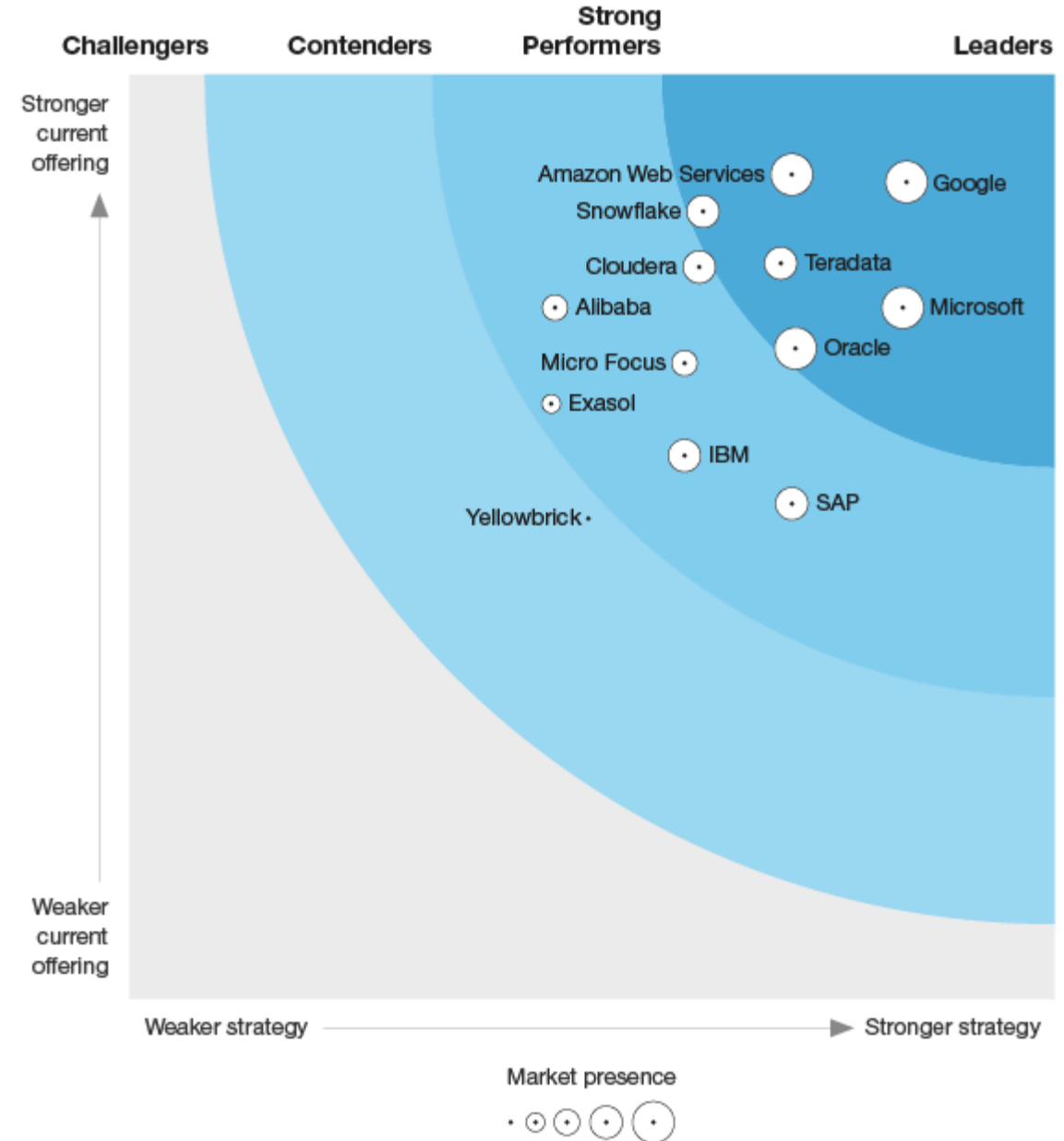


# **Характеристика Google BigQuery, доступ к облачному сервису**

# Характеристики BigQuery

«Заказчикам нравится ценность для бизнеса, архитектура, высокий уровень масштабирования, геопространственные возможности, мощные возможности AI / ML, хорошие возможности безопасности и широкие аналитические сценарии предлагаемые Google BigQuery», - говорится в отчете Forrester. Сегодняшним лидерам данных требуется платформа хранилищ данных, которая обеспечивает как глубину, так и широту охвата, а с помощью BigQuery организации могут раскрыть более глубокие возможности науки о данных и машинного обучения, одновременно способствуя демократизации данных и обеспечивая высочайший уровень доступности.



Сегодня клиенты по всему миру используют BigQuery для выполнения критически важных для бизнеса рабочих нагрузок аналитики, чтобы обеспечить ускорение бизнес-аналитики, аналитику Интернета вещей, аналитику клиентов, аналитику на основе AI / ML, науку о данных, совместную работу с данными и сервисы данных.

Forrester дал Google BigQuery оценку 5 из 5 по 19 различным критериям, в том числе:

Forrester criteria with full 5/5 scoring	Why this matters for Google Cloud's customers
<ul style="list-style-type: none"><li>✓ Data Lake Integration</li><li>✓ Data Ingestion</li><li>✓ Data Types Criteria</li></ul>	We help customers break down data silos through the investments we've made across data lake ingestion, data lake integration, and our support for various data types.
<ul style="list-style-type: none"><li>✓ High Availability/Disaster Recovery</li></ul>	Our industry-leading SLA of 99.99% uptime <sup>1</sup> enables our customers to have a highly available data warehouse that supports growing analytics demands and quick access to mission-critical insights.
<ul style="list-style-type: none"><li>✓ Performance features</li><li>✓ Scalability features</li></ul>	We enable petabyte-scale, real-time analytics and automatic High Availability (HA) over serverless infrastructure. BigQuery users have run more than 10,000 concurrent queries across their organization.
<ul style="list-style-type: none"><li>✓ ML/Data Science</li></ul>	Our investments in AI, including BigQuery ML's <sup>2</sup> SQL-based capabilities, give data analysts and data scientists a way to build and deploy ML models right within BigQuery.

