NOSQL:
CAP Theorem

CAP Theorem

- Continue at 8:43 PM

CAP Theorem

Distributed system with Shared Data: Vasanti Bhat-Nayak and Grace Hopper need a package from R to do a naïve Bayes classification. If there were only one server that contained this package, then consistency would be easy. But, availability would be restricted. When multiple R users want to download a package, the server gets clogged. Therefore, the cran packages are replicated on multiple servers around the world. When a package needs to be updated, then the master node asks all servers to update simultaneously. So when Vasanti and Grace download a package from different servers they will get the same version of the Naive Bayes package.

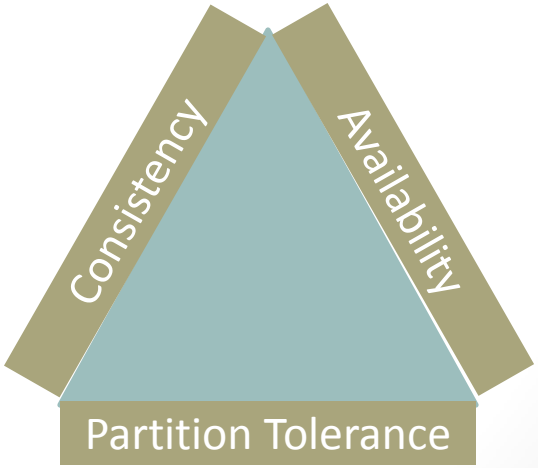
CAP Theorem

Distributed system with Shared Data: Vasanti Bhat-Nayak and Grace Hopper need a package from R to do a naïve Bayes classification. If there were only one server that contained this package, then consistency would be easy. But, availability would be restricted. When multiple R users want to download a package, the server gets clogged. Therefore, the cran packages are replicated on multiple servers around the world. When a package needs to be updated, then the master node asks all servers to update simultaneously. So when Vasanti and Grace download a package from different servers they will get the same version of the Naive Bayes package.

Partition of the Distributed System: But, what happens if on that day the Andorran server that Vasanti uses, can't be updated because of a communication error. The database has two choices: (1) It can wait until the Andorran server is fixed and then do the update. (2) Or, it updates all the other servers that allow the update. In the first case we forgo availability and nobody has access to the most recent Naive Bayes package. In the second case Vasanti and Grace will have different results because the packages are different.

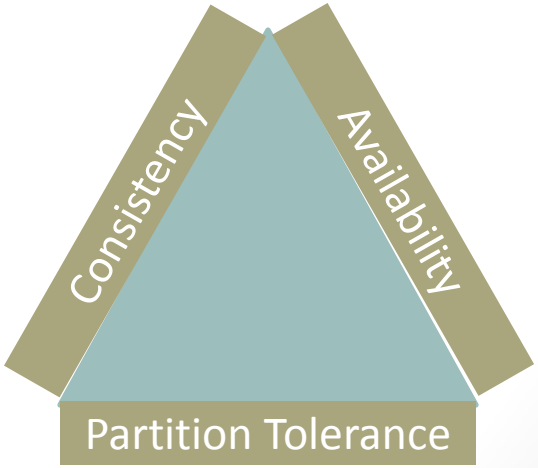
CAP Theorem

- CAP stands for:
  - **C**onsistency
  - **A**vailability
  - **P**artition Tolerance



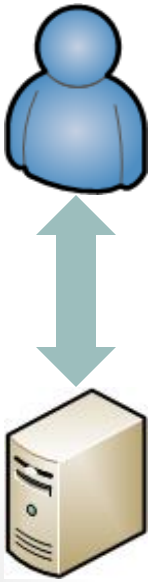
CAP Theorem

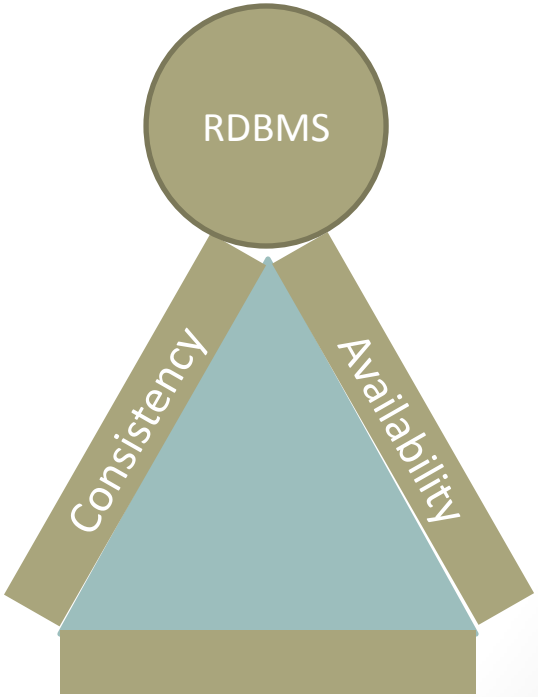
- CAP stands for:
  - **C**onsistency: All nodes see the same data at the same time
  - **A**vailability: Nodes are available for updates and reads
  - **P**artition Tolerance: Arbitrary message loss or partial failure does not bring down the system



