

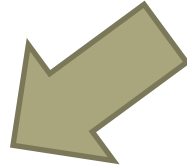
NOSQL: Scale-out

Scale-up vs. Scale-out

Before we discuss the nature of NOSQL, we should discuss the reasons for NOSQL.

Scale Up vs. Scale Out

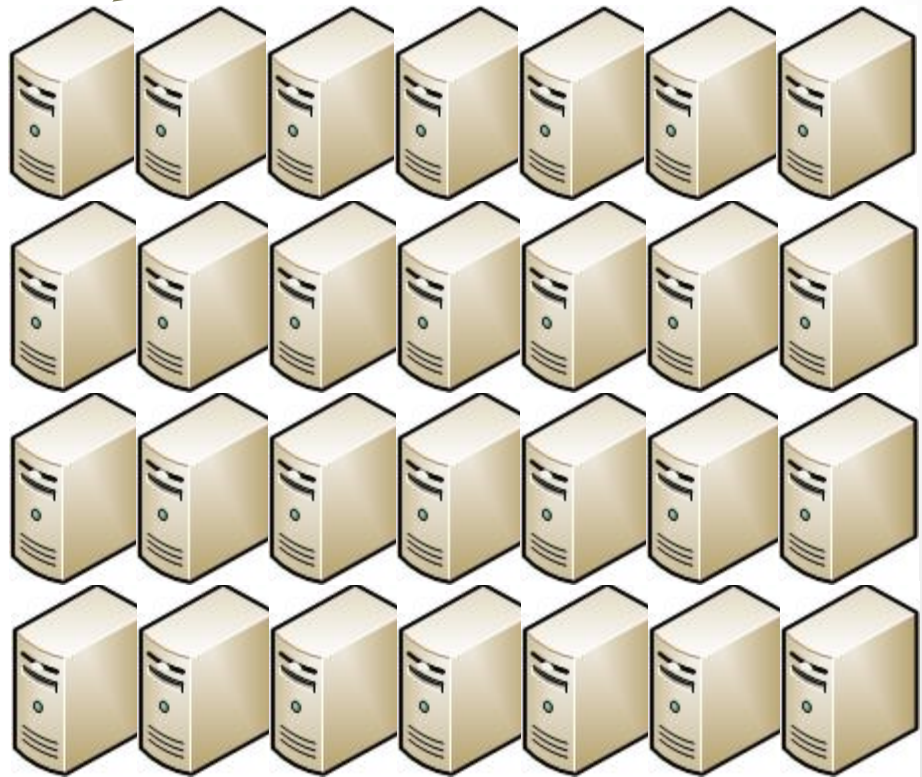
Scale Up



Scale Up vs. Scale Out

Scale Up

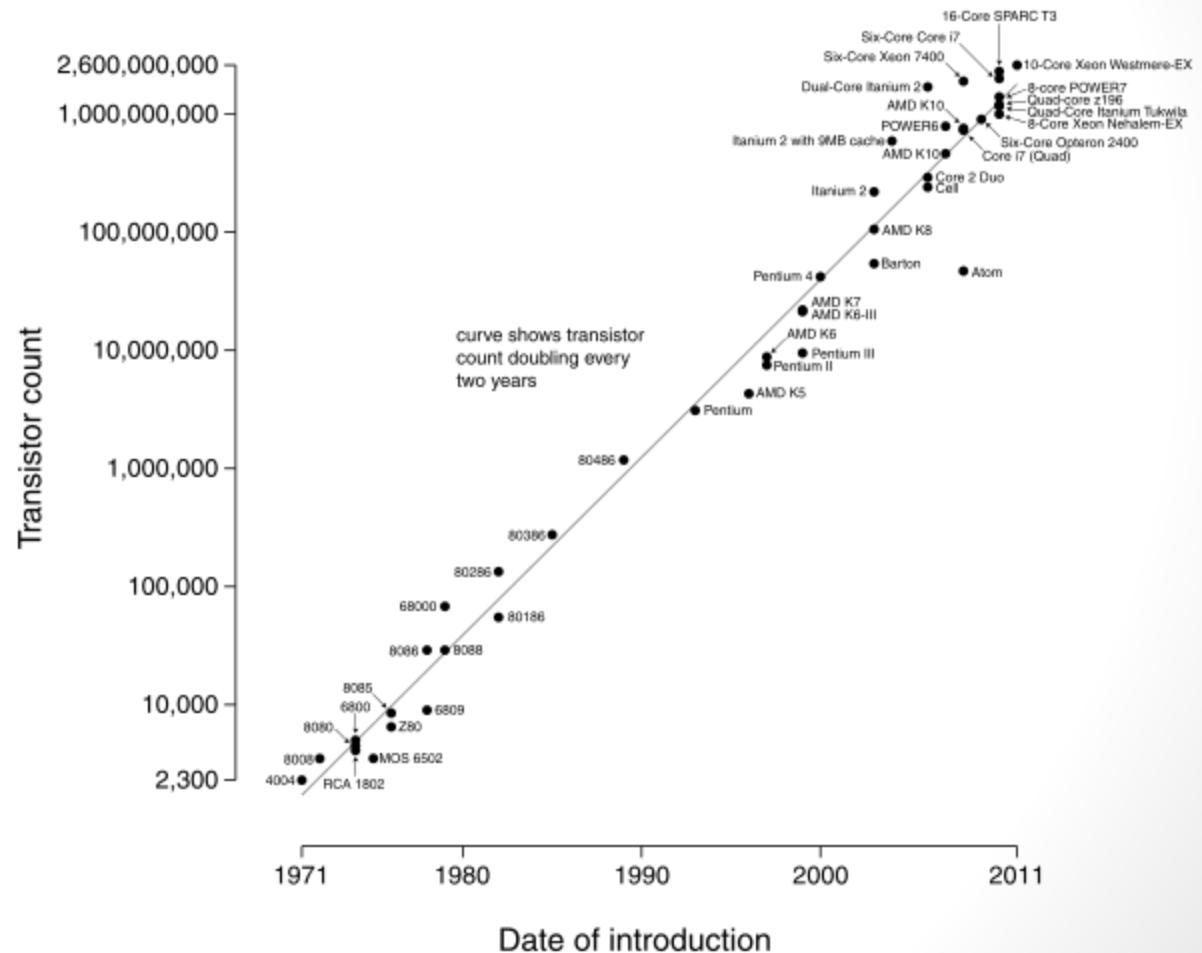
Scale Out



Scale-up vs. Scale-out

- Scale-up
- Moore's Law

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Scale-up vs. Scale-out



Grace Hopper

Scale-up vs. Scale-out



Grace Hopper

"In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for more systems of computers."

Cloud: Scale-out

- The primary characteristic of NOSQL is scale out.
- From a practical level, scale out requires an adjustable number of commodity computers.
- Cluster Elasticity:
[http://en.wikipedia.org/wiki/Elasticity %28data store%29](http://en.wikipedia.org/wiki/Elasticity_%28data_store%29)
- Virtual Machine
 - One computer “mimics” another computer. (A system platform supports execution of an operating system)
 - Allows hardware standardization.
 - Allows one server to “host” many computers.
 - Virtual machines in the cloud can be set up and taken down (dehydrated, reduced to an image).
- Cloud: What is the “cloud”? Remote access to a single point provides many online services like servers and storage.
([http://en.wikipedia.org/wiki/Cloud computing](http://en.wikipedia.org/wiki/Cloud_computing)).

Cloud: Services

- Amazon Web Services
- GoGrid
- Google Compute Engine
- Microsoft Azure
- Rackspace
- SoftLayer



Scale-out and the “Cloud”