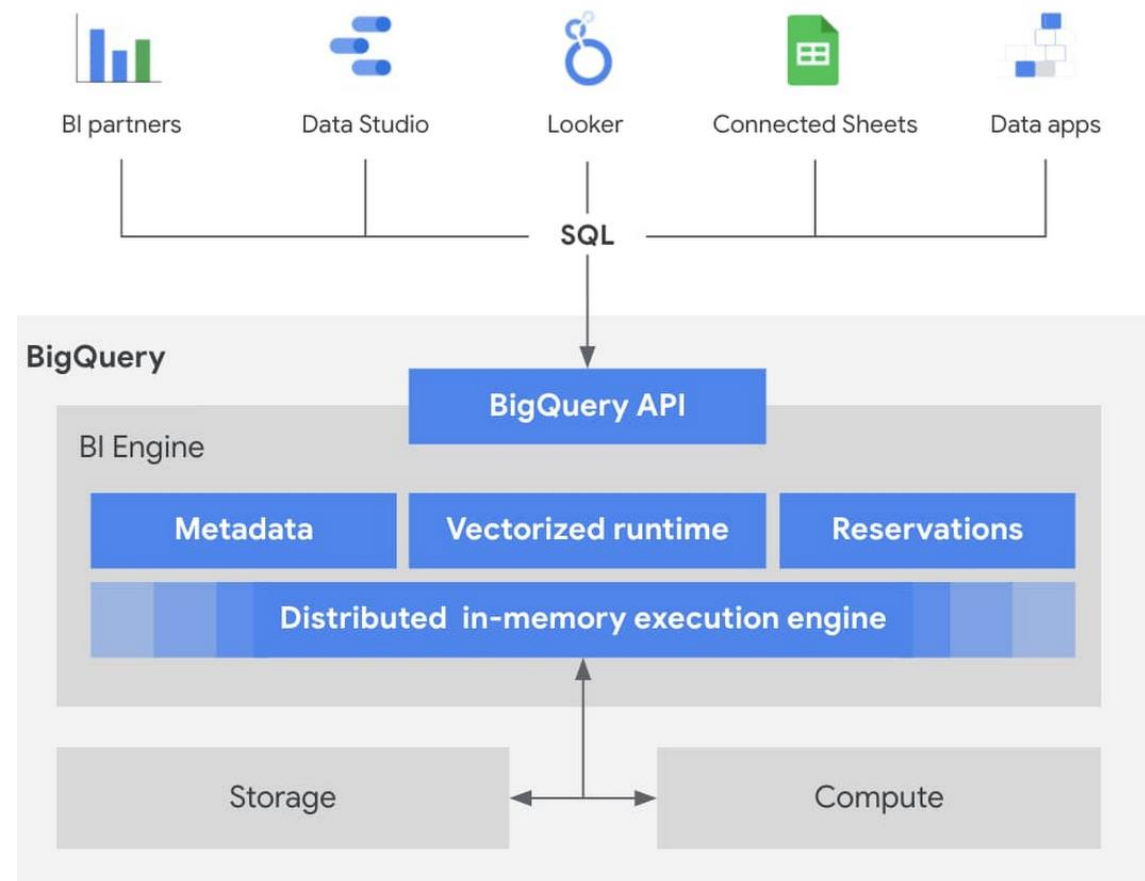
# Дополнительные преимущества

- Google - первый гипермасштабируемый провайдер, предлагающий решение для анализа данных в нескольких облаках. [С BigQuery Omni](#) заказчики могут с легкостью выполнять кросс-облачную аналитику и добиваться бизнес-результатов, которых они не могли достичь с помощью разрозненного подхода, предлагаемого другими поставщиками.
- BigQuery разработан так, чтобы он был хорошо масштабируемым, более открытым и совместимым, чтобы клиенты могли объединять данные из баз данных SQL, традиционных озер неструктурированных данных (в объектных хранилищах типа S3) и даже электронных таблиц с помощью любого инструмента анализа.
- Недавние инновации, такие как [BigQuery BI Engine](#), позволяют предоставлять лучшие аналитические возможности, обеспечивая время ответа на запросы менее секунды из любого инструмента бизнес-аналитики, от Google Looker и Connected Sheets до Tableau, Microsoft Power BI, Qlik Sense и других.
- С BigQuery организации получают как широту, так и глубину возможностей для преобразования своей аналитической стратегии.
- Эти преимущества помогают ускорить цифровую трансформацию и переосмыслить свой бизнес с помощью инноваций на основе данных.

# Преимущества BigQuery BI Engine

BI Engine теперь встроен в BigQuery API. Пользователи любого инструмента бизнес-аналитики или пользовательских приложений, которые подключаются к BigQuery API через JDBC/ODBC, теперь могут воспользоваться преимуществами BI Engine, зарезервировав емкость BI Engine в своих проектах GCP и указав размер памяти.

BI Engine включает распределенный механизм выполнения в памяти для обеспечения постоянной скорости работы. BI Engine использует современные методы, такие как векторизованная обработка, расширенное кодирование данных, адаптивное кэширование и планы запросов, настроенные для выполнения в памяти. Когда запросы превышают размер резервирования, BI Engine легко адаптируется к работе в смешанном режиме с помощью слотов BigQuery.



# Преимущества BigQuery BI Engine

- ❑ Упрощенная архитектура: быстро приступите к работе без управления сложными конвейерами данных или серверами. Традиционные системы бизнес-аналитики требуют, чтобы пользователи перемещали данные с платформ хранилищ данных в витрины данных или платформы бизнес-аналитики для поддержки быстрого интерактивного анализа. Обычно для этого требуются сложные конвейеры ETL для перемещения данных. Время, необходимое для выполнения этих заданий ETL, может задержать создание отчетов и поставить под угрозу актуальность данных для критически важных систем поддержки принятия решений. BI Engine выполняет анализ на месте в BigQuery. Это избавляет от необходимости перемещать данные или создавать сложные конвейеры преобразования данных.
- ❑ Интеллектуальная настройка: очень мало настроек конфигурации. Самонастраивающаяся конструкция: BI Engine автоматически настраивает запросы, перемещая данные между хранилищем в памяти BI Engine и хранилищем BigQuery, чтобы обеспечить оптимальную производительность и время загрузки информационных панелей. Администраторы BigQuery могут легко добавлять и удалять объем памяти BI Engine с шагом 1 ГБ с помощью облачной консоли.
- ❑ Сверхбыстрый: соответствие производительности и скорости бизнеса за счет сокращения времени на получение аналитических данных. BI Engine обеспечивает время ответа на запрос менее секунды с минимальным временем загрузки и улучшенным параллелизмом для данных, хранящихся в BigQuery.

