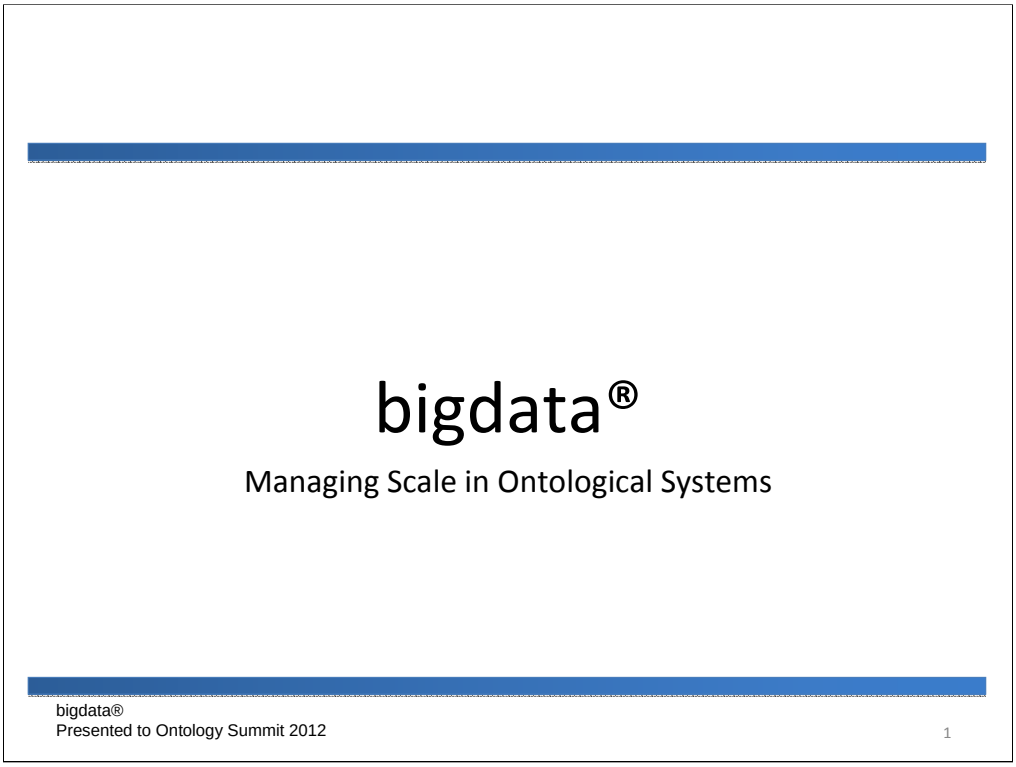


This presentation offers a brief look scale in ontological (semantic) systems, tradeoffs in expressivity and data scale, and both information and systems architectural concerns for very large data scales.