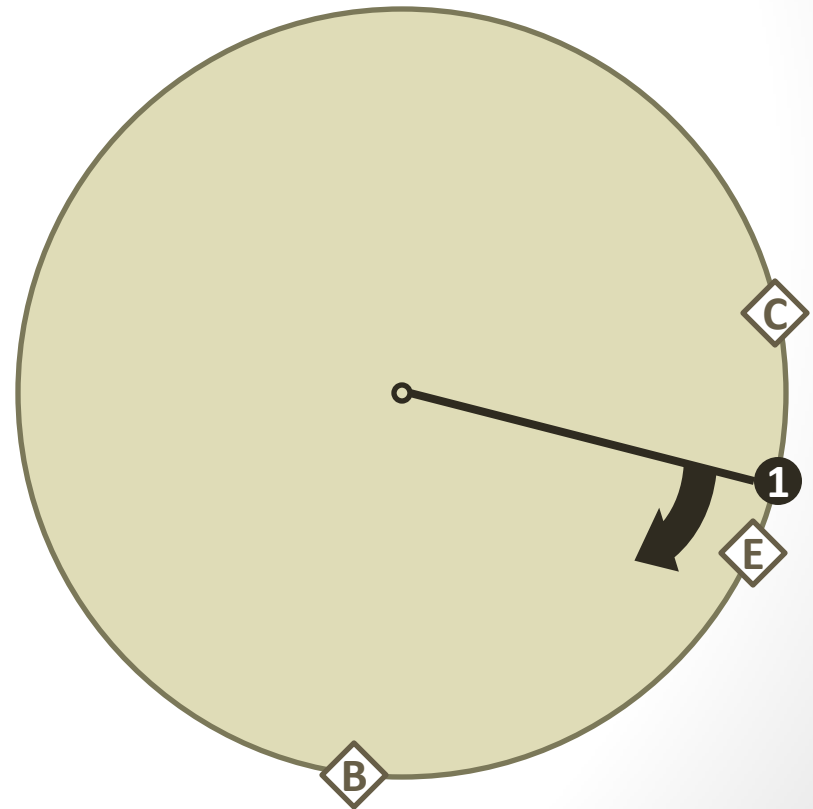
- **Elasticity** has made cloud computing feasible
- Clouds generally employ **virtual machines** that can be created at a moments notice, reduced to an image (dehydrated), re-started from an image, and deleted (recycled).
- How do we partition storage or usage among an unknown number of machines? Often we do not know ahead of time if new machines will become available or which machines will be recycled.
- Storage and usage are mapped to machines by a hash table. In traditional hash tables a change in the number of slots requires most keys to be remapped.
- We need a strategy to minimize remapping of storage and usage among the available computers: Consistent Hashing:
[http://en.wikipedia.org/wiki/Consistent hashing](http://en.wikipedia.org/wiki/Consistent_hashing)

Consistent Hashing

- Consider a hash map where each object is mapped to a point on the circumference of a circle. For instance an object is mapped to the number of minutes on a clock.
- Computers, Files, Processes, etc., are mapped in this manner on the same circle.
- A computer “claims” all files and processes who have a hash that is clock wise to that computer.

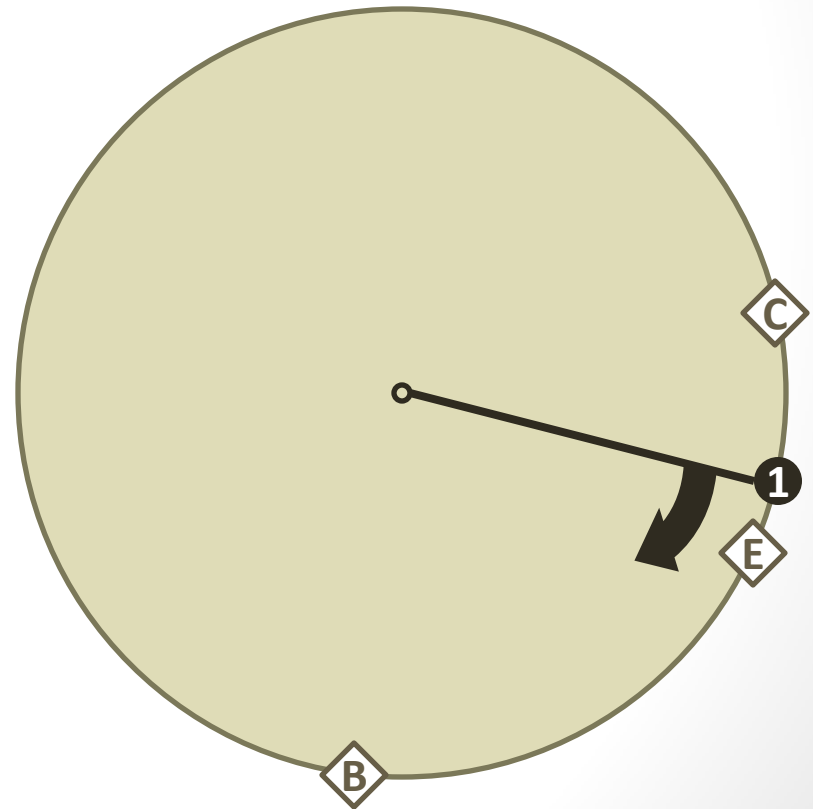
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	1
E	Data Object	18	1
1	Machine 1	17	E C B