**How to speed up  
BI workloads on BigQuery**  
**in 2 steps & <30 seconds!**



# Преимущества BigQuery BI Engine

- ❑ Упрощенная архитектура: быстро приступите к работе без управления сложными конвейерами данных или серверами. Традиционные системы бизнес-аналитики требуют, чтобы пользователи перемещали данные с платформ хранилищ данных в витрины данных или платформы бизнес-аналитики для поддержки быстрого интерактивного анализа. Обычно для этого требуются сложные конвейеры ETL для перемещения данных. Время, необходимое для выполнения этих заданий ETL, может задержать создание отчетов и поставить под угрозу актуальность данных для критически важных систем поддержки принятия решений. BI Engine выполняет анализ на месте в BigQuery. Это избавляет от необходимости перемещать данные или создавать сложные конвейеры преобразования данных.
- ❑ Интеллектуальная настройка: очень мало настроек конфигурации. Самонастраивающаяся конструкция: BI Engine автоматически настраивает запросы, перемещая данные между хранилищем в памяти BI Engine и хранилищем BigQuery, чтобы обеспечить оптимальную производительность и время загрузки информационных панелей. Администраторы BigQuery могут легко добавлять и удалять объем памяти BI Engine с шагом 1 ГБ с помощью облачной консоли.



- ❑ Сверхбыстрый: соответствие производительности и скорости бизнеса за счет сокращения времени на получение аналитических данных. BI Engine обеспечивает время ответа на запрос менее секунды с минимальным временем загрузки и улучшенным параллелизмом для данных, хранящихся в BigQuery.

# Анализ данных BigQuery с помощью записных книжек Kaggle



**Kaggle** интегрирован в **BigQuery**, корпоративное облачное хранилище данных Google Cloud. Эта интеграция означает, что пользователи BigQuery могут выполнять сверхбыстрые SQL-запросы, обучать модели машинного обучения с помощью SQL и анализировать их с помощью ядер, бесплатной размещенной среды блокнотов Jupyter от Kaggle.

Используя вместе ядра BigQuery и Kaggle, можно использовать интуитивно понятную среду разработки для запроса данных BigQuery и выполнения машинного обучения без необходимости перемещать или загружать данные. Как только ваша учетная запись Google Cloud связана с записной книжкой или сценарием Kernels, можно составлять запросы прямо в записной книжке с помощью клиентской библиотеки API BigQuery, запускать ее в BigQuery и выполнять практически любой анализ данных оттуда. Например, вы можете импортировать новейшие библиотеки науки о данных, такие как Matplotlib, scikit-learn и XGBoost, для визуализации результатов или обучения новейшим моделям машинного обучения. Более того, есть бесплатная возможность подключения графических процессоров, до 16 ГБ оперативной памяти и девять часов времени выполнения.





Google  
BigQuery



kaggle

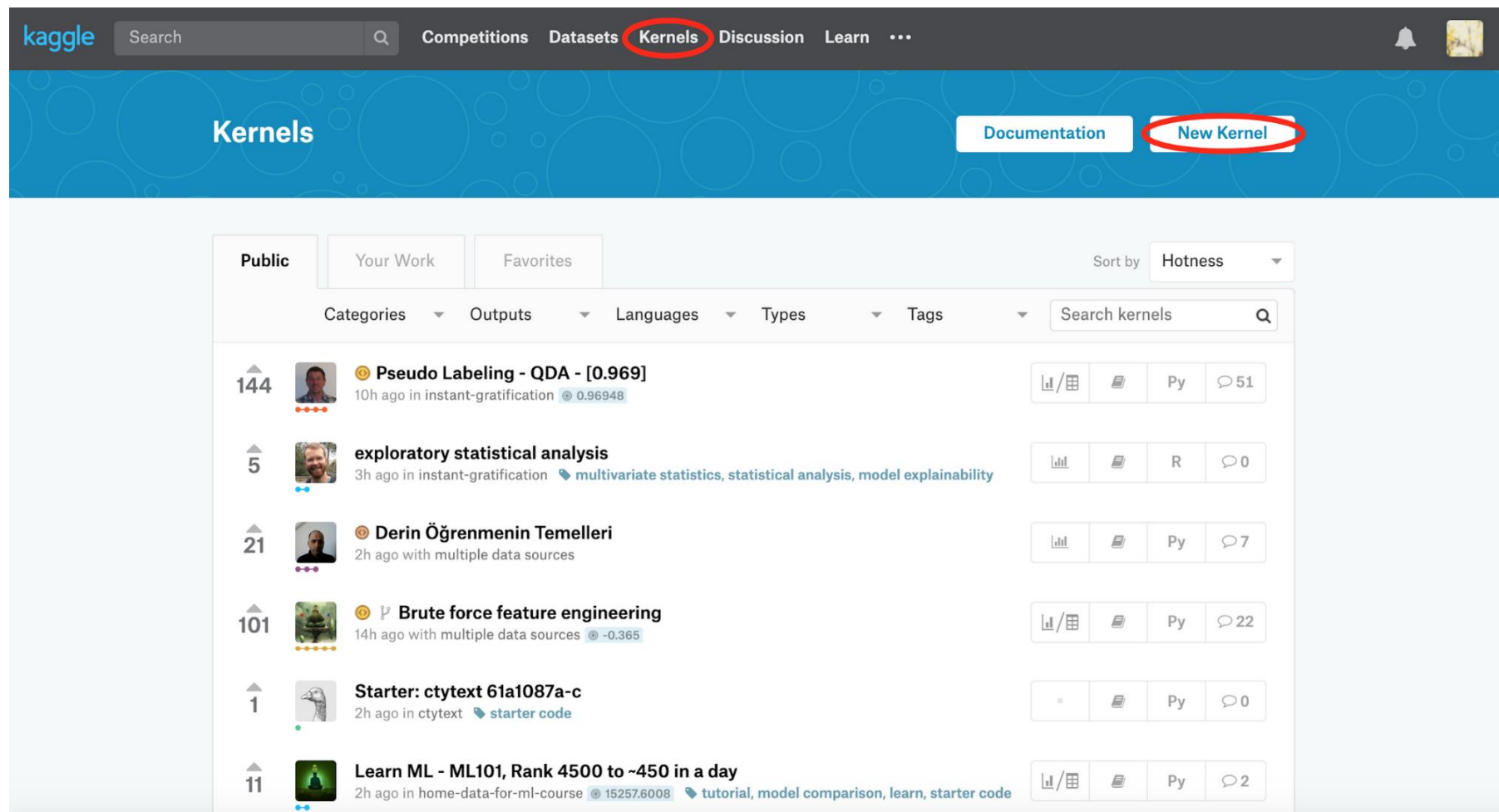
Для пользователей BigQuery наиболее заметным преимуществом является то, что теперь существует широко используемая интегрированная среда разработки (IDE) - ядра Kaggle, которая позволяет выполнять запросы и анализ данных в одном месте. Это превращает фрагментированный рабочий процесс анализа данных в более цельный процесс по сравнению с предыдущим способом, когда вы сначала запрашиваете данные в редакторе запросов, а затем экспортируете данные в другое место для завершения анализа.