SYSTAP Company Overview

Overview

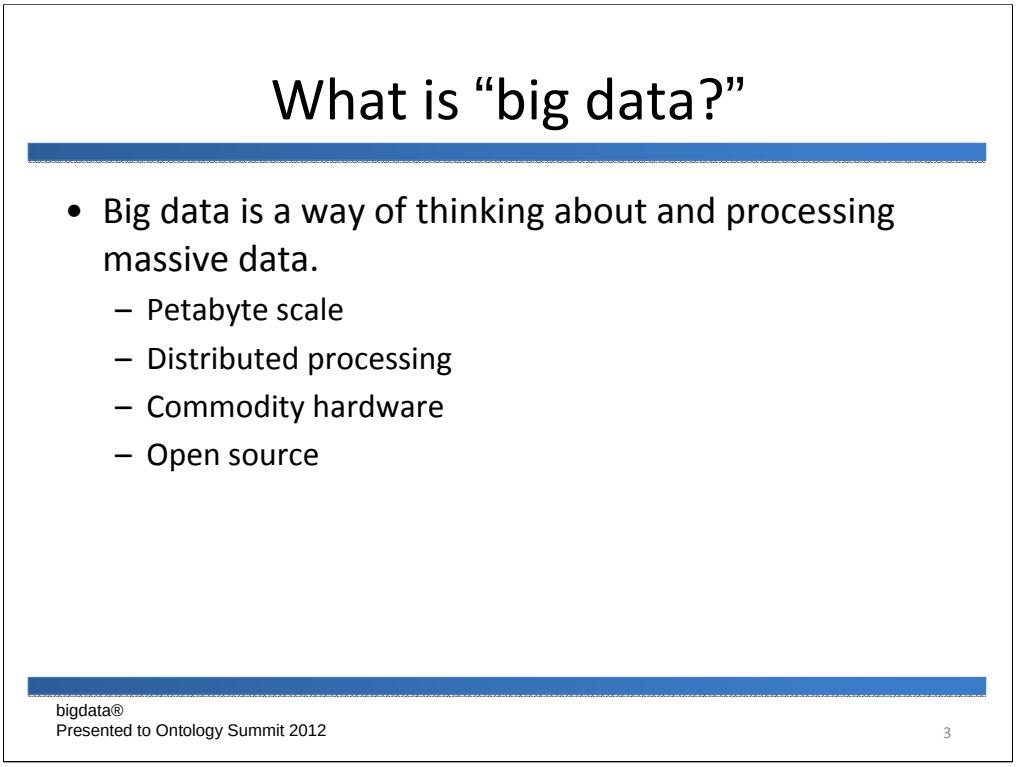
- LLC, Small Business, Founded 2006
- 100% Employee Owned, 2 Principals
- 35 Years Combined Experience, 16 Years With Semantic Web Technologies

Customers & Use Cases

- **Intelligence Community**
  - Federation and semantic alignment at scale to facilitate rapid threat detection and analysis
- **Telecommunications**
  - Horizontal data integration across enterprise services
- **Health Care**
  - Data integration and analytics
- **Network Storage**
  - Embedded device monitoring and root cause analysis
- **Collaboration and Knowledge Portals**
  - Bioinformatics, manufacturing, NGOs, etc.
- **OEM Resellers**

Corporate Services & Product Offering

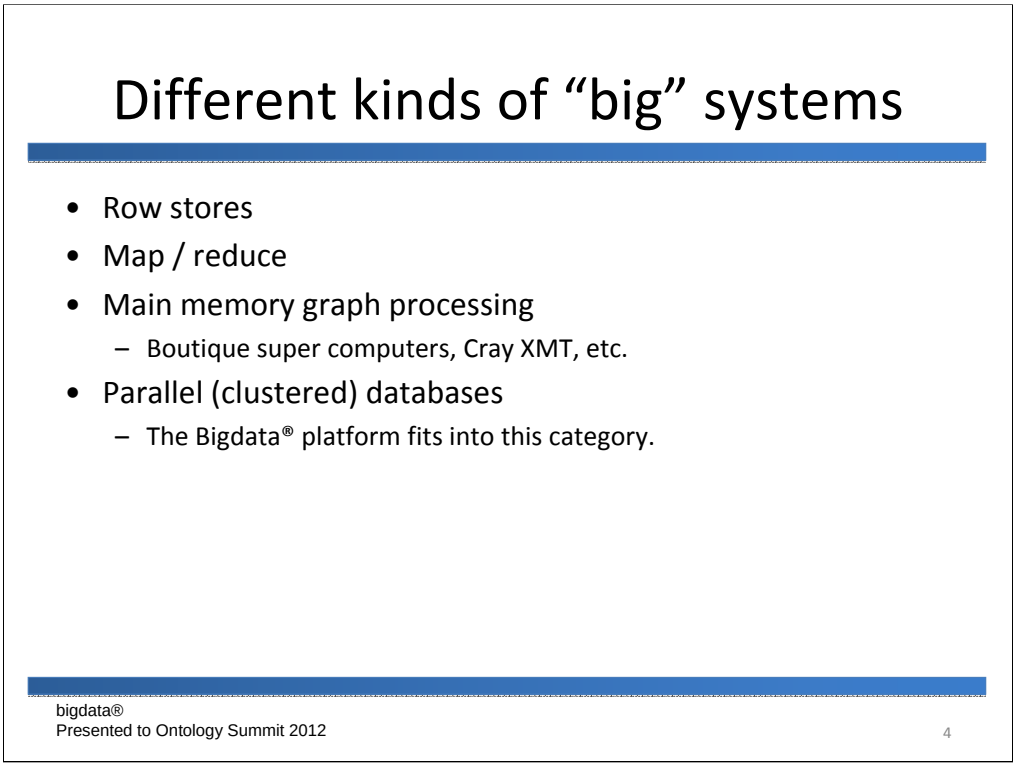
- **Semantic Web Consulting Services**
  - System vision, design, and architecture
  - Information architecture development
  - Ontology development and inference planning
  - Relational data mapping and migration
  - Rapid prototyping
- **Bigdata®, an open-source, horizontally-scaled high-performance RDF database**
  - Dual licensing (GPL, commercial)
  - Infrastructure planning
  - Technology identification and assessment
  - Benchmarking and performance tuning
  - Feature development
  - Training & Support