CAP Theorem

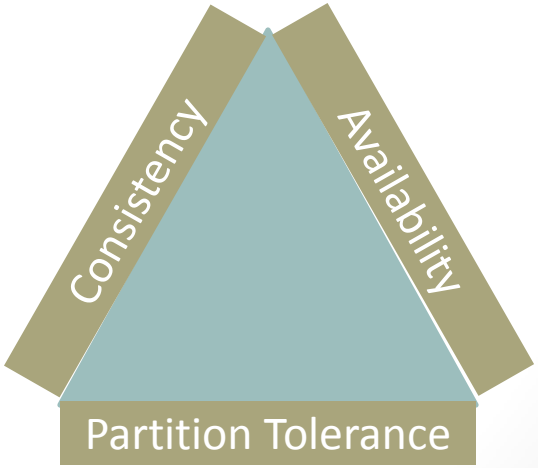
- Assume a single node with one set of data.
- This simple system resembles a typical RDBMS.
- Partition tolerance is irrelevant, because we only have one node.





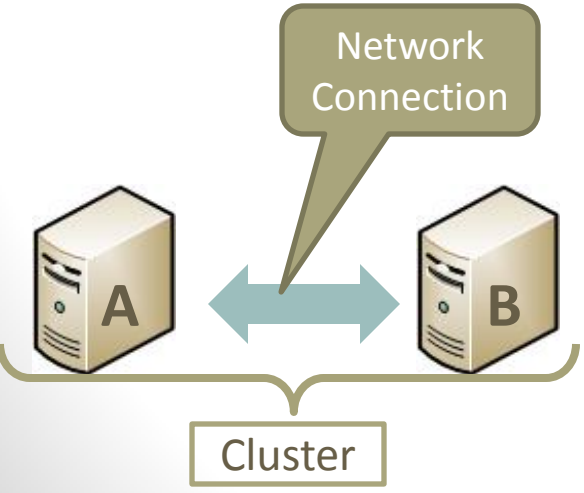
CAP Theorem

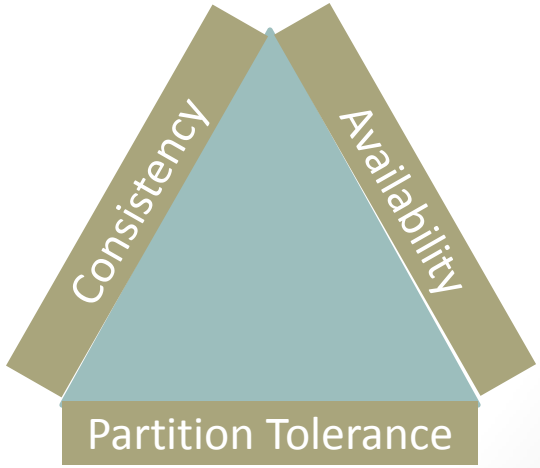
- The CAP theorem was formulated by Eric Brewer http://en.wikipedia/wiki/CAP_theorem
- Two formulations of the CAP theorem:
  - You can have at most two of the CAP properties for any shared data system.
  - During a network partition, a distributed system must choose either Consistency or Availability.



CAP Theorem

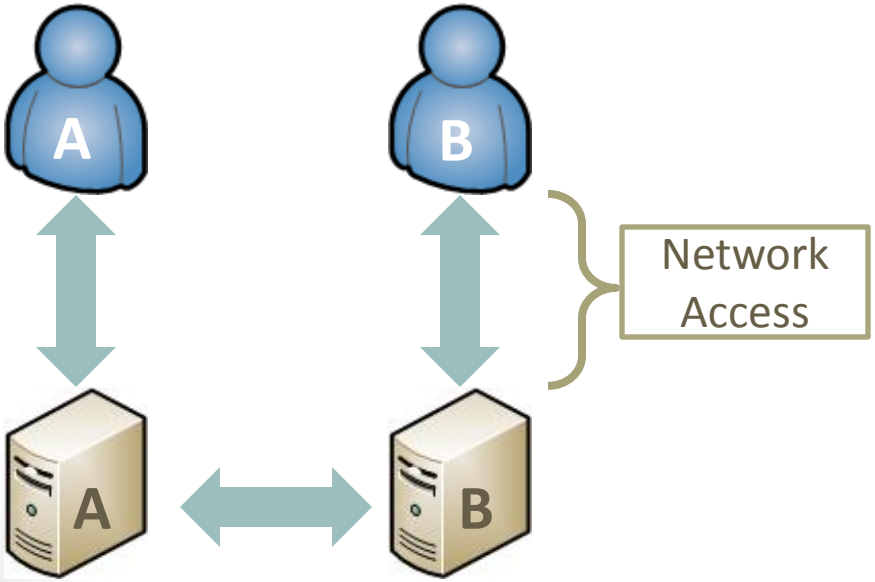
- Assume a cluster with shared and replicated data.
- The cluster consists of two connected nodes called A and B.

