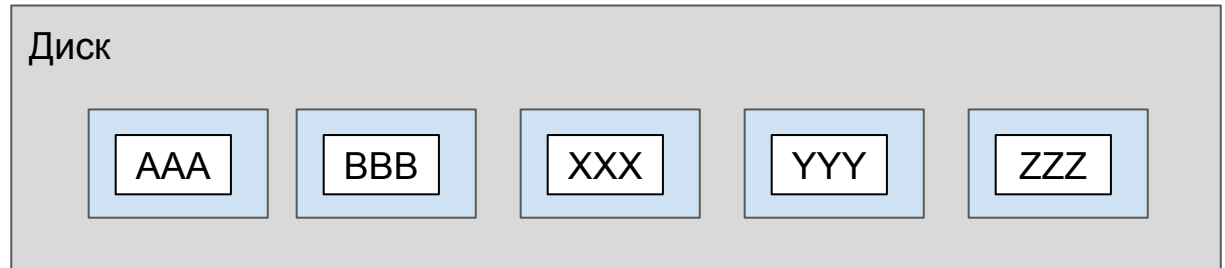
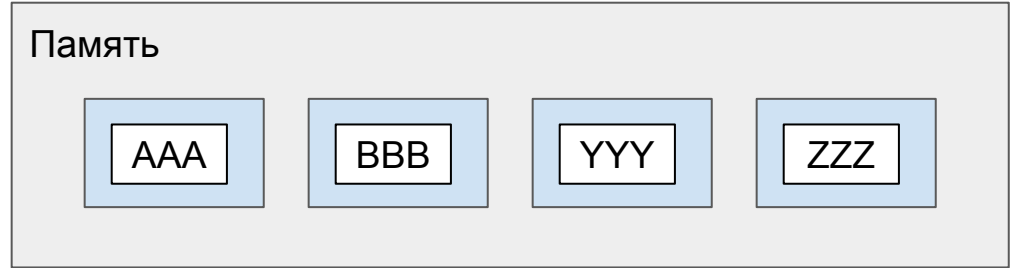
1.4 → 1.4 = Успех?

	Без сжатия	Сжатие на диске
Память (Gb)	1.4	<u>1.4</u>
Диск (Gb)	1.4	0.3
WAL (Gb)	3.3	2.7
Latency, fast SSD (ms)	0.20	0.21
Latency, slow SSD (ms)	0.37	0.37

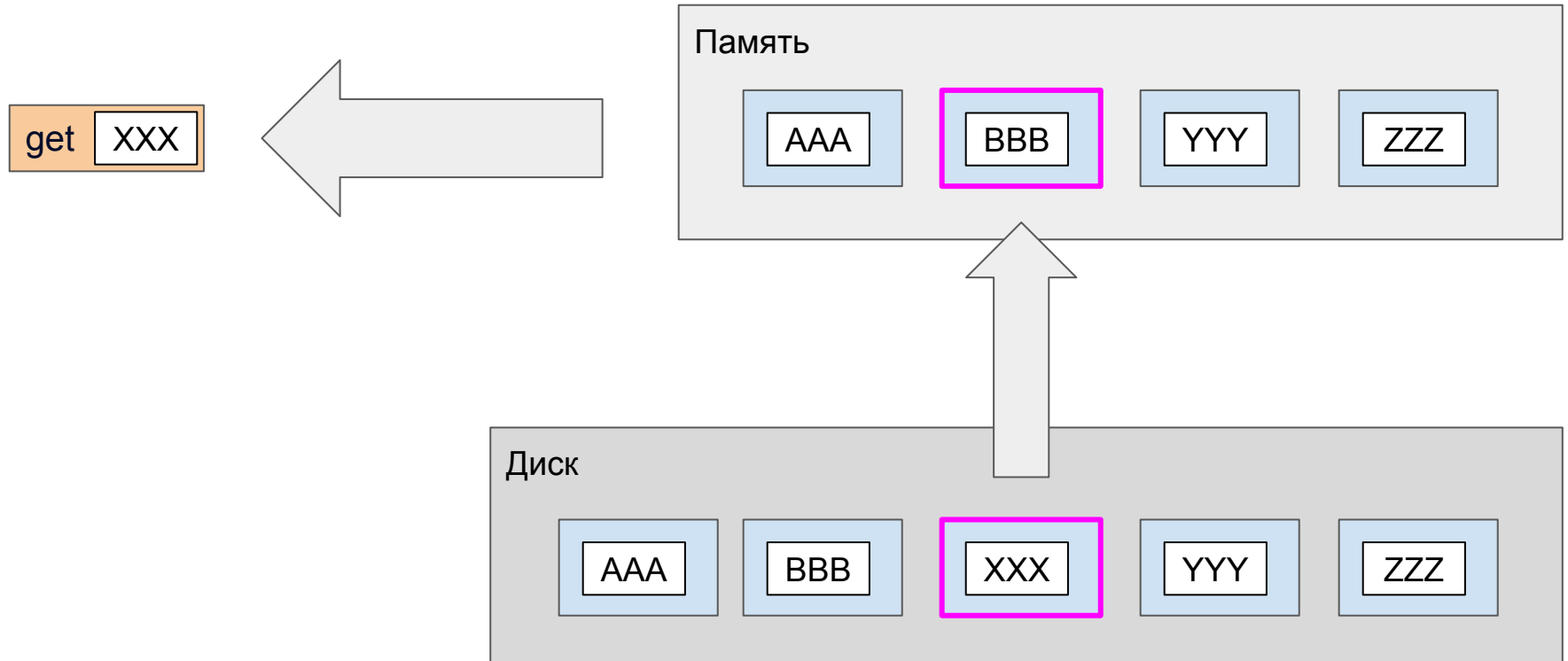
Out of Memory, Page Replacement?