Кроме того, Kaggle - это платформа для обмена, которая позволяет легко сделать ваши ядра общедоступными. Kaggle позволяет вам распространять вашу работу с открытым исходным кодом, а также обсуждать науку о данных с первоклассными специалистами в области науки о данных.

Чтобы впервые начать работу с BigQuery, включите свою учетную запись в песочнице BigQuery, которая предоставляет до 10 ГБ бесплатного хранилища, 1 терабайт в месяц обработки запросов и 10 ГБ запросов на создание модели BigQuery ML.

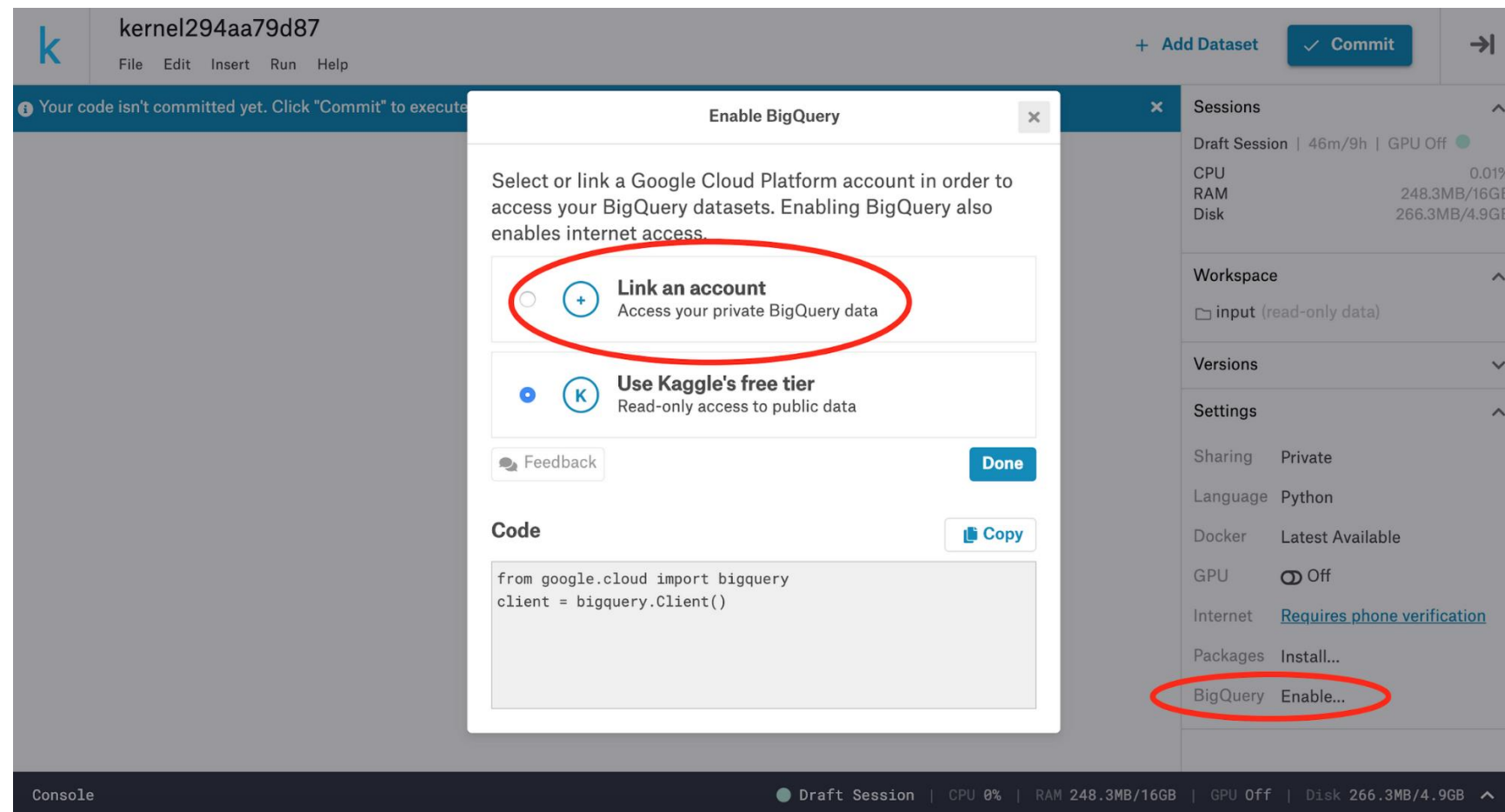


Чтобы начать анализировать наборы данных BigQuery в ядрах, зарегистрируйтесь в Kaggle. После входа в систему нажмите «Kernels» на верхней панели, а затем «New kernel», чтобы немедленно запустить новый сеанс IDE. Kaggle предлагает ядра двух типов: сценарии и записные книжки. В этом примере выбран вариант записных книжек.





В среде редактора ядер свяжите свою учетную запись BigQuery с учетной записью Kaggle, нажав «BigQuery» на правой боковой панели, а затем нажмите «Связать учетную запись». Как только ваша учетная запись будет связана, вы сможете получить доступ к своим собственным наборам данных BigQuery с помощью клиентской библиотеки BigQuery API .





Давайте попробуем это, используя набор данных Ames Housing, который общедоступен на Kaggle. Этот набор данных содержит 79 независимых переменных, описывающих (почти) все аспекты жилых домов в Эймсе, штат Айова, а также их окончательную продажную цену. Давайте составим запрос, чтобы получить представление о данных. Мы хотим выяснить, какие типы домов представлены в этом наборе данных, а также в скольких из них установлено или не установлено центральное кондиционирование воздуха. Вот как выглядит запрос:

# Tutorial: How to use BigQuery in Kaggle Kernels Draft saved

File Edit Insert Run Help

+ Add Dataset

Commit

```

# Write the query
query1 = """
    SELECT
        DISTINCT HouseStyle AS HousingStyle,
        CentralAir AS HasAirConditioning,
        COUNT(HouseStyle) AS Count
    FROM
        `my-example-housing-dataset.ameshousing.train`
    GROUP BY
        HousingStyle,
        HasAirConditioning
    ORDER BY
        HousingStyle,
        HasAirConditioning DESC
    """

