Following are the advantages of using ArangoDB −

Consolidation

As a native multi-model database, ArangoDB eliminates the need to deploy multiple databases, and thus decreases the number of components and their maintenance. Consequently, it reduces the technology-stack complexity for the application. In addition to consolidating your overall technical needs, this simplification leads to lower total cost of ownership and increasing flexibility.

Simplified Performance Scaling

With applications growing over time, ArangoDB can tackle growing performance and storage needs, by independently scaling with different data models. As ArangoDB can scale both vertically and horizontally, so in case when your performance demands a decrease (a deliberate, desired slow-down), your back-end system can be easily scaled down to save on hardware as well as operational costs.

Reduced Operational Complexity

The decree of Polyglot Persistence is to employ the best tools for every job you undertake. Certain tasks need a document database, while others may need a graph database. As a result of working with single-model databases, it can lead to multiple operational challenges. Integrating single-model databases is a difficult job in itself. But the biggest challenge is building a large cohesive structure with data consistency and fault tolerance between separate, unrelated database systems. It may prove nearly impossible.

Polyglot Persistence can be handled with a native multi-model database, as it allows to have polyglot data easily, but at the same time with data consistency on a fault tolerant system. With ArangoDB, we can use the correct data model for the complex job.

Strong Data Consistency

If one uses multiple single-model databases, data consistency can become an issue. These databases aren’t designed to communicate with each other, therefore some form of transaction functionality needs to be implemented to keep your data consistent between different models.

Supporting ACID transactions, ArangoDB manages your different data models with a single back-end, providing strong consistency on a single instance, and atomic operations when operating in cluster mode.

Fault Tolerance

It is a challenge to build fault tolerant systems with many unrelated components. This challenge becomes more complex when working with clusters. Expertise is required to deploy and maintain such systems, using different technologies and/or technology stacks. Moreover, integrating multiple subsystems, designed to run independently, inflict large engineering and operational costs.

As a consolidated technology stack, multi-model database presents an elegant solution. Designed to enable modern, modular architectures with different data models, ArangoDB works for cluster usage as well.

Lower Total Cost of Ownership

Each database technology requires ongoing maintenance, bug fixing patches, and other code changes which are provided by the vendor. Embracing a multi-model database significantly reduces the related maintenance costs simply by eliminating the number of database technologies in designing an application.

Transactions

Providing transactional guarantees throughout multiple machines is a real challenge, and few NoSQL databases give these guarantees. Being native multi-model, ArangoDB imposes transactions to guarantee data consistency.