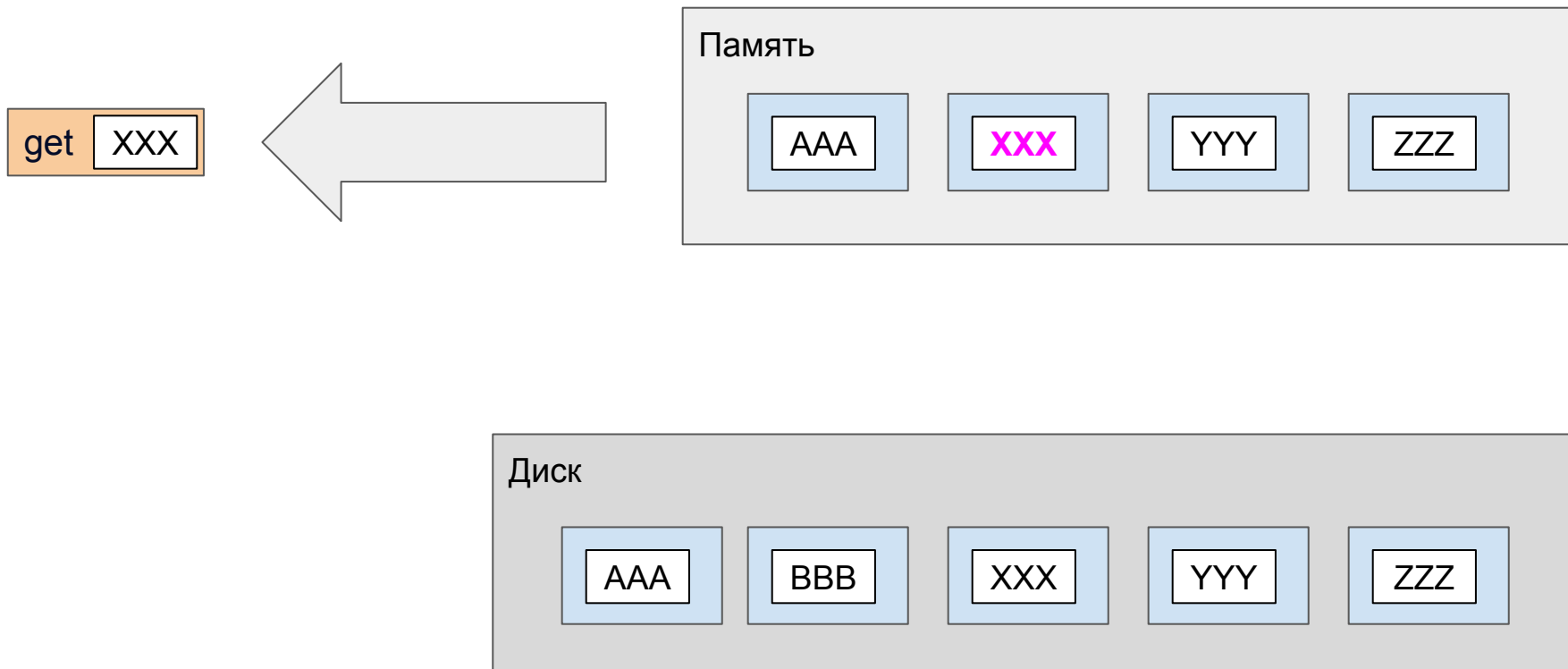
Чтение вытесненной страницы



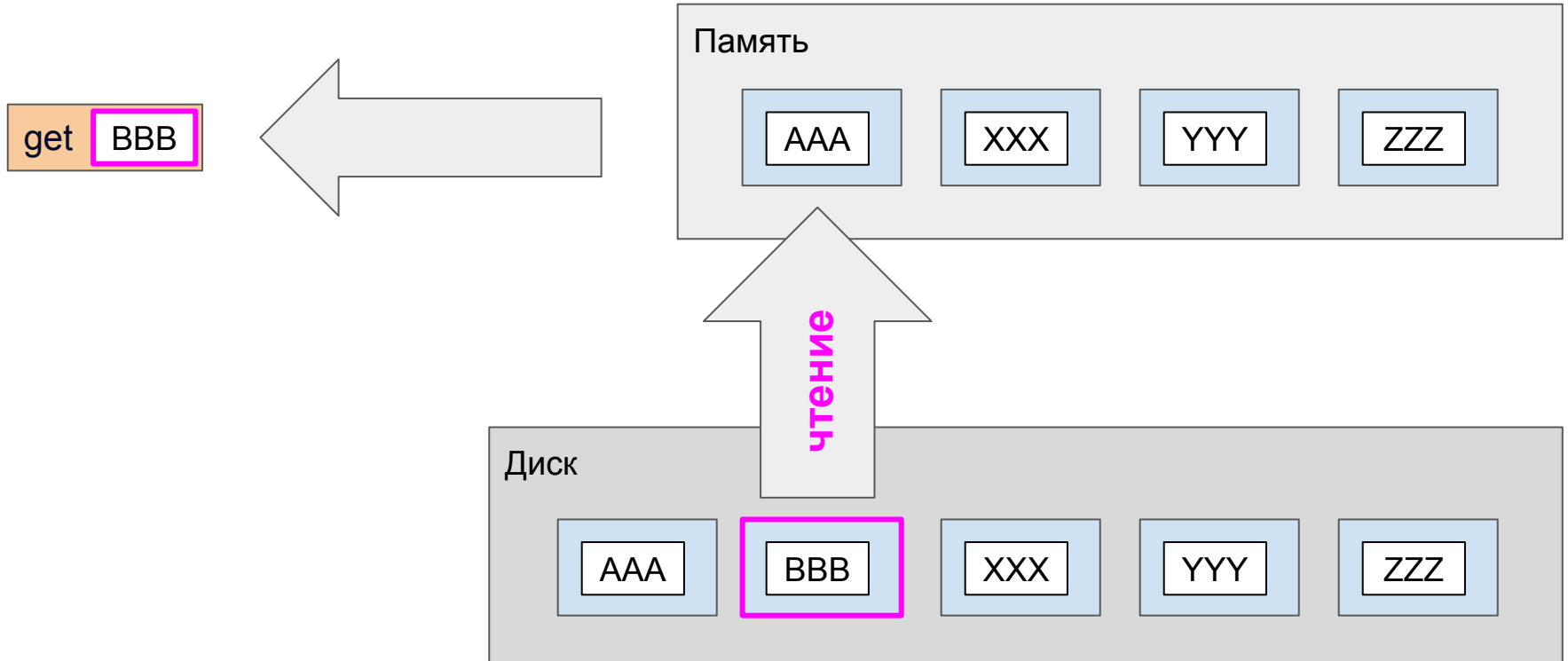
Чтение вытесненной страницы



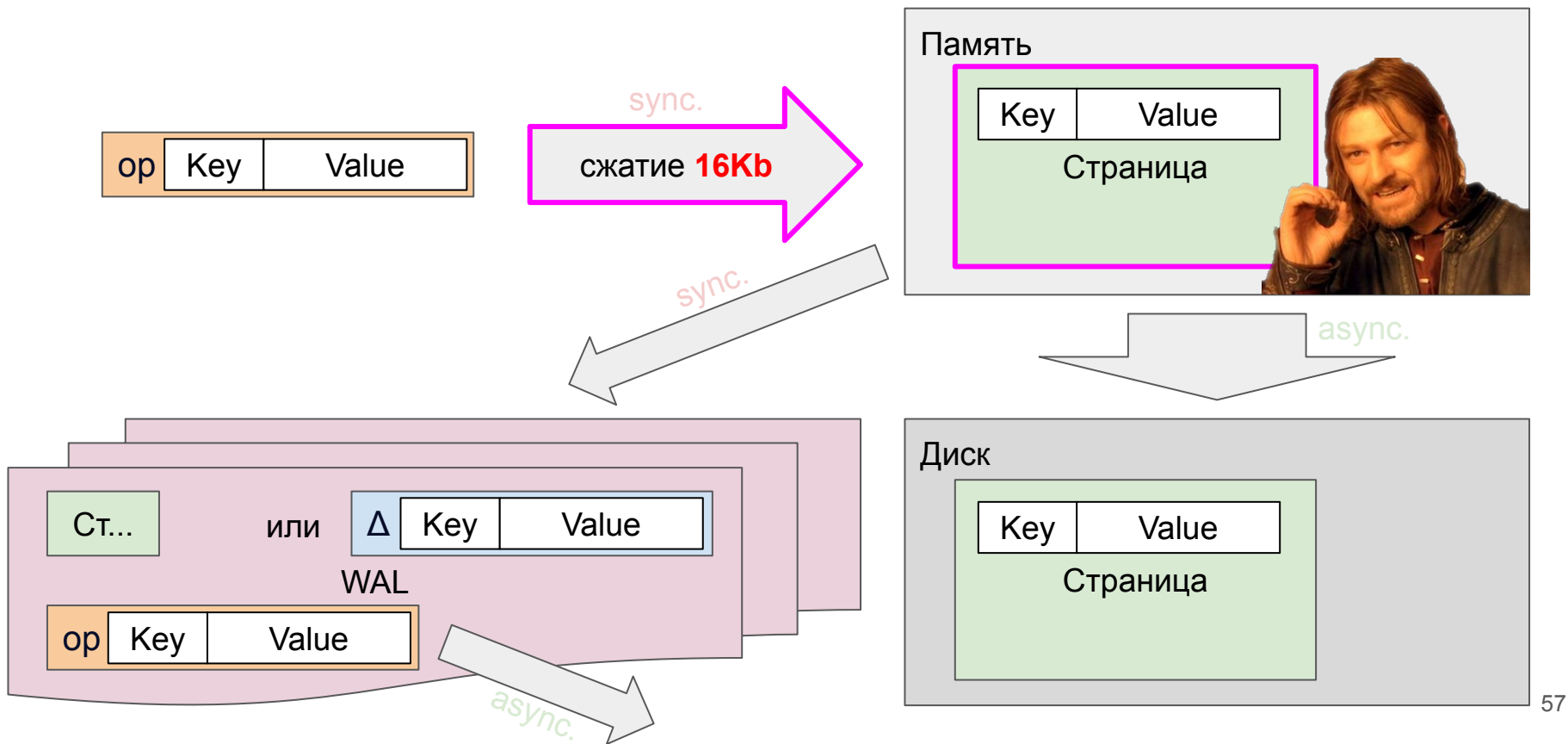
Чтение вытесненной страницы



Чтение вытесненной страницы = performance drop



Нужно же просто сжимать страницы везде!



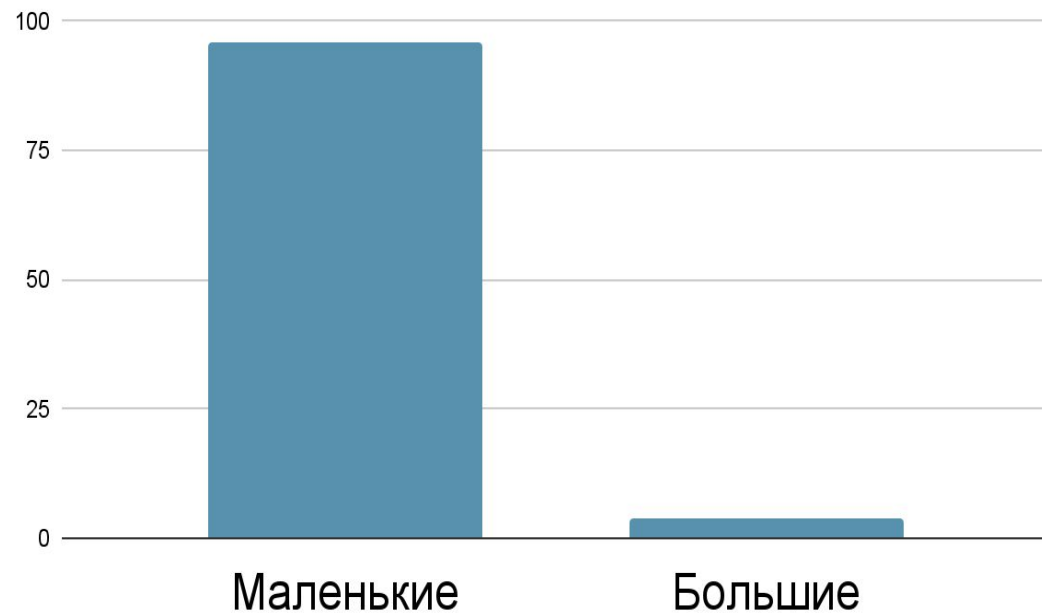
Эффективность сжатия

Key	Value
-----	-------

Эффективность сжатия

Key	Value
-----	-------

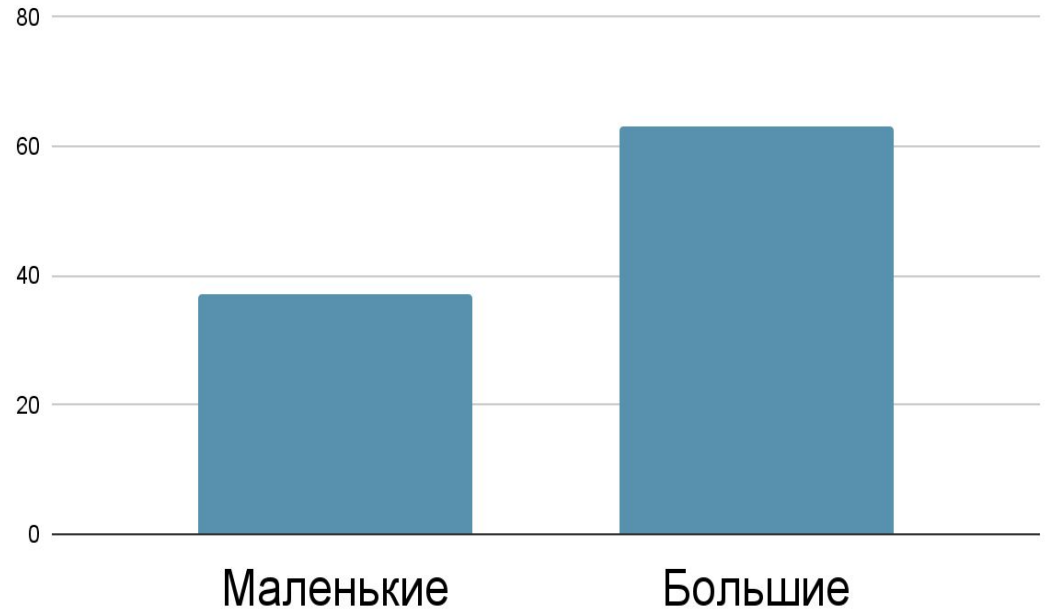
Соотношение числа малых и больших данных