10 years ago everyone knew that the database was a frozen market. Today, nothing could be further from the truth This is a time of profound upheaval in the database world.

There are a number of trends which are driving this. One is the continued pressure on commodity hardware prices and performance. Another is the growth in open source software. Today, all “big data” systems leverage open source software and many federal contracts require it. At the same time the speed of processors has hit the wall so applications are forced to parallelize and use distributed architectures. SSD has also changed the playing field, virtually eliminating disk latency in critical systems.

The central focus for the bigdata® platform central focus is a parallel semantic web database architecture. There are other distributed platforms, including those based on custom hardware solutions, for main memory graph processing. These tend to lack core features of a database architecture such as durability and isolation.



There are many different approaches to massive data. I have listed a few here.

Row stores

Row stores provide single key access to schema flexible data. The original inspiration was Google’s “bigtable” system.

Map reduce

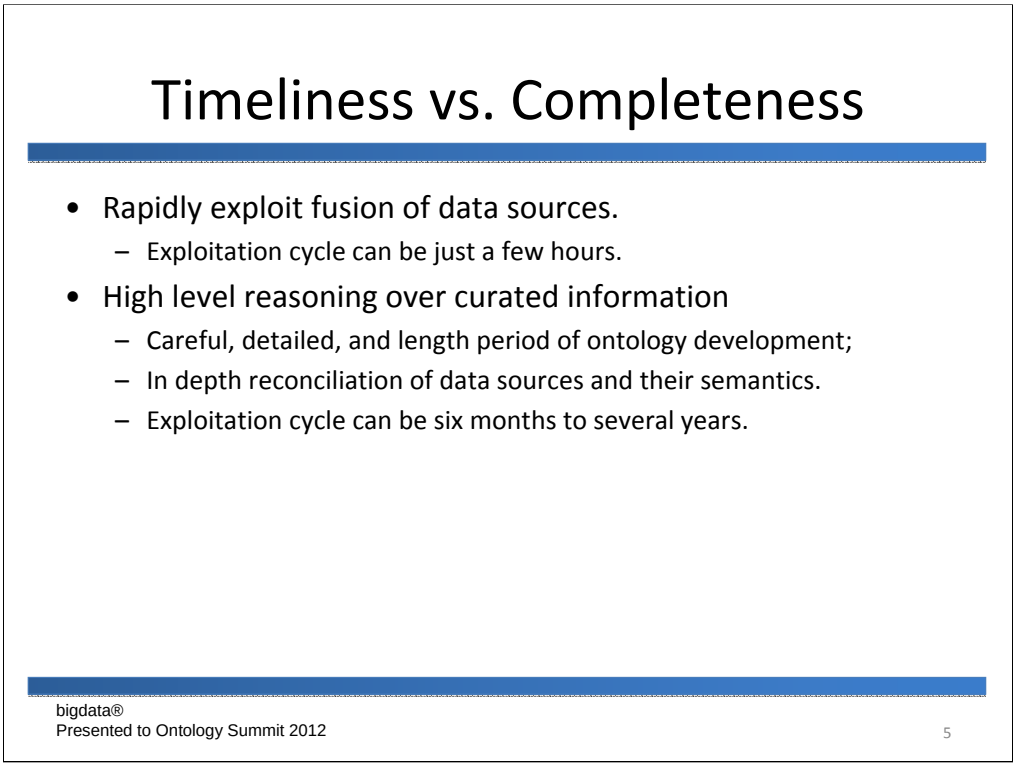
Map reduce decompose a problem, performs a local computation, and then aggregates the outputs of the local computation. Map reduce is great for problems with good input locality. The general map/aggregation pattern is an old one. The modern map/reduce systems are inspired by Google’s “map/reduce” system.