```

```

In[7]:
# Set up the query
query_job1 = client.query(query1)

# Make an API request to run the query and return a pandas DataFrame
housestyleAC = query_job1.to_dataframe()

# See the resulting table made from the query
print(housestyleAC)

```

	HousingStyle	HasAirConditioning	Count
0	1.5Fin	True	136
1	1.5Fin	False	18
2	1.5Unf	True	10
3	1.5Unf	False	4
4	1Story	True	685
5	1Story	False	41
6	2.5Fin	True	7
7	2.5Fin	False	1
8	2.5Unf	True	4
9	2.5Unf	False	7

Sessions

Draft Session | 11m/9h | GPU Off

CPU 0.05%  
RAM 482.2MB/16GB  
Disk 266.3MB/4.9GB

Workspace

☐ input (read-only data)

Versions

1 uncommitted draft  
 Jessica Li's draft

4 committed versions

	V4	12h ago	0	0	✓
	V3	13h ago	+70	-48	✓
	V2	16h ago	+66	-12	✓
	V1	1d ago	+51	0	✓

Settings

Sharing Public

Language Python

Docker Latest Available

GPU ☐ Off

Internet ☐ On

Packages Install...

BigQuery Enabled

Console

Draft Session | CPU 0% | RAM 482.2MB/16GB | GPU Off | Disk 266.3MB/4.9GB





Google  
BigQuery



kaggle

## Построение моделей машинного обучения с использованием запросов SQL

Помимо анализа данных, BigQuery ML позволяет создавать и оценивать модели машинного обучения с помощью запросов SQL. С помощью нескольких запросов любой специалист по данным может построить и оценить модели регрессии без глубоких знаний фреймворков машинного обучения или языков программирования.

Всего одним запросом можно создать модель машинного обучения на основе SQL внутри ядер. Вы можете продолжать использовать ядра для создания более сложных запросов для анализа и оптимизации вашей модели для получения лучших результатов. Вы даже можете опубликовать свое ядро, чтобы поделиться им с сообществом Kaggle и в Интернете после завершения анализа.

```
%%bigquery
```

```
CREATE MODEL IF NOT EXISTS `bqml_tutorial.sample_model`  
OPTIONS(model_type='logistic_reg') AS
```

```
SELECT
```

```
  IF(totals.transactions IS NULL, 0, 1) AS label,  
  IFNULL(device.operatingSystem, "") AS os,  
  device.isMobile AS is_mobile,  
  IFNULL(geoNetwork.country, "") AS country,  
  IFNULL(totals.pageviews, 0) AS pageviews
```

```
FROM
```

```
  `bigquery-public-data.google_analytics_sample.ga_sessions_*
```

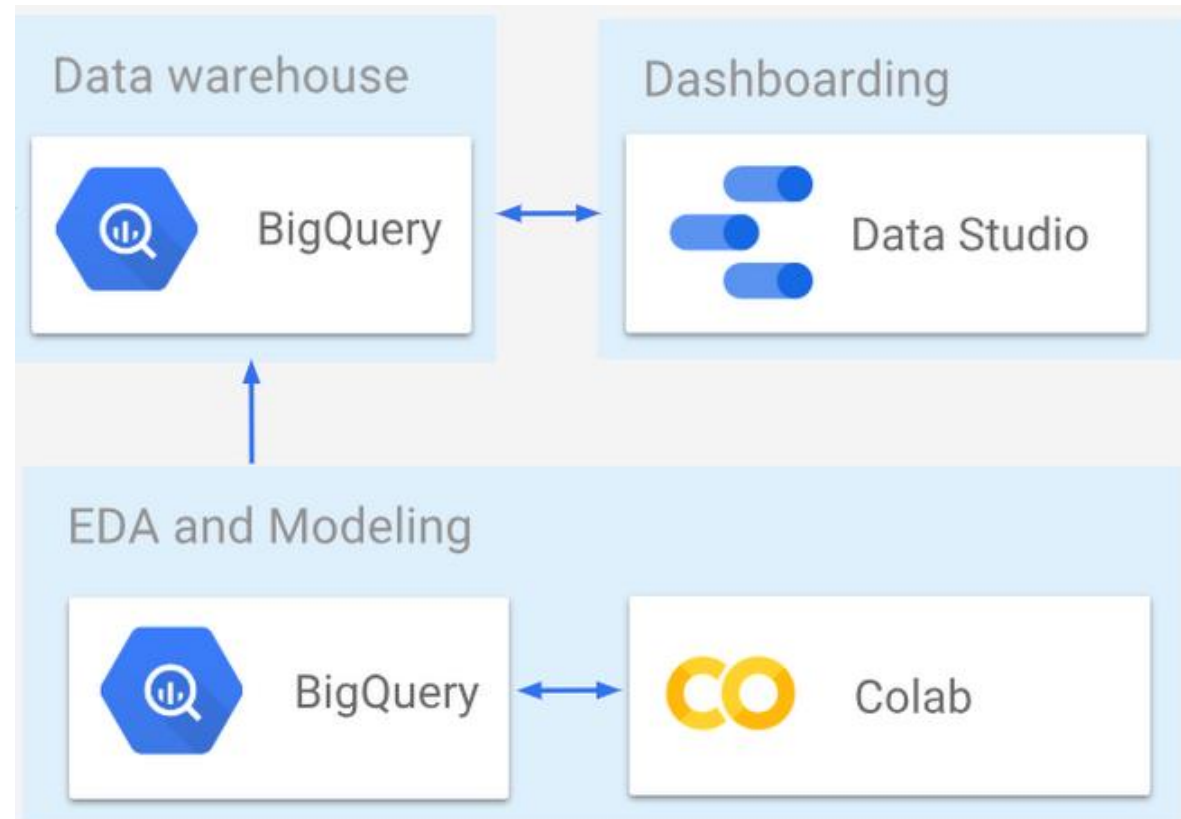
```
WHERE
```

```
  _TABLE_SUFFIX BETWEEN '20160801' AND '20170630'
```



## Выделим два эффективных инструмента для интерактивного анализа больших данных:

- ❑ BigQuery , высокомасштабируемое корпоративное хранилище данных Google Cloud Platform (которое включает общедоступные наборы данных для изучения)
- ❑ Colab , бесплатная среда Jupyter для ноутбуков на основе Python, которая полностью работает в облаке и объединяет текст, код и выходные данные в одном документе.