Эффективность сжатия

Key	Value
-----	-------

Соотношение объема малых и больших данных



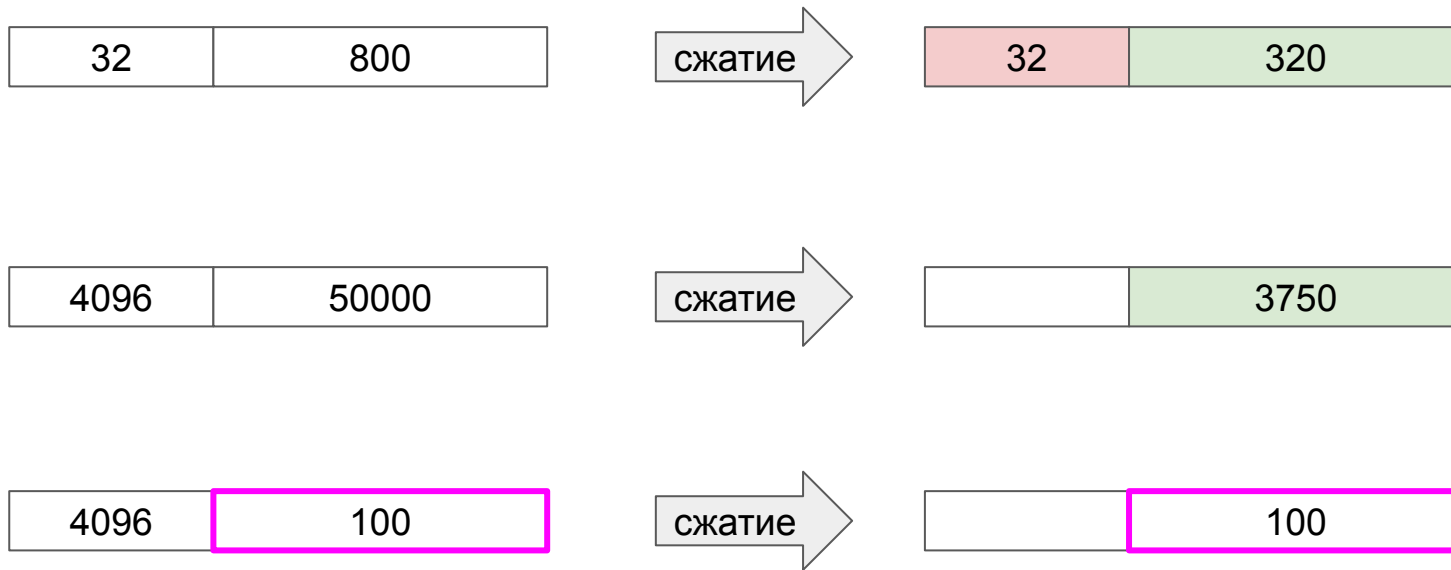
Эффективность сжатия больших данных



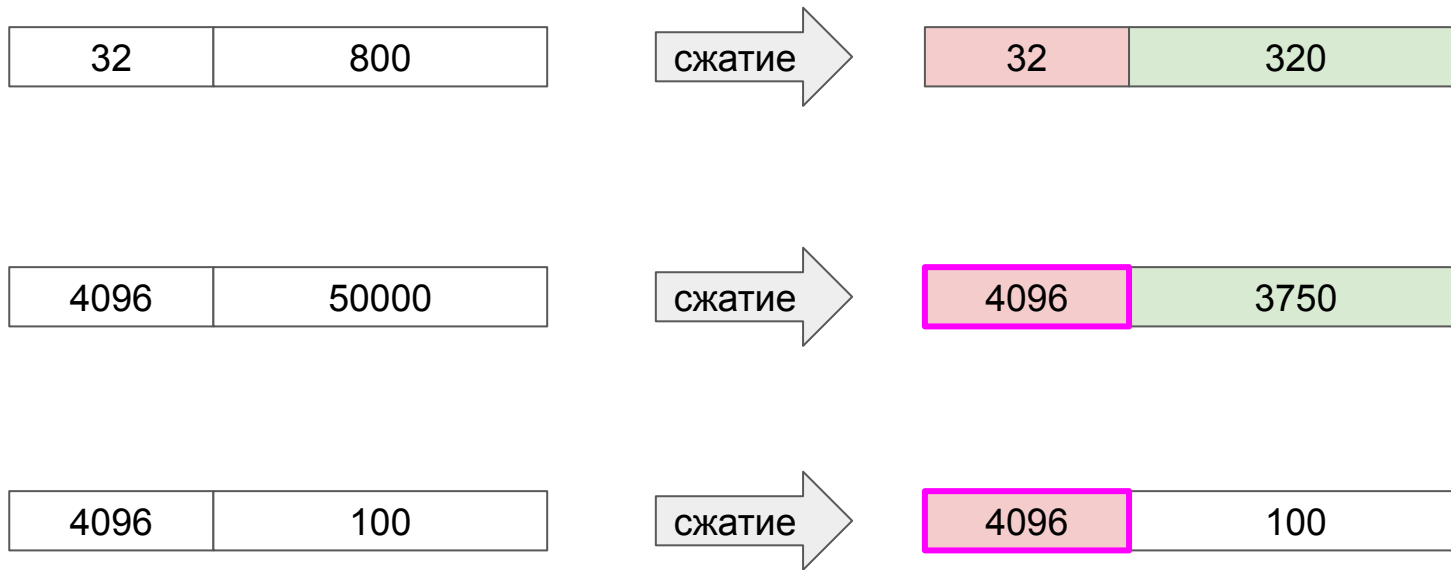
Эффективность сжатия огромных данных



Просто игнорируем небольшое

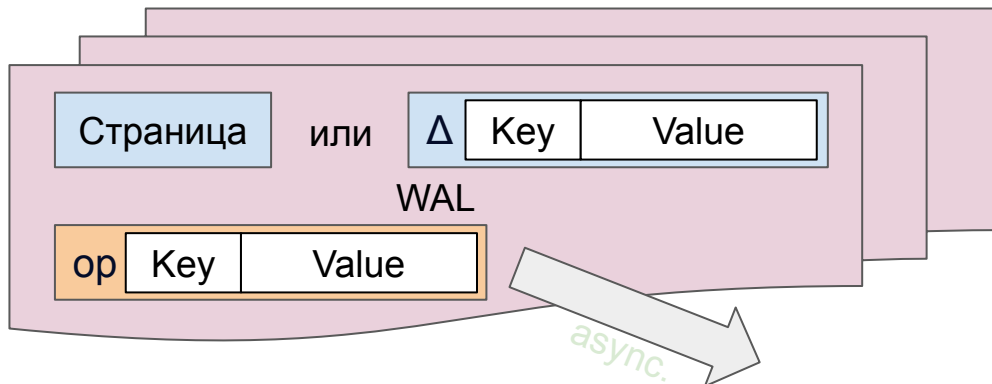
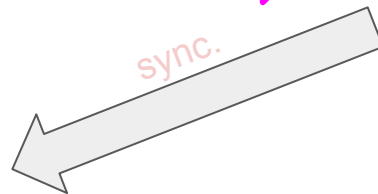
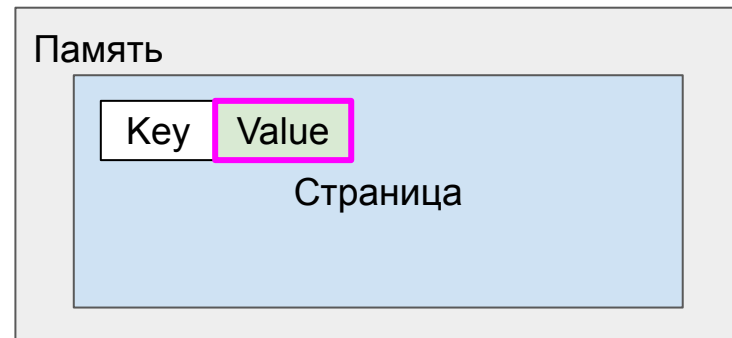
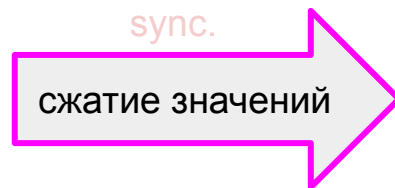
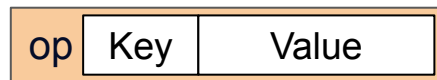