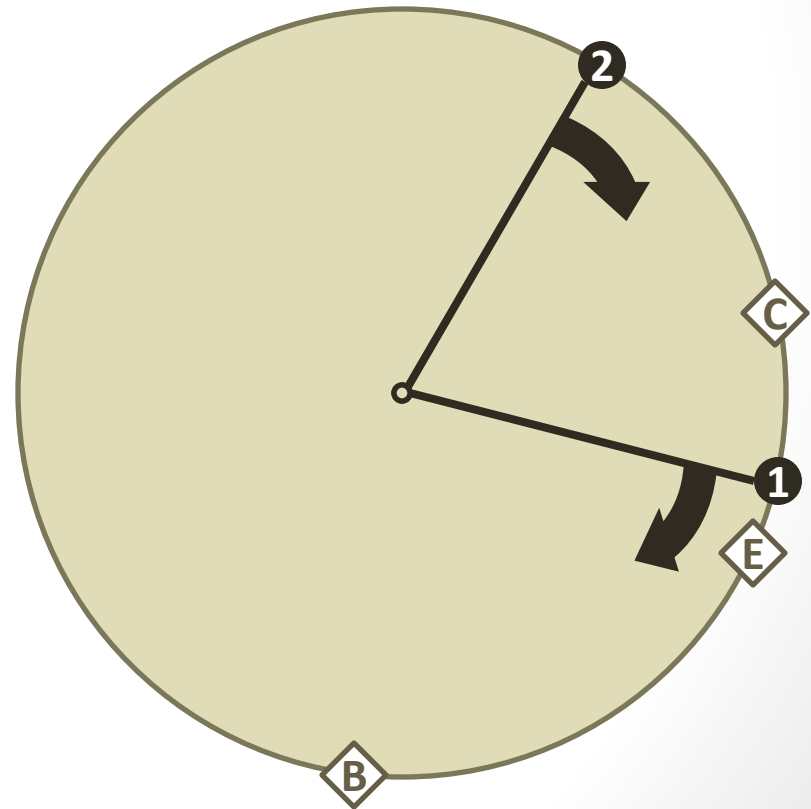
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	1
E	Data Object	18	1
1	Machine 1	17	E C B



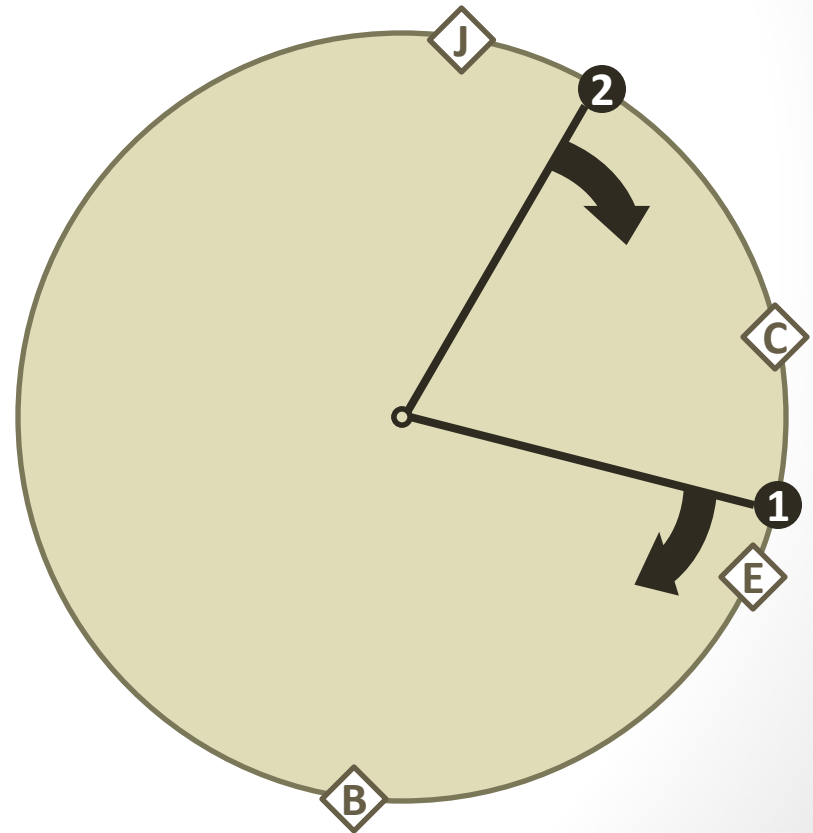
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	2
E	Data Object	18	1
1	Machine 1	17	E B
2	Machine 2	5	C