BigQuery полезен для хранения и запроса (с помощью SQL) очень больших наборов данных. Python хорошо работает с BigQuery, обладая функциями параметризации и записи запросов различными способами, а также библиотеками для перемещения наборов данных между фреймами данных pandas и таблицами BigQuery. И Colab специально помогает улучшить результаты BigQuery с помощью таких функций, как интерактивные формы, таблицы и визуализации, используя такие модули, как Pandas, Seaborn и Numpy и др., для анализа данных.



# Что такое BigQuery?

Хранение и запрос массивных наборов данных может занять много времени и дорого без правильного оборудования и инфраструктуры. BigQuery - это **корпоративное хранилище данных**, которое решает эту проблему, обеспечивая сверхбыстрые запросы SQL с использованием вычислительной мощности инфраструктуры Google. Просто переместите свои данные в BigQuery, и позвольте нам взять на себя тяжелую работу. Вы можете контролировать доступ как к проекту, так и к своим данным, исходя из потребностей вашего бизнеса, например, предоставляя другим возможность просматривать или запрашивать ваши данные.

# Анатомия запроса BigQuery

Google BigQuery - это молниеносная аналитическая база данных. Клиенты считают, что производительность BigQuery позволяет экспериментировать с огромными наборами данных без компромиссов. Но насколько на самом деле быстр BigQuery? И что нужно для достижения скорости BigQuery? Давайте проверим общедоступные образцы bigquery: wikipedia\_benchmark, в частности таблицу Wiki100B. Эта таблица содержит 100 миллиардов строк и имеет размер около 7 терабайт.

Запустим  
типичный  
SQL-  
запрос:

100B Benchmark with 3 wildcards ? Query Editor UDF Editor X

```
1 SELECT language, SUM(views) as views
2 FROM [bigquery-samples:wikipedia_benchmark.Wiki100B]
3 WHERE REGEXP_MATCH(title, "G.*o.*o.*g")
4 GROUP BY language
5 ORDER BY views desc;
```

No Cached Results X

RUN QUERY Save Query Save View Format Query Show Options

Query complete (24.7s elapsed, 4.06 TB processed)

✓

Вы можете видеть, что этот запрос выполняется менее чем за 30 секунд. Это впечатляет, поскольку для обработки такого количества данных BigQuery пришлось:

- Прочитать около 1 ТБ данных, затем распаковать их до 4 ТБ (при условии сжатия ~ 4:1)
- Выполнить 100 миллиардов регулярных выражений с 3 символами в каждом
- Передать 1,25 ТБ данных по сети (1 ТБ сжат для начального чтения и 0,25 ТБ для агрегирования)

Предположим на секунду, что все механизмы распределенной аналитики используют одинаковое количество ресурсов для обработки запроса и что запросы можно полностью распараллелить. При прочих равных, если нужно было построить распределенный кластер, способный выполнять этот запрос так быстро, нужно было развернуть как минимум:

- Около 330 выделенных жестких дисков 100 МБ / с для чтения 1 ТБ данных
- Сеть 330 Gіgabit для передачи 1,25 ТБ данных
- 3300 ядер для распаковки 1 ТБ данных и обработки 100 миллиардов регулярных выражений по 1 мкс

Это много ресурсов! Так что впечатляет, что BigQuery позволяет использовать все это всего за несколько секунд, необходимых для выполнения этой работы.

Но еще более впечатляющим является то, что мы не знаем, что это происходит - мы просто нажимаем «Выполнить запрос», и BigQuery автоматически позаботится обо всем остальном. BigQuery полностью скрывает сложность крупномасштабных аналитических технологий.

Мы просто создали проект Google Cloud, включили биллинг и выполнили этот SQL-запрос в BigQuery. Нам не нужно было развертывать какое-либо программное обеспечение, виртуальные машины или кластеры. Нам не нужно было определять какие-либо индексы, ключи, ключи сортировки или какие-либо аналогичные задачи администрирования баз данных.

Эта способность легко использовать огромные вычислительные ресурсы **демократизирует аналитику больших данных.**

# BigQuery под капотом

Мы провели предварительную оценку того, что потребуется для выполнения типичного большого запроса в другом месте менее чем за 30 секунд:

- 330 жестких дисков 100 МБ / с; 3300 ядер и сеть 330 Gigabit

Это очень оптимистические предположения, игнорирующие множество сложностей:

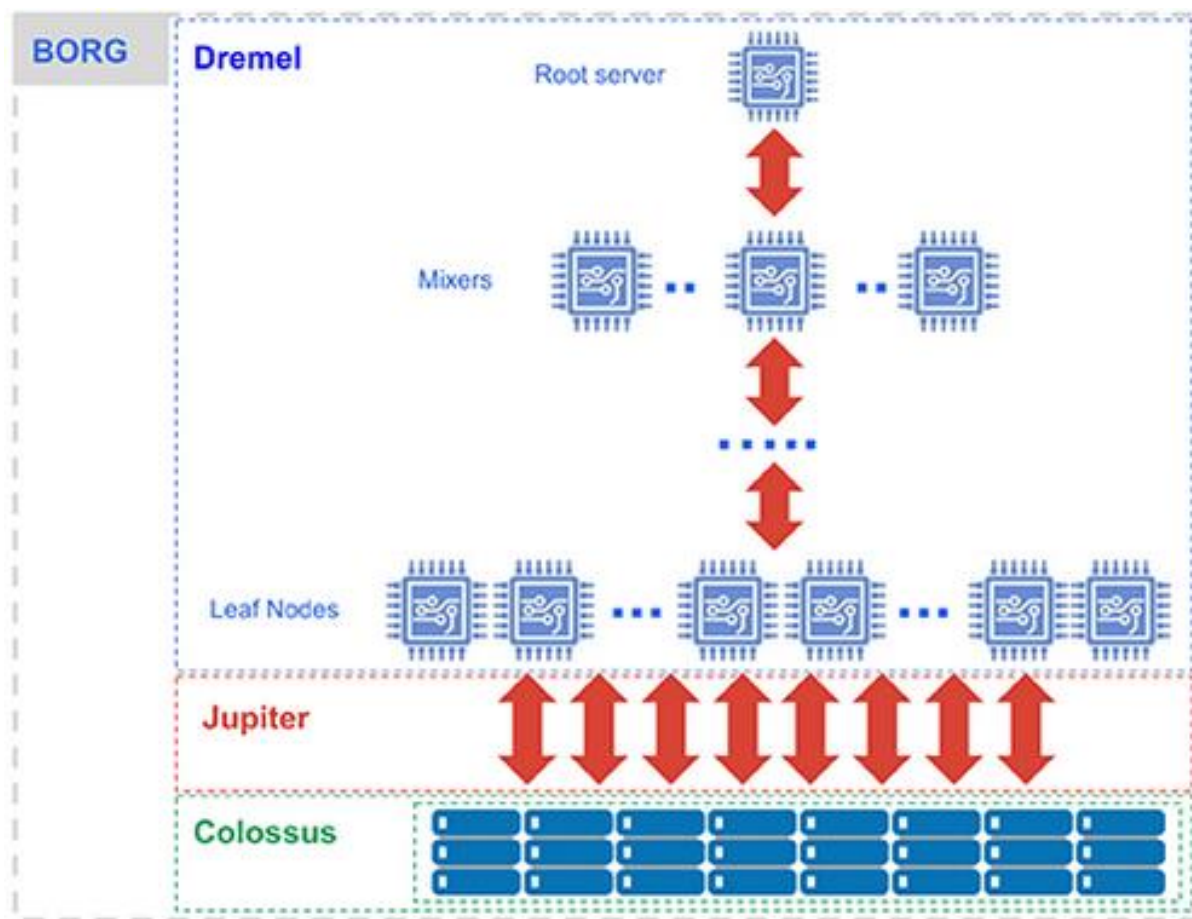
- Даже если есть данные на 300 дисках, данные должны быть идеально распределены между этими дисками, вы должны иметь возможность читать на полной скорости со всех этих дисков одновременно, а в противном случае каждый диск должен быть простаивающим. И не забудьте учесть несколько факторов репликации для избыточности.
- Потребуется согласовать такое количество процессоров, чтобы они запускались одновременно и выполняли одинаковый объем работы. Предполагая, что ~ 100 32-процессорных машин, один из серверов будет выходить из строя в среднем каждый день, что приведет к отключению всех 3300 процессоров, поэтому вам потребуется дополнительная координация для обработки этих сбоев без замедления, включая развертывание дополнительной вычислительной избыточности, предпочтительно между несколькими зонами.
- Чтобы так быстро координировать такую большую пропускную способность сети, нужно убедиться, что пропускная способность сети равномерно распределена по каждому экземпляру и другие сложности.