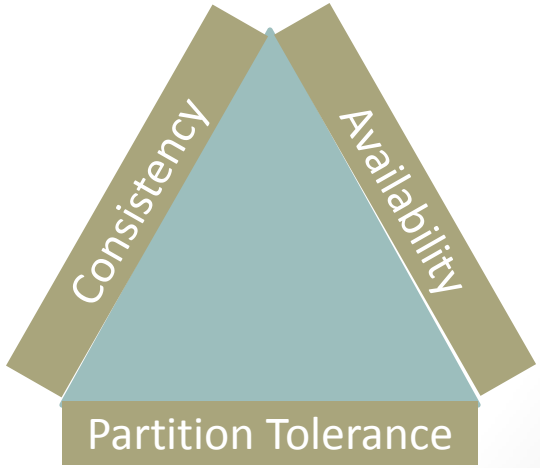




CAP Theorem

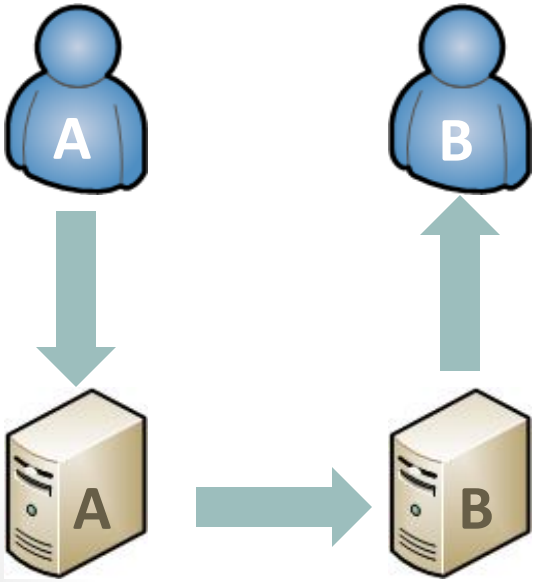
- Assume a cluster with shared and replicated data.
- The cluster consists of two connected nodes called A and B.
- The cluster is used by two users, called A and B. Each user has network access to a separate node

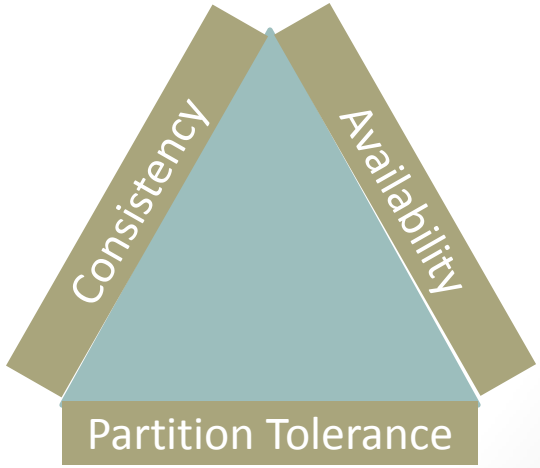




CAP Theorem

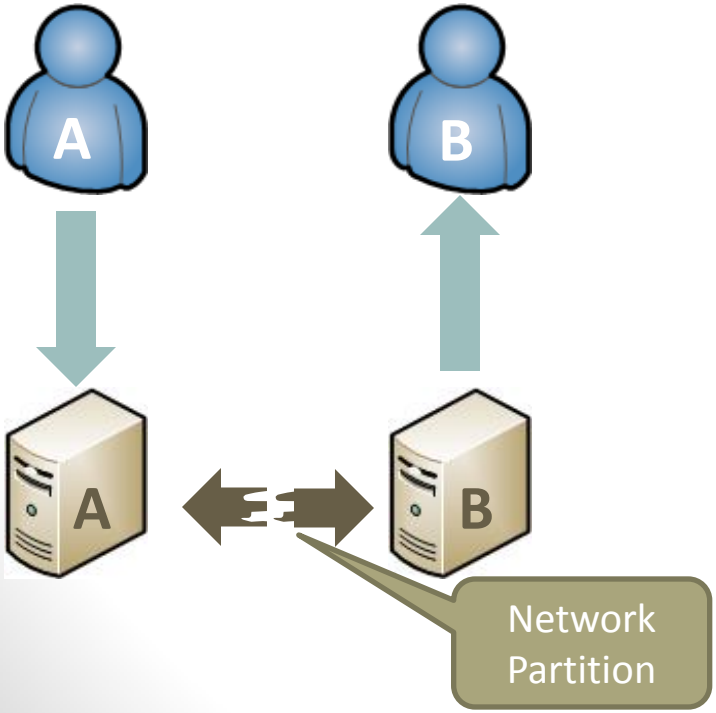
- Scenario 1: Network is available and Data are Consistent
  1. User A updates node A
  2. Update is communicated to node B
  3. User B reads the update from node B