Ключи сжимать нельзя

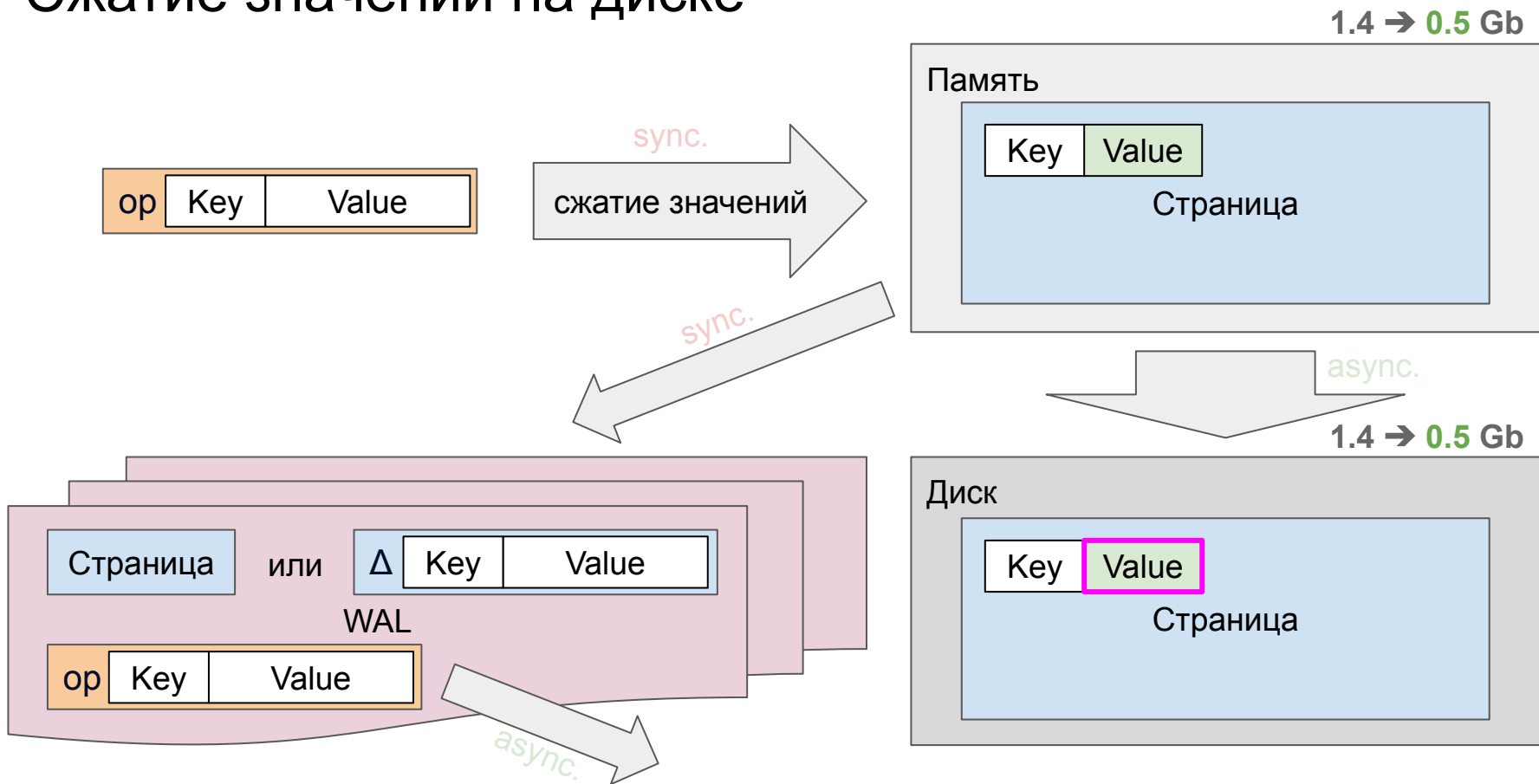


Сжатие значений в памяти

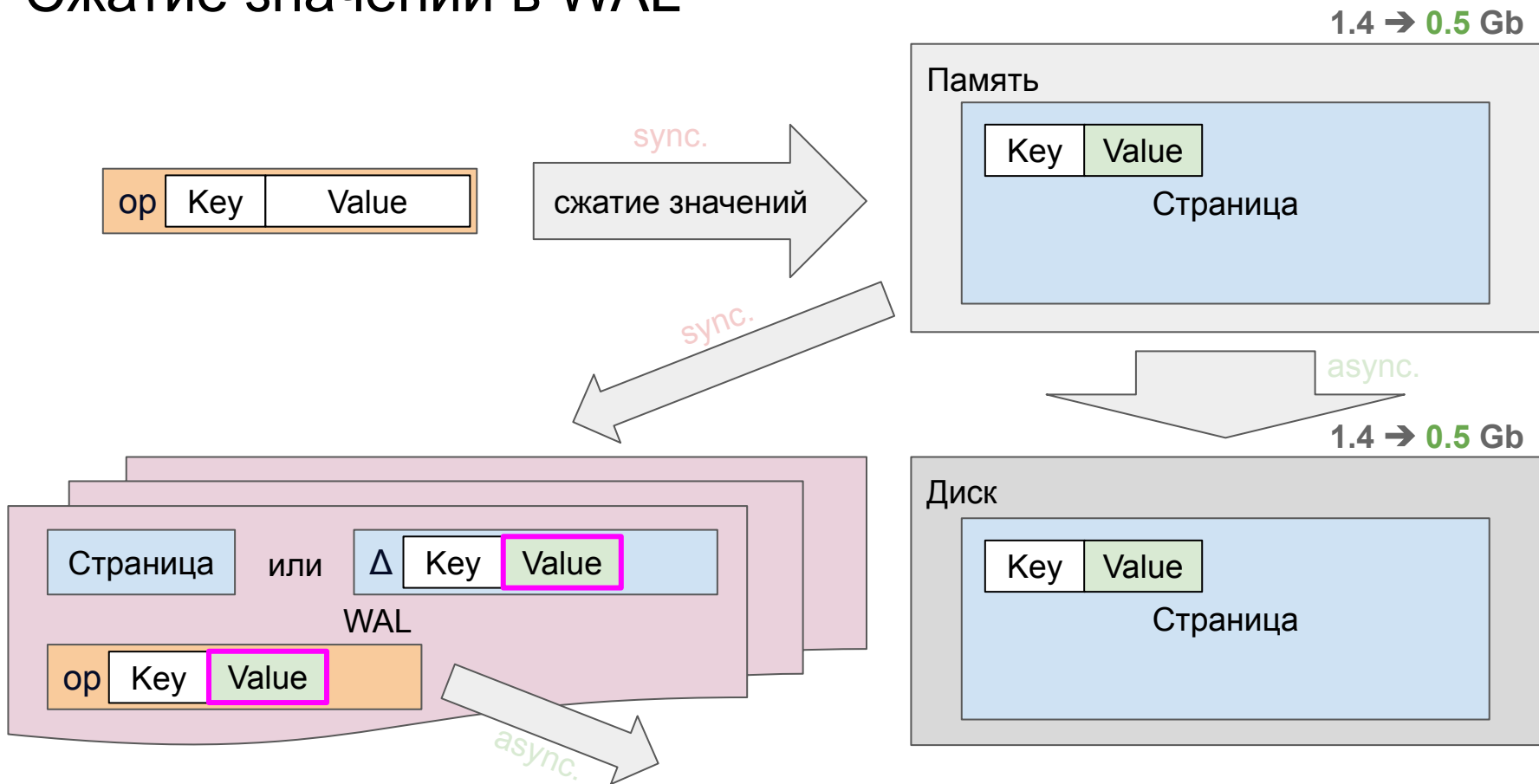
1.4 → 0.5 Gb



Сжатие значений на диске

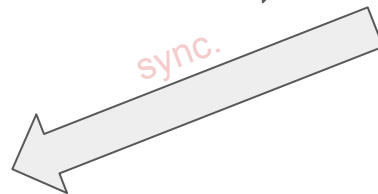
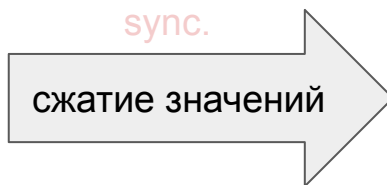
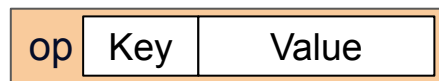


Сжатие значений в WAL

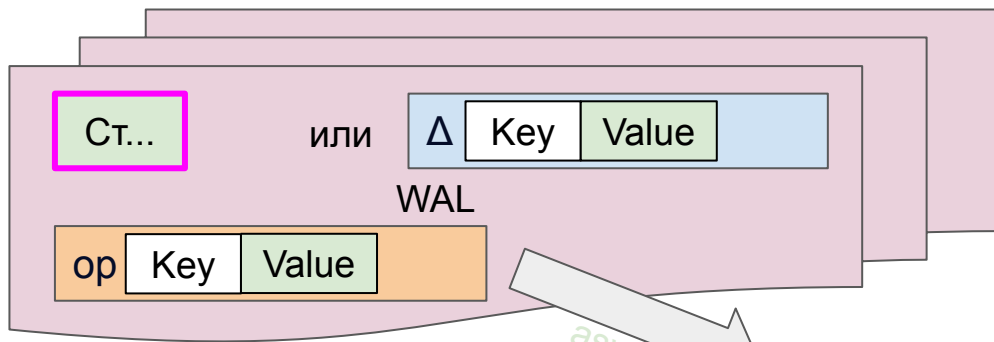


Сжатие страниц + значений в WAL

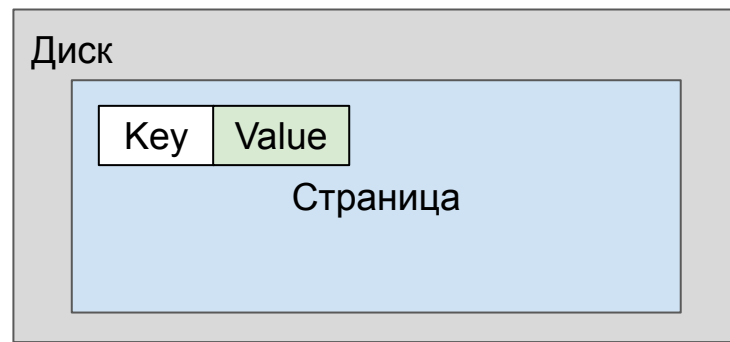
1.4 → 0.5 Gb



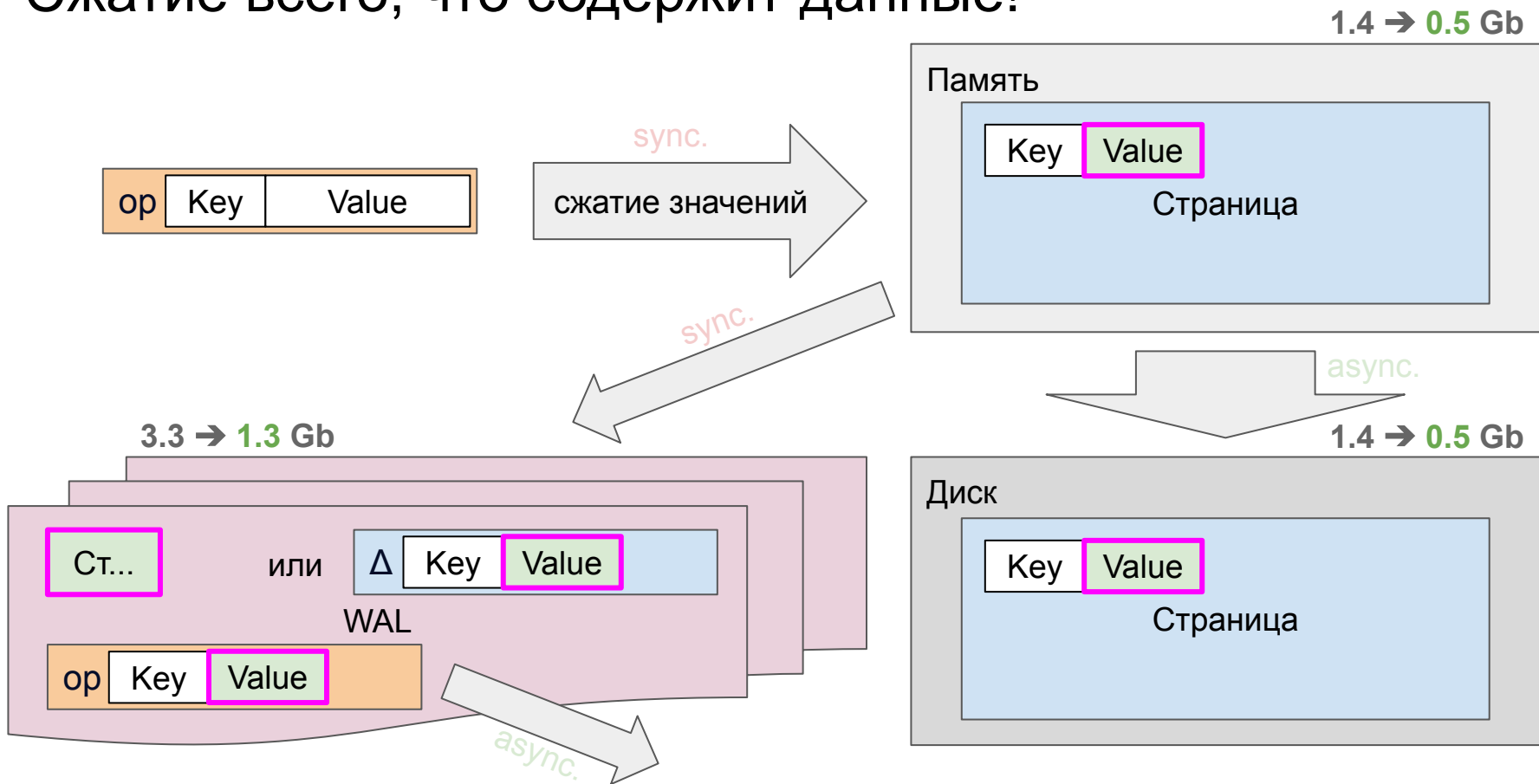
3.3 → 1.3 Gb



1.4 → 0.5 Gb



Сжатие всего, что содержит данные!