Main memory graph processing

These systems provide in-memory graph processing, but lack many of the features of a database architecture (isolation, durability, etc). The intelligence community has a long history in this area. More recently “RAM clouds” are emerging in the private sector. There are also custom hardware solutions in the space, such as the Cray XMT.

Parallel databases

Parallel database platforms offer ACID (Atomic, Consistent, Isolated, Durable) data processing. This world can be divided up again into traditional relational databases, column stores, and increasingly, semantic web database platforms.



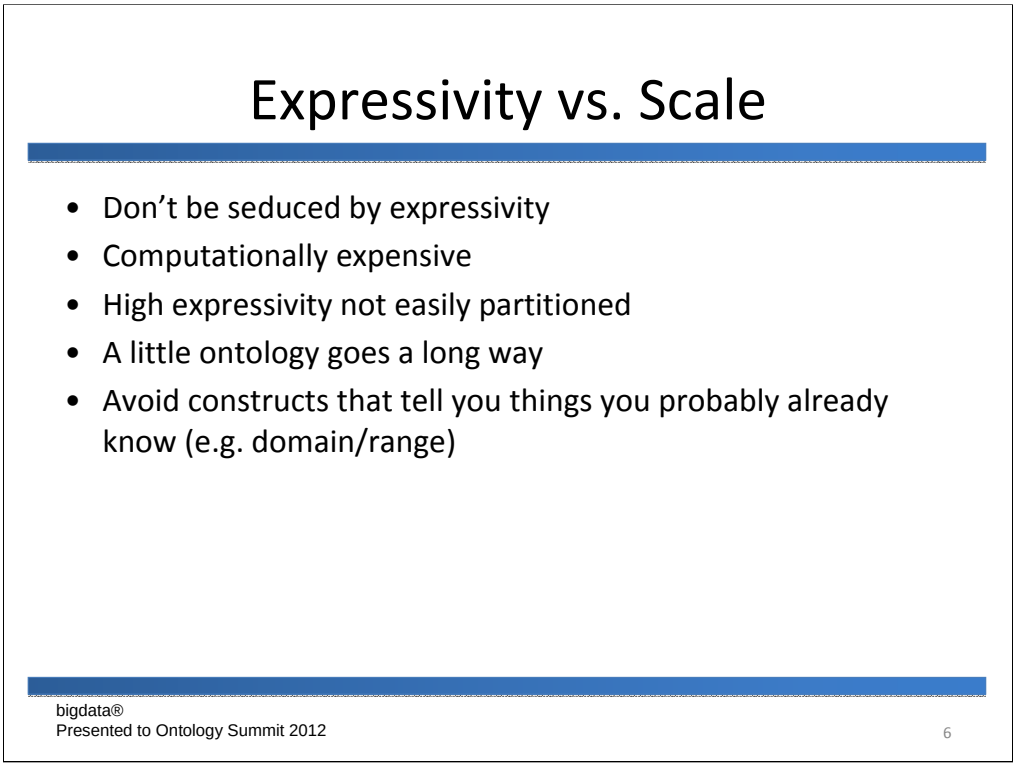
There are two very different ways in which semantic technology can unlock value from data. It is important to recognize which world fits your problem and how to best invest your resources.