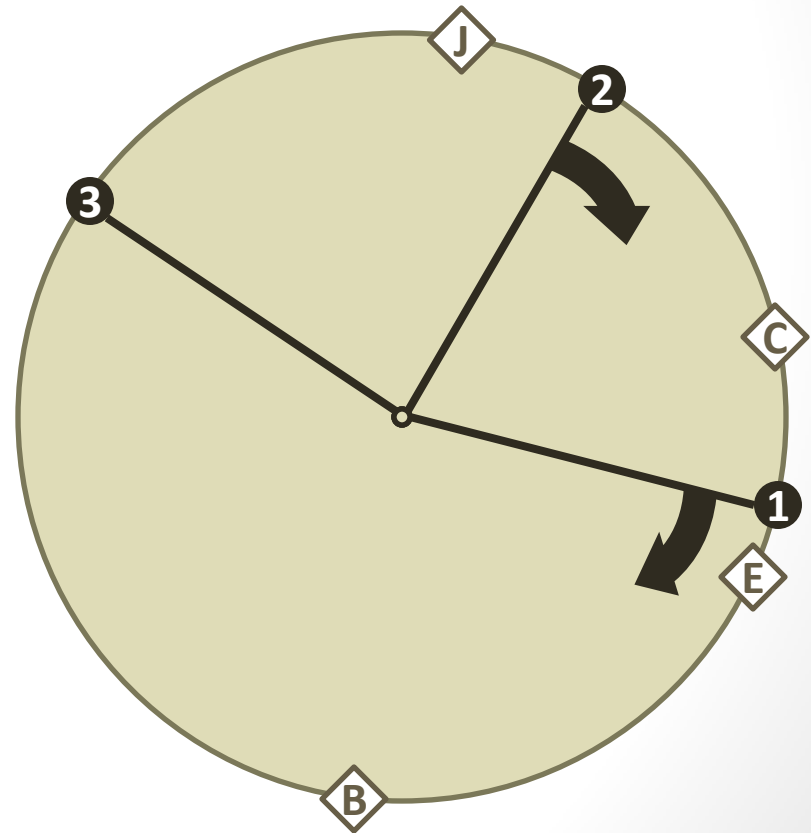
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	2
E	Data Object	18	1
J	Data Object	2	1
1	Machine 1	17	E J B
2	Machine 2	5	C