Сжатие на диске vs сжатие в памяти

	Без сжатия	Сжатие на диске	Сжатие в памяти
Память (Gb)	1.4	1.4	<u>0.5</u>
Диск (Gb)	1.4	0.3	0.5
WAL (Gb)	3.3	2.7	1.3
Latency, fast SSD (ms)	0.20	0.21	0.19
Latency, slow SSD (ms)	0.37	0.37	0.29

Сжатие на диске vs сжатие в памяти

	Без сжатия	Сжатие на диске	Сжатие в памяти
Память (Gb)	1.4	1.4	0.5
Диск (Gb)	1.4	0.3	<u>0.5</u>
WAL (Gb)	3.3	2.7	1.3
Latency, fast SSD (ms)	0.20	0.21	0.19
Latency, slow SSD (ms)	0.37	0.37	0.29

Сжатие на диске vs сжатие в памяти

	Без сжатия	Сжатие на диске	Сжатие в памяти
Память (Gb)	1.4	1.4	0.5
Диск (Gb)	1.4	0.3	0.5
WAL (Gb)	3.3	2.7	<u>1.3</u>
Latency, fast SSD (ms)	0.20	0.21	0.19
Latency, slow SSD (ms)	0.37	0.37	0.29

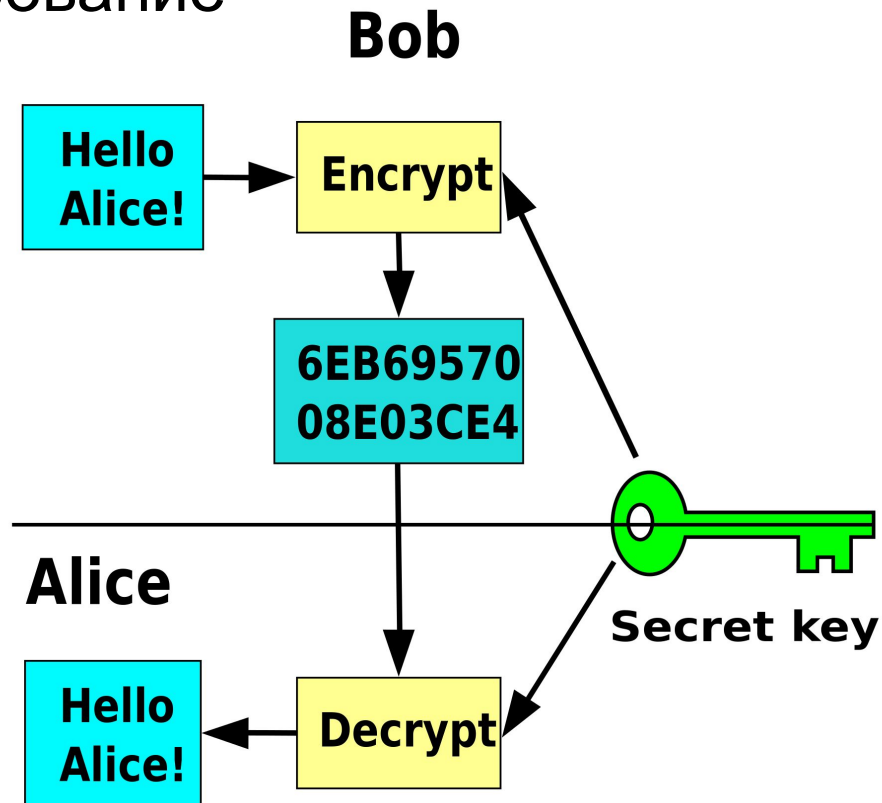
Сжатие на диске vs сжатие в памяти

	Без сжатия	Сжатие на диске	Сжатие в памяти
Память (Gb)	1.4	1.4	0.5
Диск (Gb)	1.4	0.3	0.5
WAL (Gb)	3.3	2.7	1.3
Latency, fast SSD (ms)	0.20	0.21	<u>0.19</u>
Latency, slow SSD (ms)	0.37	0.37	<u>0.29</u>
Latency, slow HDD (ms)	?	?	<u>?</u>

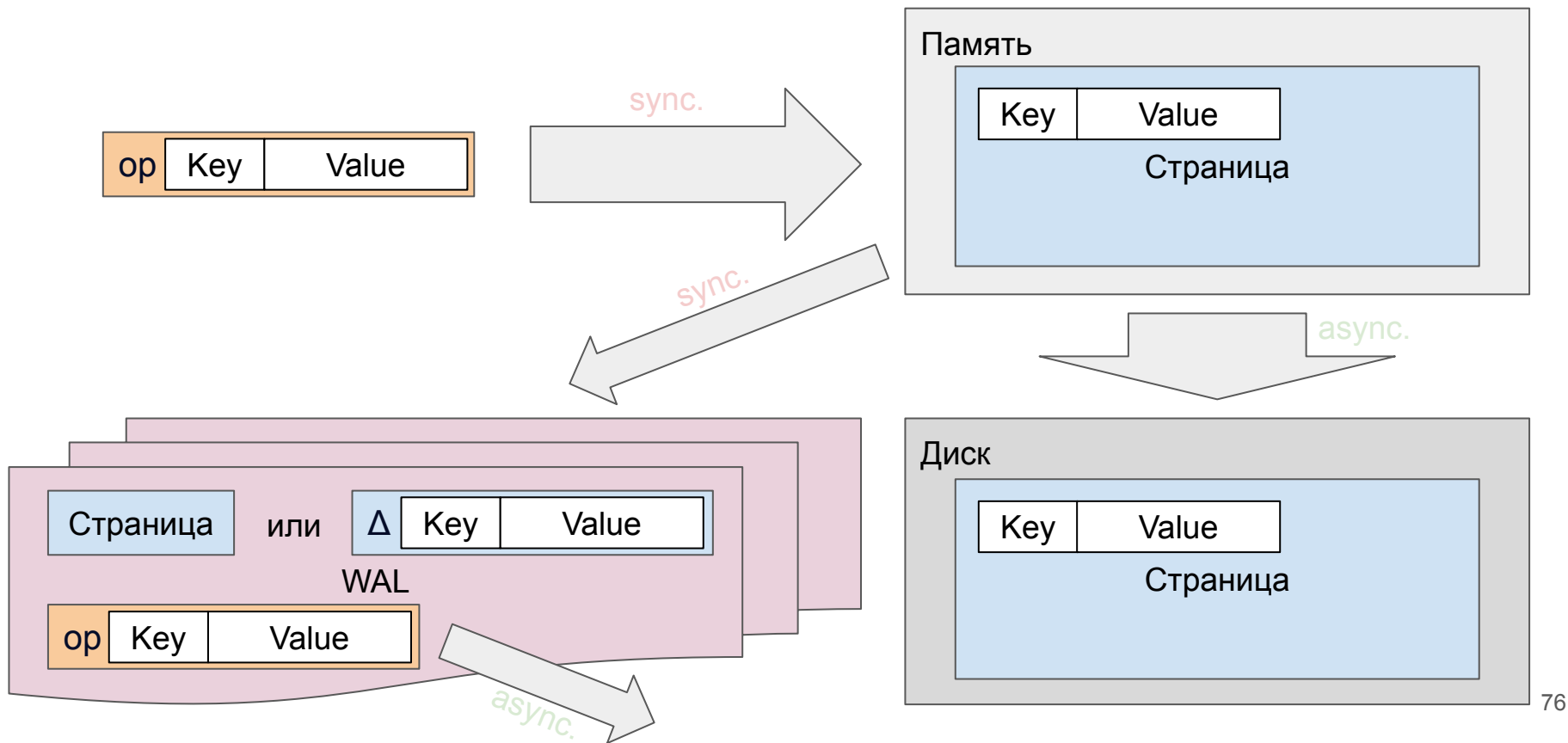
Нужны оба подхода!

	Без сжатия	Сжатие на диске	Сжатие в памяти
Память (Gb)	1.4	1.4	0.5
Диск (Gb)	1.4	<u>0.3</u>	<u>0.5</u>
WAL (Gb)	3.3	2.7	1.3
Latency, fast SSD (ms)	0.20	0.21	0.19
Latency, slow SSD (ms)	0.37	0.37	0.29

Сжатие != Шифрование



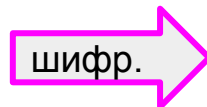
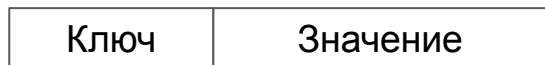
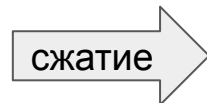
Куда приложить шифрование?



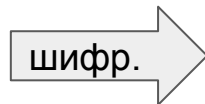
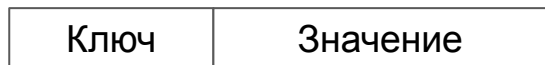
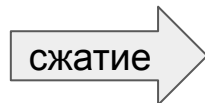
Сжимаем значения