Rapid information exploitation

In this world, the benefit is derived from the rapid pace at which new data and new data sources can be combined and exploited.

High level reasoning over curated information

In this world, the benefit is derived from non-trivial inferences drawn over highly vetted data.



Many times people try to have both expressivity and scale. This is very expensive

Don't be seduced by expressivity

Just because you CAN say it doesn't mean you SHOULD say it. Stick to things that are strictly useful to building your big data application.

Computationally expensive

Expressivity is not free. It must be paid for either with load throughput or query latency, or both.

Not easily partitioned

Higher expressivity often involves more than one piece of information from the abox – meaning you have to cross server boundaries. With lower expressivity you can replicate the ontology everywhere on the cluster and answer questions LOCALLY.

A little ontology goes a long way

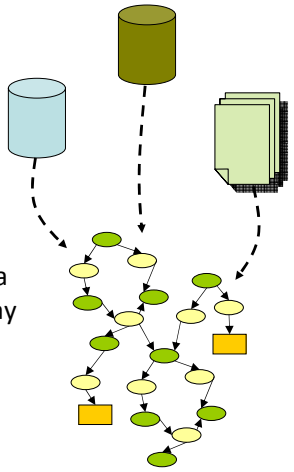
There can be a lot of value just getting the data federated and semantically aligned.

Avoid constructs that tell you things you probably already know (e.g. domain/range)

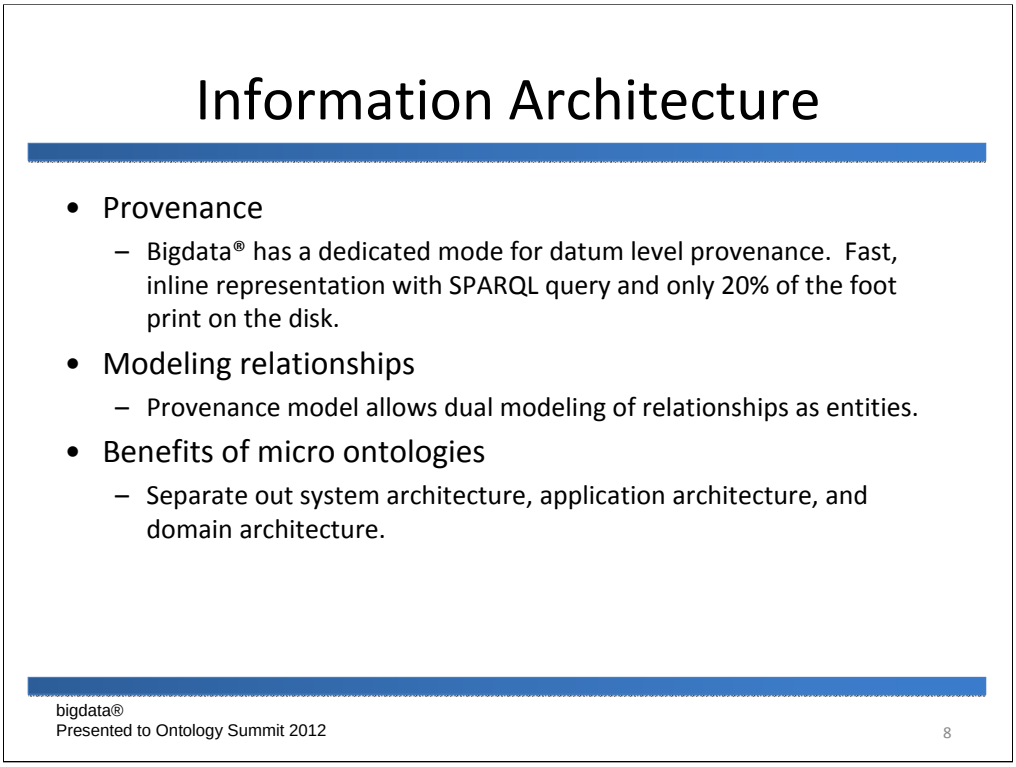
Domain and range are interesting. People think they are supposed to be for describing what properties and reverse properties a particular type of resource should have, but in reality they are used to infer a resource's type based on its properties and reverse properties. This type of inference is almost never useful in real systems (classification of resources are almost always known in advance).