# BigQuery под капотом

BigQuery работает в нескольких центрах обработки данных, каждый из которых имеет сотни тысяч ядер, десятки петабайт емкости хранилища и терабайты пропускной способности сети.

## Dremel: двигатель исполнения

- Чтобы запросы выполнялись быстро, требуется больше, чем просто много оборудования. Запросы BigQuery обрабатываются механизмом запросов Dremel, которая организует ваш запрос, разбивая его на части и повторно собирая результаты.





# BigQuery под капотом

Dremel превращает ваш SQL-запрос в дерево выполнения. Листья дерева он называет «слотами» и выполняет тяжелую работу по чтению данных из Колосса и выполнению любых необходимых вычислений. В примере из последнего сообщения слоты читают 100 миллиардов строк и проверяют регулярное выражение для каждой из них.

Ветви дерева - это «смесители», которые выполняют агрегирование. Между ними есть «случайный выбор», который использует сеть Google Jupiter для чрезвычайно быстрого перемещения данных из одного места в другое. Все микшеры и слоты находятся в ведении Borg, который распределяет аппаратные ресурсы.

Dremel динамически распределяет слоты по запросам по мере необходимости, поддерживая распределение между несколькими пользователями, которые все запрашивают одновременно. Один пользователь может получить тысячи слотов для выполнения своих запросов.

Dremel широко используется в Google - от поиска до рекламы, от YouTube до Gmail, - поэтому большое внимание уделяется постоянному совершенствованию Dremel. Пользователи BigQuery получают возможность непрерывного улучшения производительности, надежности, эффективности и масштабируемости без простоев и обновлений, связанных с традиционными технологиями.

# Колосс: Распределенное хранилище

BigQuery использует [Colossus](#), распределенную файловую систему Google последнего поколения. Каждый центр обработки данных Google имеет свой собственный кластер Colossus, и каждый кластер Colossus имеет достаточно дисков, чтобы предоставить каждому пользователю BigQuery тысячи выделенных дисков одновременно. Colossus также выполняет репликацию, восстановление (при сбое дисков) и распределенное управление (поэтому нет единой точки отказа). Colossus достаточно быстр, чтобы позволить BigQuery обеспечивать аналогичную производительность для многих баз данных в памяти, но с использованием гораздо более дешевой, но хорошо распараллеленной, масштабируемой, надежной и производительной инфраструктуры.

# Колосс: Распределенное хранилище

BigQuery использует [Colossus](#), распределенную файловую систему Google последнего поколения. Каждый центр обработки данных Google имеет свой собственный кластер Colossus, и каждый кластер Colossus имеет достаточно дисков, чтобы предоставить каждому пользователю BigQuery тысячи выделенных дисков одновременно. Colossus также выполняет репликацию, восстановление (при сбое дисков) и распределенное управление (поэтому нет единой точки отказа). Colossus достаточно быстр, чтобы позволить BigQuery обеспечивать аналогичную производительность для многих баз данных в памяти, но с использованием гораздо более дешевой, но хорошо распараллеленной, масштабируемой, надежной и производительной инфраструктуры.

BigQuery использует столбчатый формат хранения и алгоритм сжатия для хранения данных в Colossus наиболее оптимальным способом для чтения больших объемов структурированных данных. Colossus позволяет пользователям BigQuery легко масштабировать до десятков петабайт в хранилище (дорогие вычислительные ресурсы - типично для большинства традиционных баз данных и традиционных хранилищ данных).

# Borg: Вычисления

- Чтобы предоставить вам тысячи ядер ЦП, предназначенных для обработки вашей задачи, BigQuery использует преимущества [Borg](#), крупномасштабной системы управления кластерами Google. Кластеры Borg работают на [десятках тысяч машин](#) и сотнях тысяч ядер, поэтому ваш запрос, который использовал 3300 процессоров, использовал только часть емкости, зарезервированной для BigQuery, а емкость BigQuery составляет лишь часть емкости кластера Borg. Borg распределяет ресурсы сервера по заданиям; работа в данном случае - кластер Dremel.
- Машины выходят из строя, блоки питания выходят из строя, сетевые коммутаторы выходят из строя и множество других проблем может возникнуть при работе большого производственного центра обработки данных. Borg обходит его, а программный уровень абстрагируется. В масштабе Google тысячи серверов выходят из строя каждый божий день, и Borg защищает нас от этих сбоев. Кто-то отключает стойку в центре обработки данных во время выполнения вашего запроса, и вы никогда не заметите разницы.