Шифруем значения



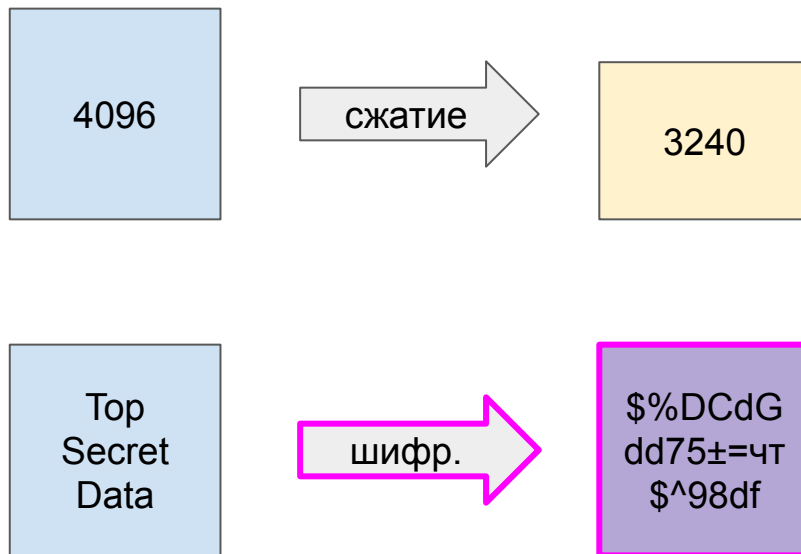
Шифруем значения



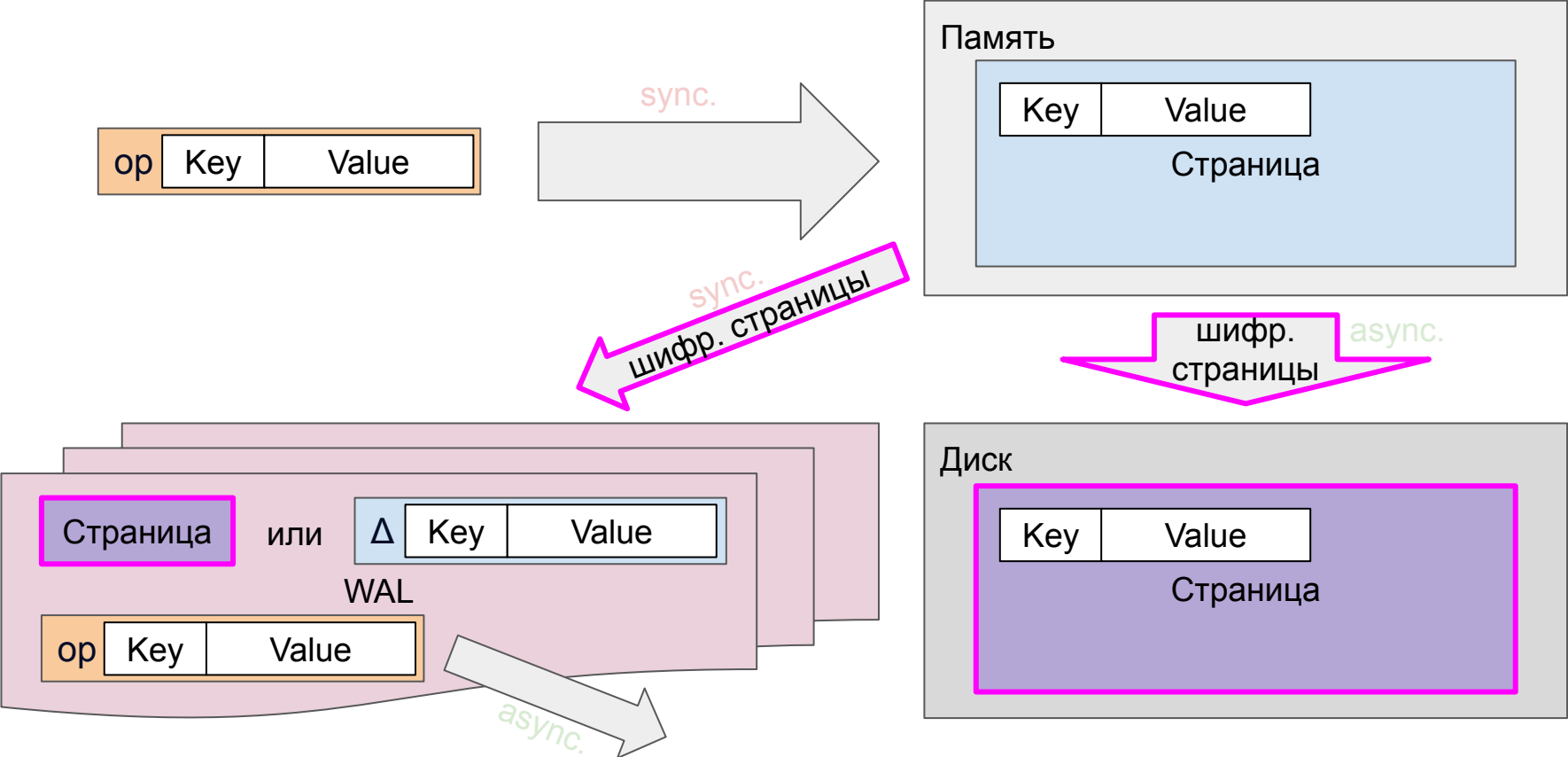
Сжимаем страницы



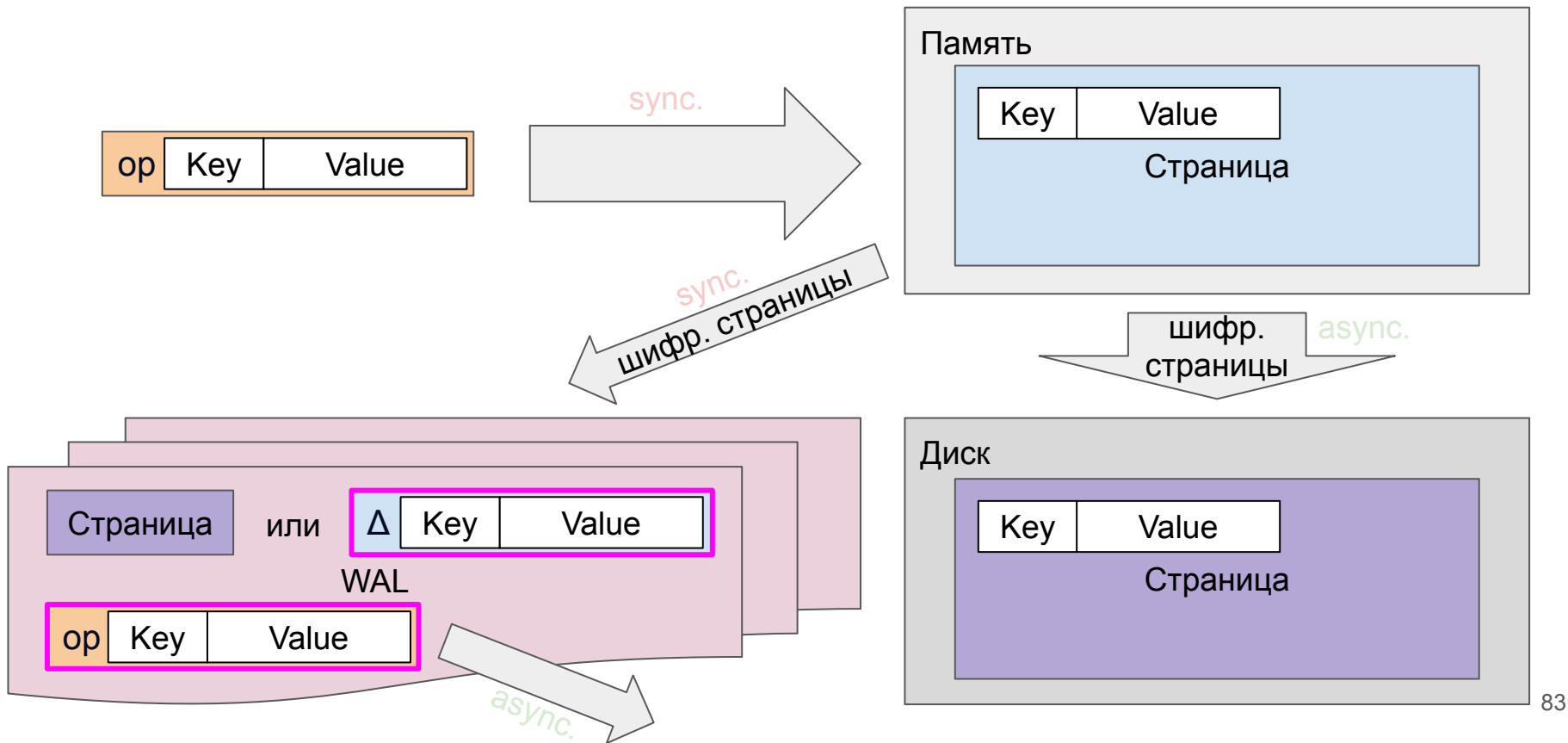
Шифруем страницы



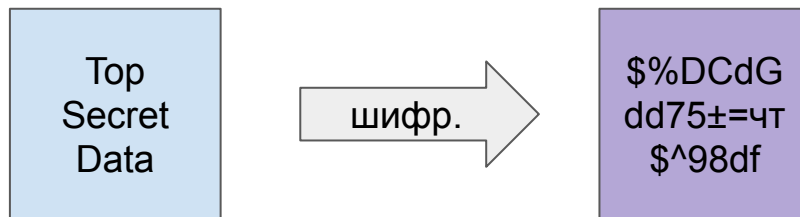
Шифрование страниц



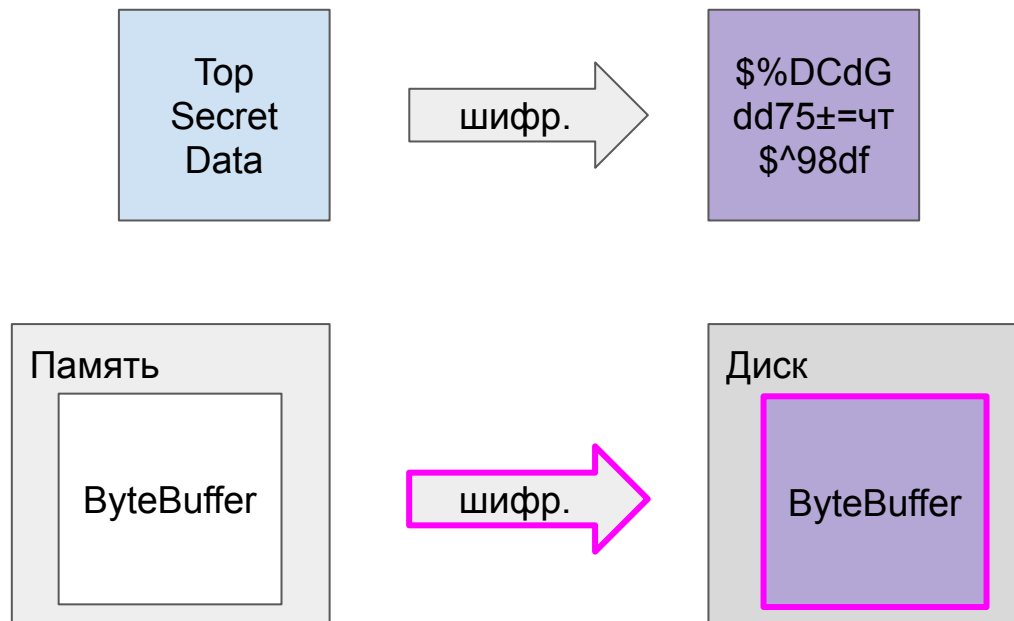
А дельты и логические операции?