The killer “big data” app

- Clouds + “Open” Data = Big Data Integration
- Critical advantages
  - *Fast* integration cycle
  - Open standards
  - Integrate heterogeneous data, linked data, structured data, and data at rest.
  - Opportunistic exploitation of data, including data which can not be integrated quickly enough today to derive its business value.
  - Maintain fine-grained provenance of federated data.



This is our take on where this is all heading. We tend to focus on high data scale and low expressivity with rapid data exploitation cycles.

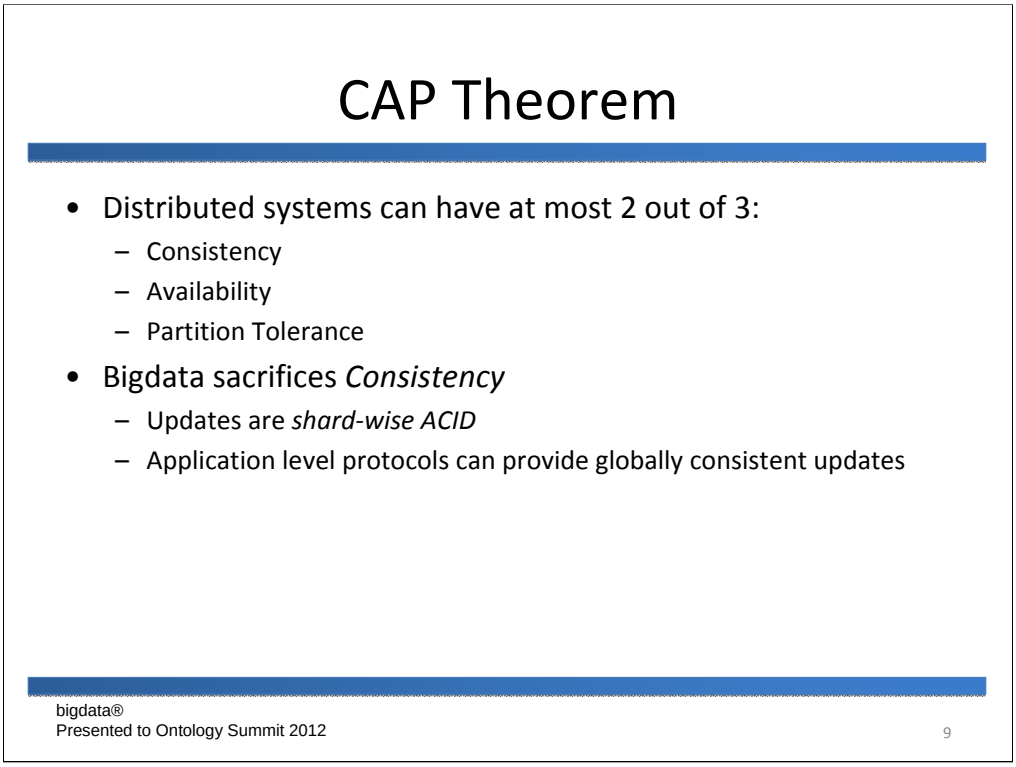


Provenance is vital in many applications, but support for provenance is non-standard and varies across platforms.

Our approach is a custom provenance mode in which each statement can be used *as-if* it were a resource. You can create multiple levels of statements about statements, interchange the data using an RDF/XML extension, and query it using SPARQL. We also provide truth maintenance for the statements about statements.

Dedicated handling for provenance is a necessity, not just a “shortcut” for reification. The cost of reifying every assertion is enormous at scale. There is also a high cognitive cost for always reifying the data. Only reifying some data makes matters worse since (a) you could find out that you need to reify something else later; and (b) you need to keep track of what is reified and what is not reified.