# Jupiter: Сеть

Помимо очевидных потребностей в координации ресурсов и вычислительных ресурсов, рабочие нагрузки больших данных часто ограничиваются пропускной способностью сети. Сеть Google Jupiter может обеспечить общую пропускную способность 1 Петабит/сек, что позволяет нам эффективно и быстро распределять большие рабочие нагрузки.

Сетевая инфраструктура Jupiter может быть самым большим отличием Google Cloud Platform. Он обеспечивает достаточную пропускную способность, чтобы позволить 100 000 машин связываться с любой другой машиной со скоростью 10 Гбит/с. Пропускная способность сети, необходимая для выполнения нашего запроса, будет использовать менее 0,1% от общей емкости системы. Эта полнодуплексная полоса пропускания означает, что местоположение внутри кластера не имеет значения. Если каждая машина может общаться с любой другой машиной на скорости 10 Гбит/с, местоположение стойки не имеют значения.

Традиционные подходы к разделению хранилища и вычислений включают хранение данных в объектном хранилище объектов, например Google Cloud Storage или AWS S3, и загрузку этих данных на виртуальные машины по запросу. Этот подход часто более эффективен, чем архитектура HDFS, но зависит от ограничений пропускной способности локальных виртуальных машин и хранилища объектов. Jupiter позволяет нам полностью обойти это и считывать терабайты данных за секунды непосредственно из хранилища для каждого SQL-запроса.

# BigQuery: Служба

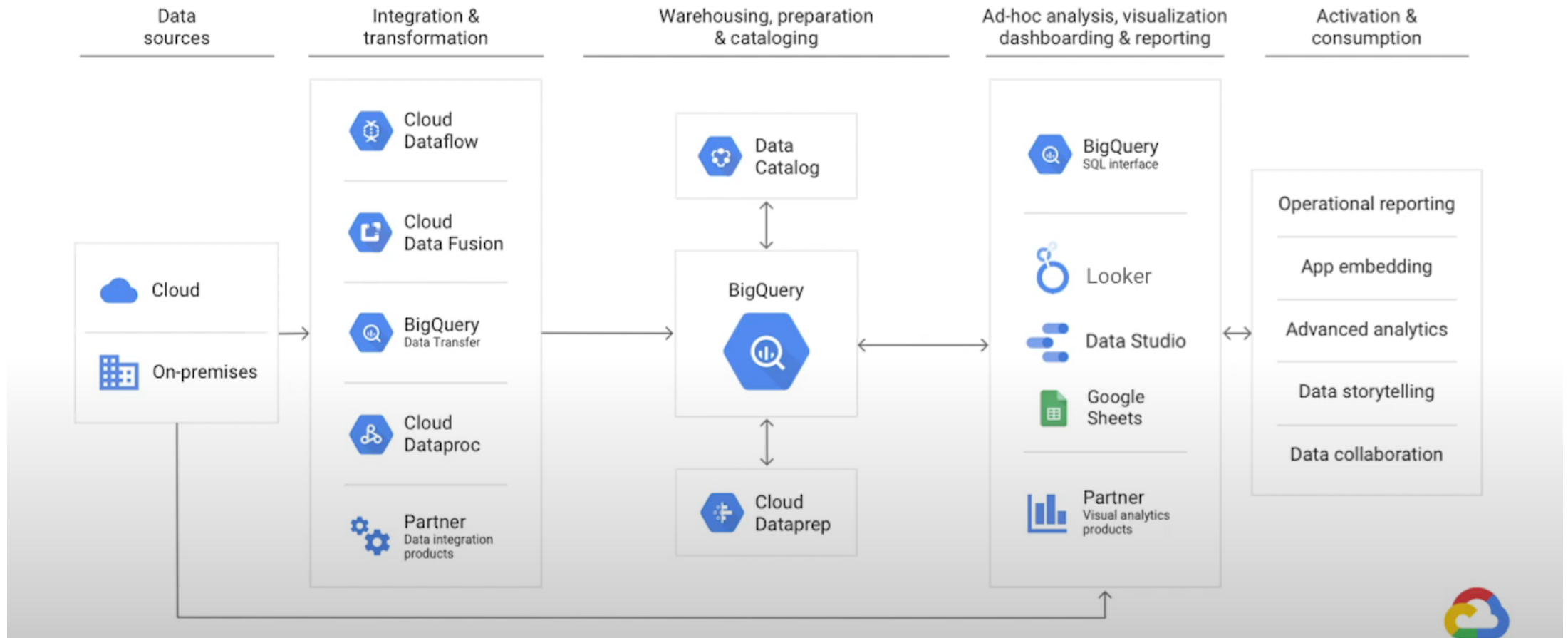
- В конце концов, эти компоненты инфраструктуры объединяются с несколькими десятками высокоуровневых технологий, API и сервисов, таких как Bigtable, Spanner (полностью управляемая реляционная база данных с неограниченным масштабом) и Stubby (шифрует DNS-запросы), чтобы сделать одну прозрачную и мощную облачную **аналитическую базу данных** - BigQuery.
- Конечная ценность BigQuery заключается не в том, что он дает вам невероятный масштаб вычислений, а в том, что вы можете использовать этот масштаб для своих повседневных SQL-запросов, даже не задумываясь о программном обеспечении, виртуальных машинах, сетях и дисках. BigQuery действительно является **бессерверной** базой данных, и простота BigQuery позволяет клиентам с десятками петабайт получить почти такой же опыт, как и клиенты бесплатного уровня.



# Итоги

- BigQuery - это хранилище данных размером в петабайты, предназначенное для легкого приема, хранения, анализа и визуализации данных. Как правило, вы собираете большие объемы данных из своих баз данных и других сторонних систем, чтобы ответить на конкретные вопросы. Вы можете загружать эти данные в BigQuery, загружая их в пакетном режиме или передавая данные напрямую, чтобы получать аналитические данные в реальном времени. BigQuery поддерживает стандартный диалект SQL, поэтому, если вы уже знаете SQL, все готово.
- Как только ваши данные будут помещены в BigQuery, вы можете приступить к выполнению запросов к ним. BigQuery - отличный выбор, когда ваши запросы требуют сканирования большой таблицы или просмотра всего набора данных. Сюда могут входить такие запросы, как суммы, средние, подсчеты, группировки или даже запросы для создания моделей машинного обучения. Типичные варианты использования BigQuery включают крупномасштабное хранение и анализ или онлайн-аналитическую обработку (OLAP).

# Flexibility and choice across the BI process



Обеспечение высокой скорости реакции и степень параллелизма для интерактивных панелей мониторинга и отчетов, извлекающих данные в реальном времени из BigQuery