

CAP-теорема

Она гласит, что в распределенной системе можно обеспечить только два свойства из трех: согласованность, доступность и устойчивость к разделению. Помогает понимать, как конкретная распределенная система будет работать и чего от нее ожидать.

- **Согласованность данных (consistency)**

Когда во всех узлах в каждый момент времени данные согласованы друг с другом, то есть не противоречат друг другу. Если в одном из узлов в ячейке базы данных есть данные, такие же данные есть на всех остальных узлах.

- **Доступность (availability)**

Когда любой запрос может быть обработан системой, вне зависимости от ее состояния.

- **Устойчивость к разделению (partition tolerance)**

Когда расщепление системы на несколько изолированных секций не приводит к некорректному отклику от каждой из секций: отвалилась сеть между двумя узлами, но каждый из них может корректно отвечать своим клиентам.

Все распределенные базы данных так или иначе в лучшем случае реализуют два свойства из трех, жертвуя оставшимся. (Взято в [сайта](#))

Теперь, зная определения, распределим базы данных по типу (CA, AP, CP):

DragonFly - CA:

Благодаря непротиворечивым многопоточным процессам архитектура Dragonfly обеспечивает производительность, необходимую современным приложениям: миллионы операций в секунду, и все это из одного экземпляра.

- Архитектура Dragonfly рассчитана на вертикальное масштабирование на одной машине, что экономит командам затраты и сложность управления многоузловым кластером. Для наборов данных в памяти объемом до 1 ТБ Dragonfly предлагает самый простой и надежный масштаб на рынке.

Получается, что данная база данных сделана так, чтобы работала на одной машине. Значит разделять нам не на что и получаем отсутствие свойства устойчивости к разделению.

- Новый движок встроенной памяти Dragonfly, оптимизированный по пропускной способности, использует архитектуру "поток на ядро" без блокировок для обеспечения стабильной работы с низкими задержками. Реализуя настоящие асинхронные интерфейсы, Dragonfly в полной мере использует преимущества базового оборудования для обеспечения максимальной производительности.

Этот пункт как раз указывает на свойство доступности

- Dragonfly использует согласованную память во время моментального снимка, устраняя необходимость чрезмерного выделения памяти, что типично для Redis.

И последнее указывает нам на свойство согласованности данных

(Вся информация взята с [официальной странички продукта](#))

ScyllaDB - AP:

Scylla выбирает доступность и допуск к разделению, а не согласованность, так что:

- Невозможно быть одновременно последовательным и высокодоступным во время сетевого разделения;
- Если мы пожертвуем последовательностью, то сможем стать высокодоступными.

Вам нужно будет разработать свое приложение на основе моделирования данных Scylla, но конечным результатом будет приложение, которое никогда не выйдет из строя.

(Цитата с [официального сайта продукта](#))

Тут по сути анализировать нечего, так как производители сами отметили, что база данных обладает свойствами доступности и устойчивости к разделению

ArenadataBD - CP:

С Arenadata DB вы построите надёжное, масштабируемое корпоративное хранилище данных, которое будет расти вместе с вашими потребностями.

- Полное соответствие принципам строгой изоляции транзакции (принципы ACID). Одни и те же таблицы могут быть использованы для записи и чтения, без страха потерять данные. Уровень изоляции транзакций — Serializable.

Это означает, что теряется свойство доступности, так как выполняется строгая изоляция транзакций и может появляться задержка.

- Arenadata DB реализована на кластере из множества (от двух до сотен) серверов и равномерно распределяет нагрузку и данные между ними. Пользователю даже не обязательно знать, сколько именно и каких серверов используется. Достаточно работать с ADB как с обычной не кластерной системой.

Так как у нас распределяется нагрузка и данные по серверам, то выполняются свойства согласованности данных и устойчивости к разделению

(Взято с официального сайта продукта)