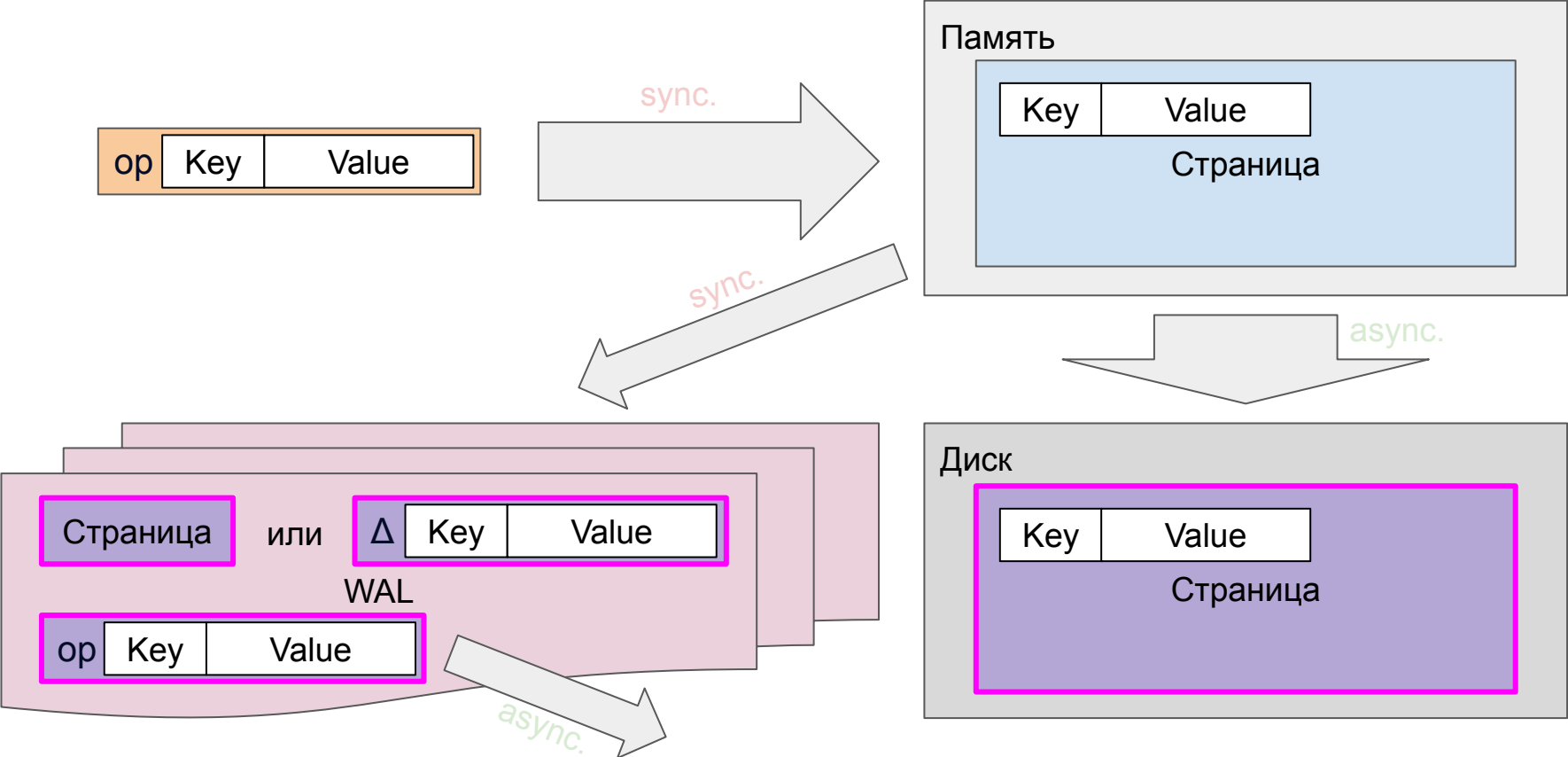
Шифруем страницы



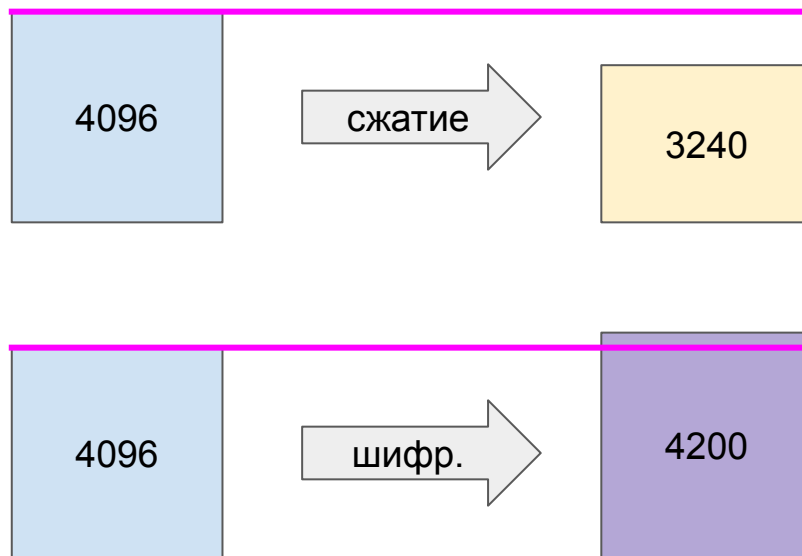
Шифруем данные перед записью на диск



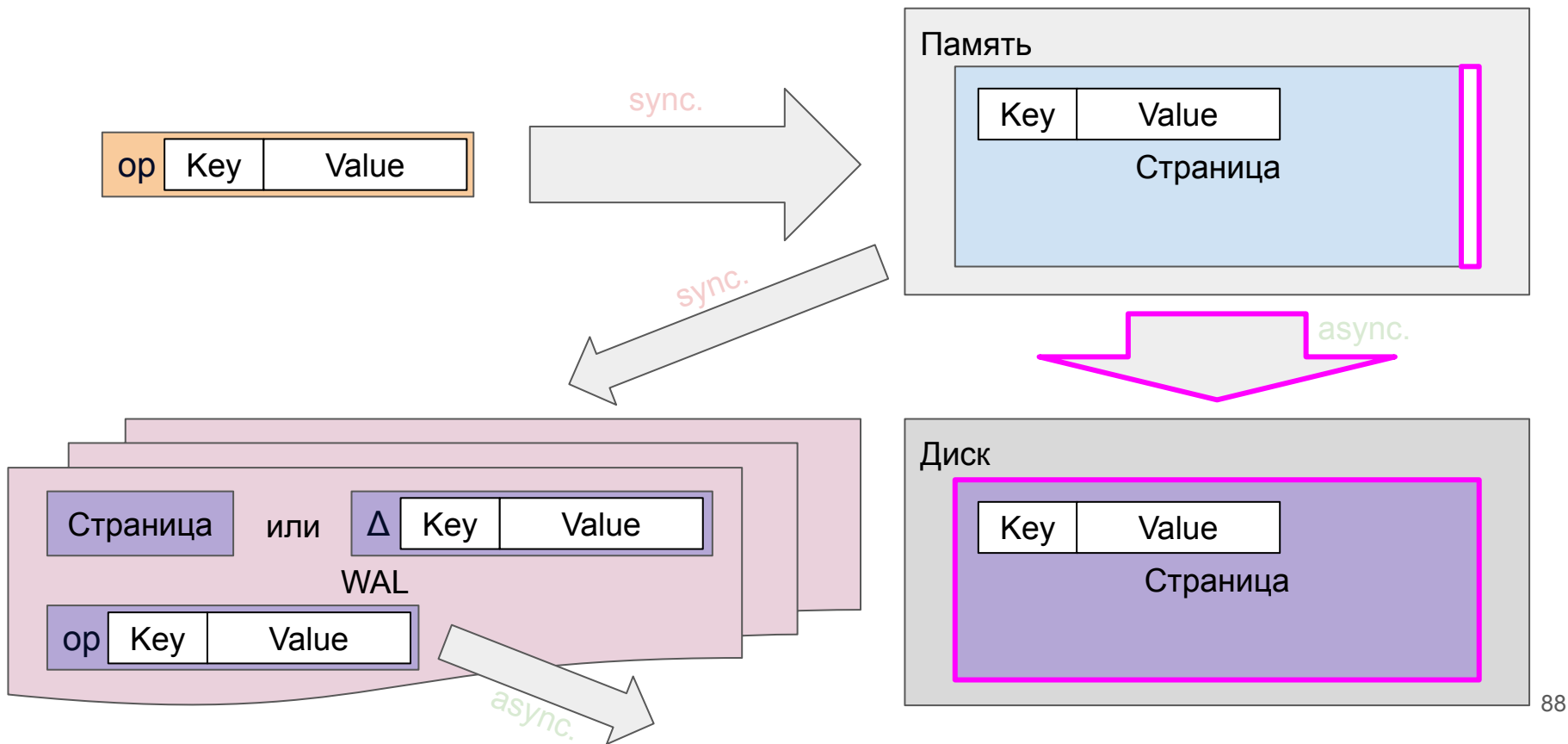
Шифрование всего что на диске



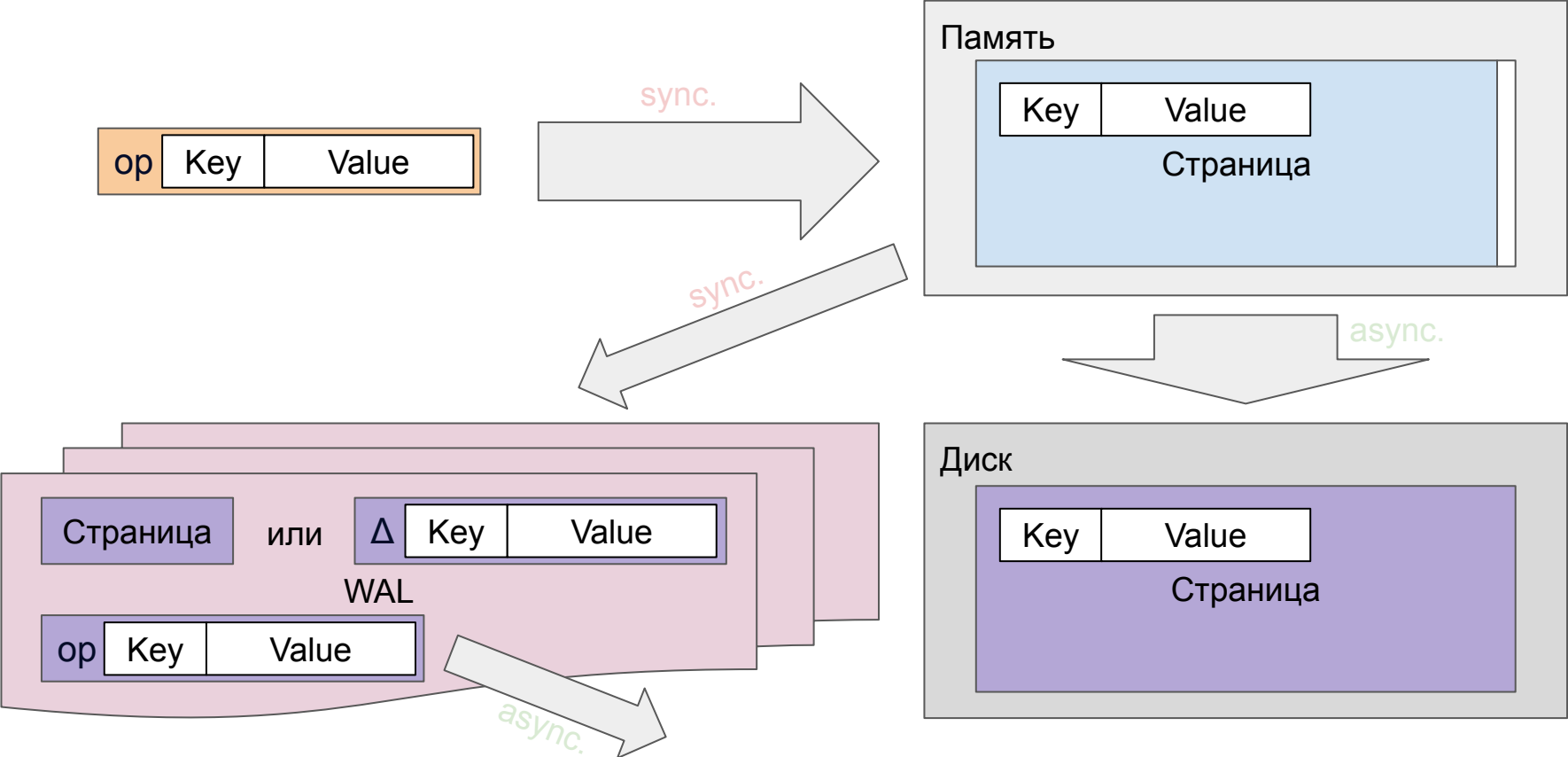
Шифрование vs сжатие



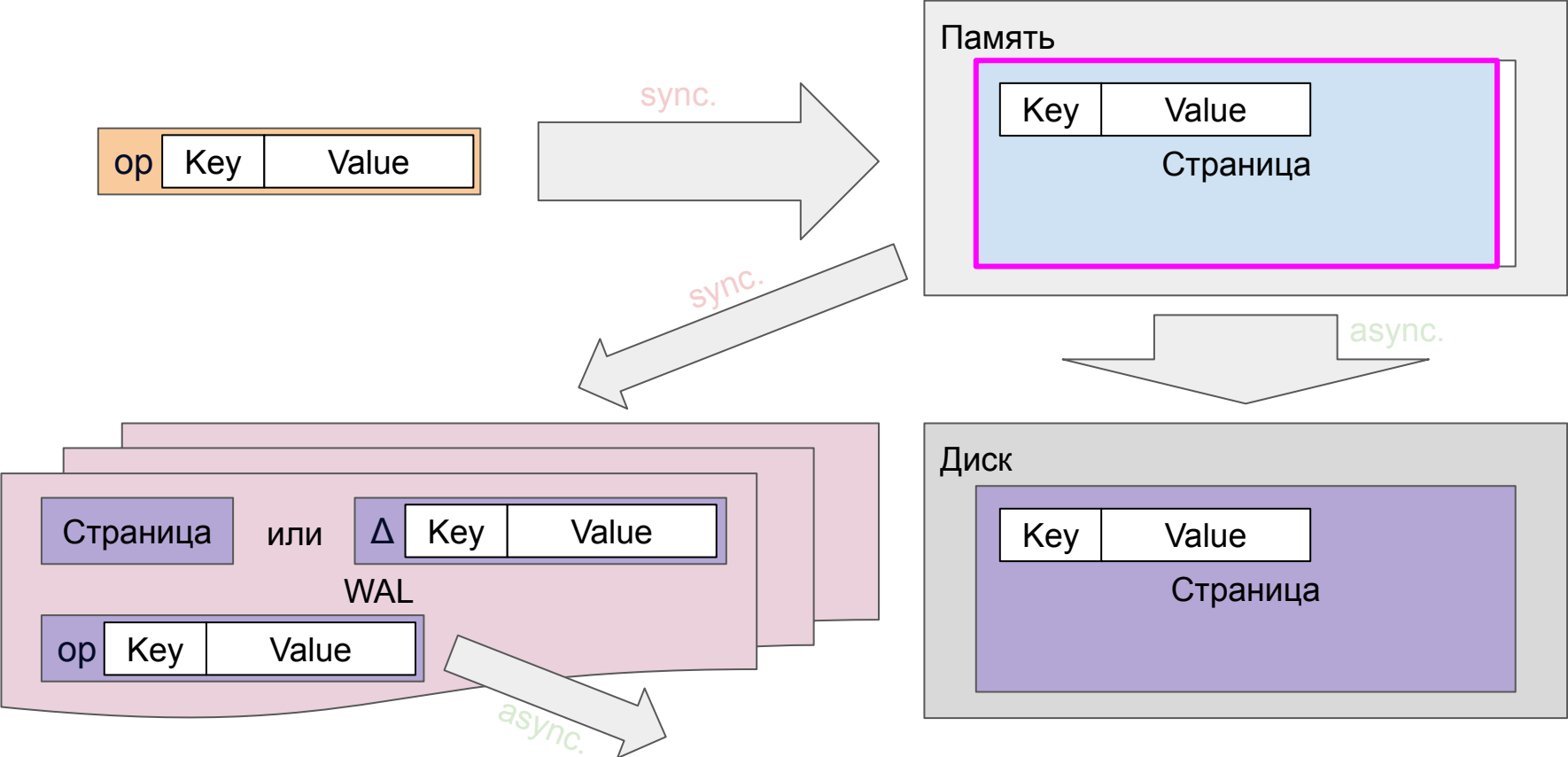
Готовимся к шифрованию



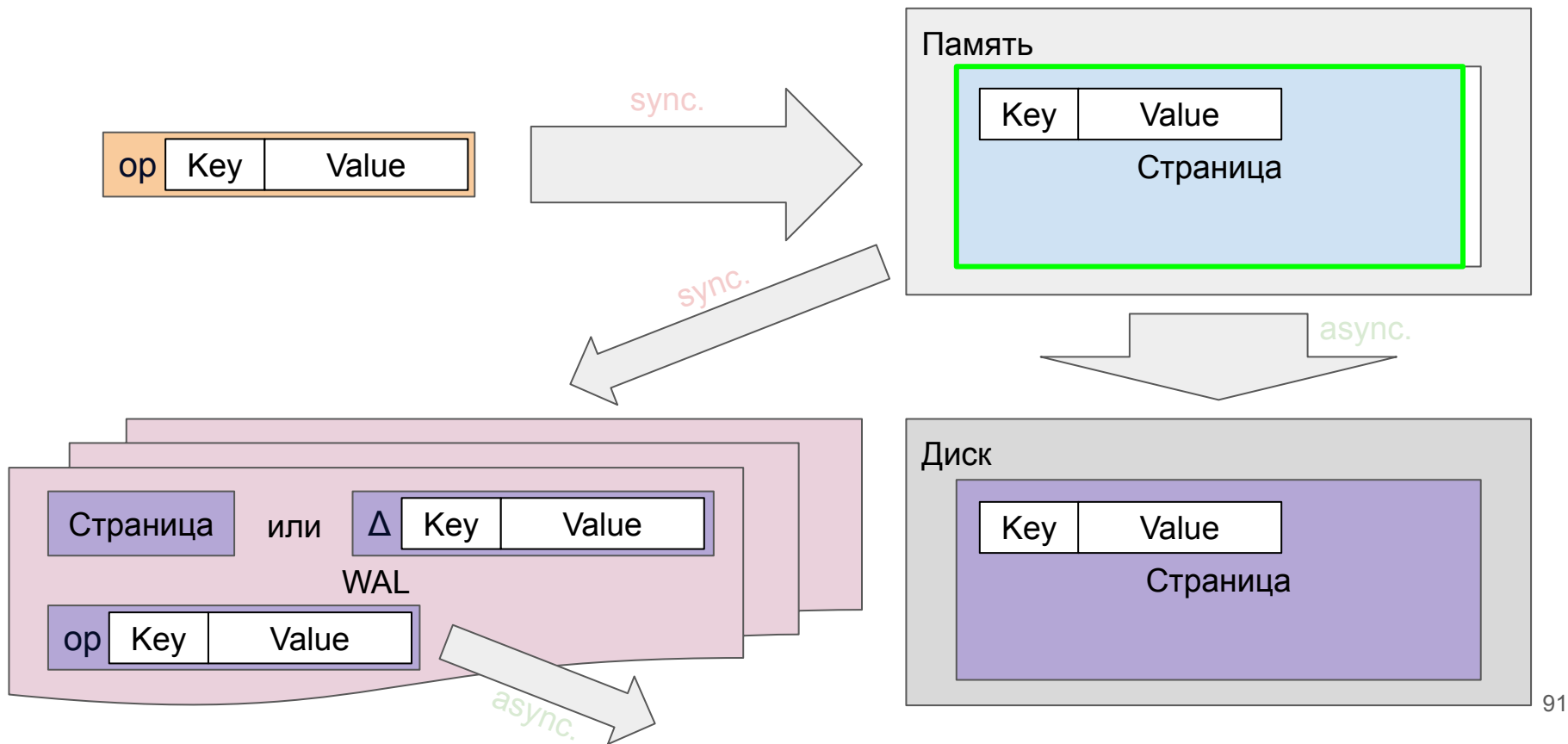
Шифрование всего, что на диске



Шифрование в памяти?