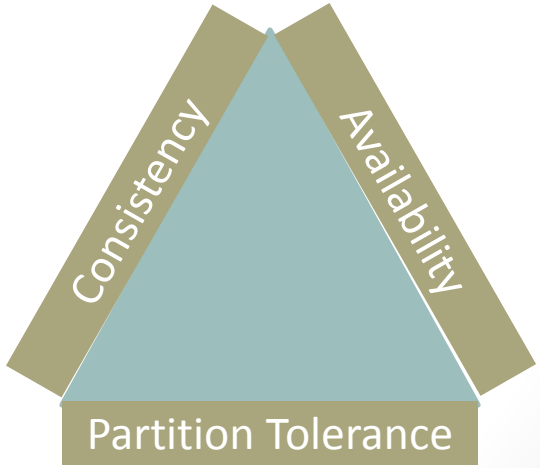




CAP Theorem

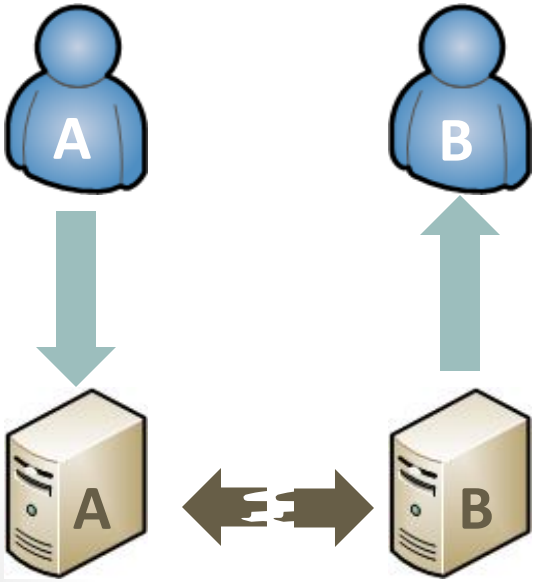
- Scenario 2: A network failure occurred.
  1. User A attempts to update node A
  2. Any Update cannot be communicated
  3. User B attempts to read the update

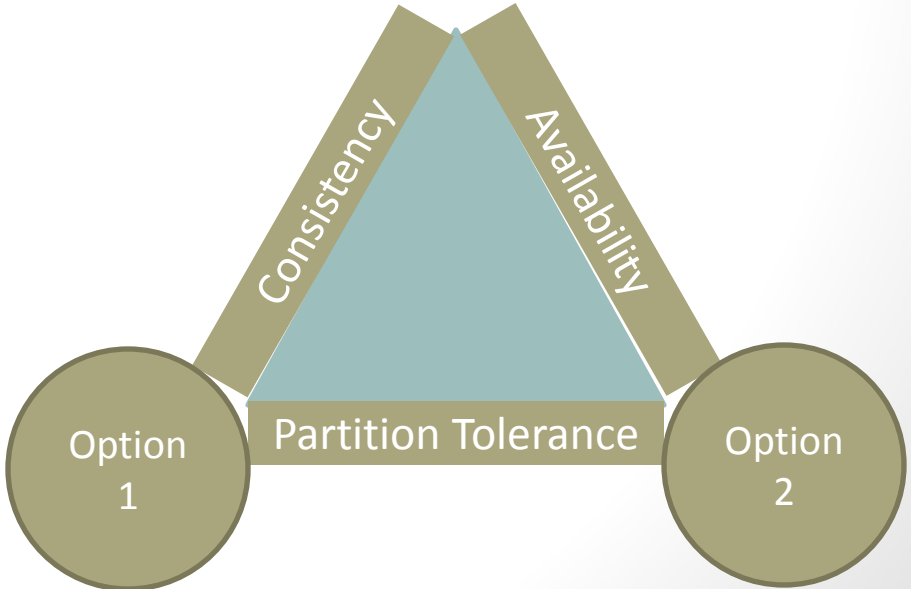




CAP Theorem

- Scenario 2: A network failure occurred. Two options:
  1. Make the database unavailable to avoid inconsistency
  2. Keep the database available and tolerate inconsistency





- Gilbert and Lynch; Perspectives on the CAP Theorem

NOSQL:
CAP Theorem