In the “ontology light, data heavy” world we tend to recommend the use of micro ontologies. This helps to partition the different aspects of the application while making only the domain distinctions which are critical for a given application. The more ontological distinctions you create, the more distinctions you have to maintain and the harder it is to ensure that the semantics of the ontological distinctions are followed correctly throughput the data and the application.

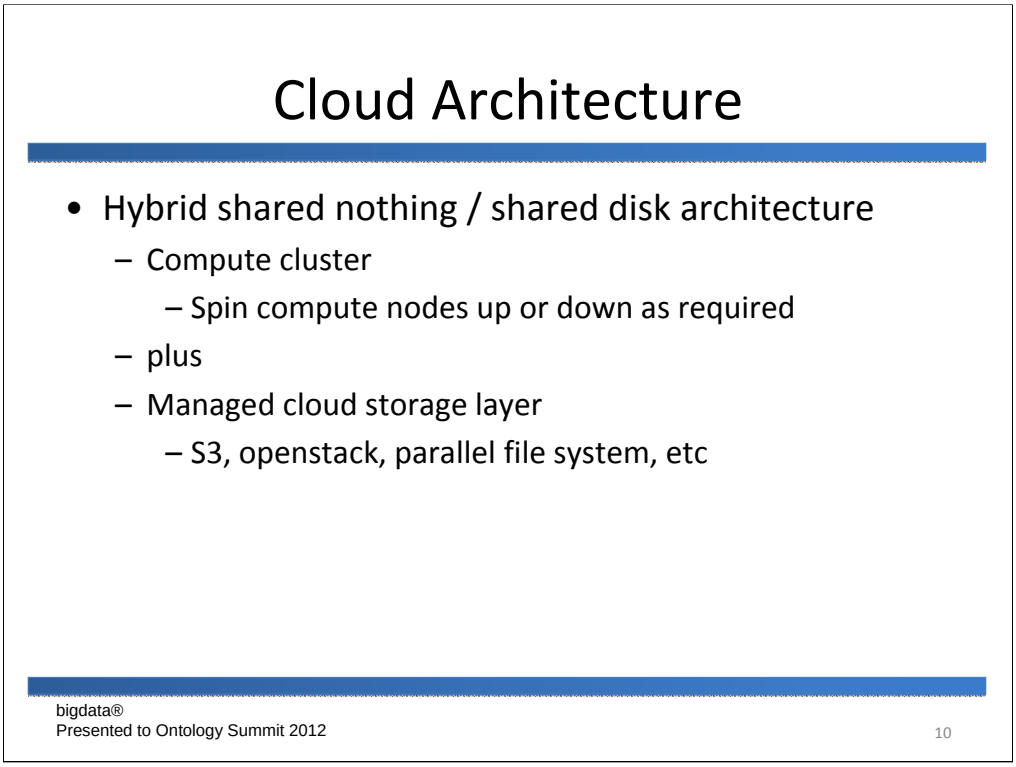


All distributed data platforms make compromises. They have to. The question is which compromises are the right ones for your application.

Bigdata gives up *consistency* on *writes* (reads have snapshot isolation). Updates are shard-wise ACID. Globally consistent updates can be created through application level protocols.

Main memory graph databases typically sacrifice either *isolation* or *consistency.* Some custom hardware platforms sacrifice *availability*, and *partition tolerance* (either it is up or it is down – no partial failures).

Each of these compromises tends to have a corresponding benefit. It all depends what you need for you application.



Large scale systems need to think about deployment architectures and managing the raw resources that they will consume. Bigdata® has evolved over time into a hybrid architecture in which there is a separation of concerns between the compute and persistence layers. The architecture still leverages local computation (including local disk access) whenever possible. We do this now through caching shards on the instances nodes while leaving read-only files on long term durable storage. This architecture makes it possible to deploy to platforms like Amazon EC2 (compute) + Amazon S3 (high 9s durable storage).

bigdata ®

Flexible
Reliable
Affordable
Web-scale computing.