Шифрование в памяти?



Мораль

- Клиент должен быть счастлив



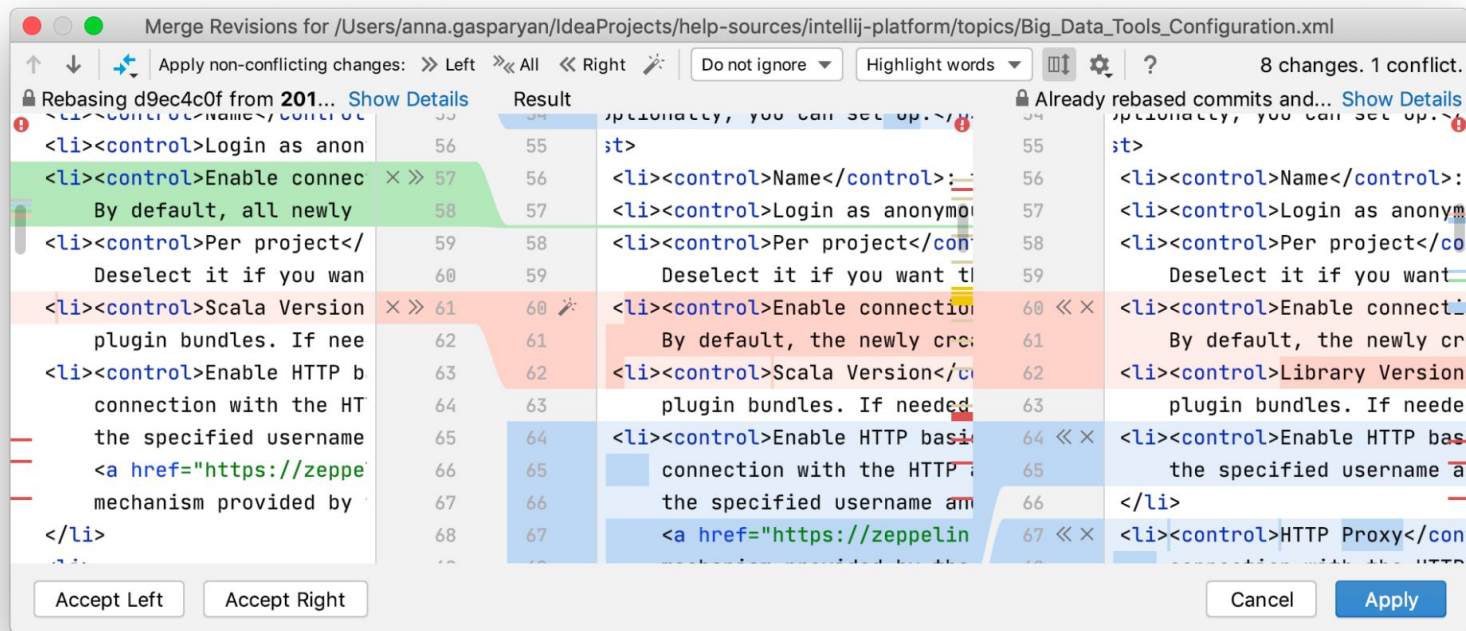
Мораль

- Клиент должен быть счастлив
- Open Source - это круто



Мораль

- Кастомер должен быть счастлив
- Open Source - это круто, если он честный



Мораль

- Клиент должен быть счастлив
- Open Source - это круто, если он честный, как у нас

Benefits of using Enterprise Open Source



Platform V
DataGrid



Техподдержка
Apache Ignite