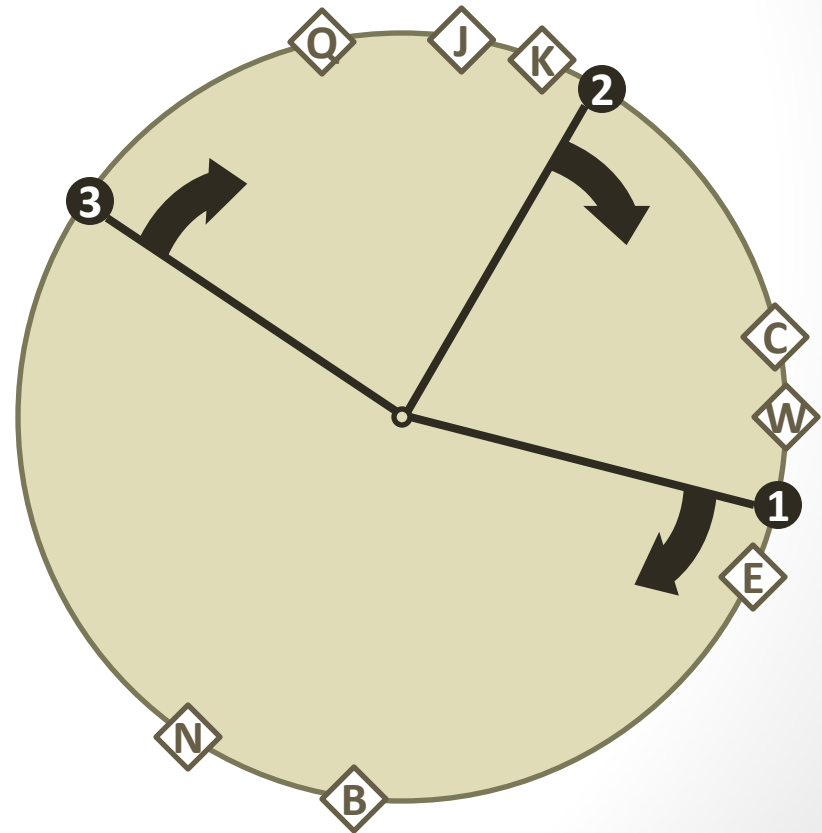
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	2
E	Data Object	18	1
J	Data Object	2	3
1	Machine 1	17	E B
2	Machine 2	5	C
3	Machine 3	51	J



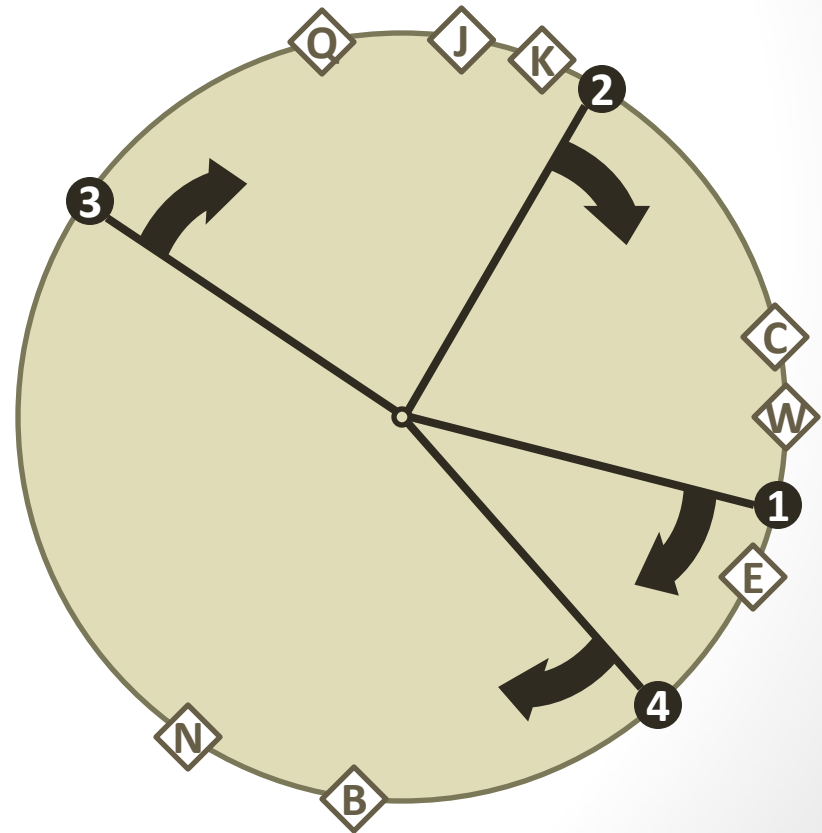
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	2
E	Data Object	18	1
J	Data Object	2	3
K	Data Object	4	3
N	Data Object	35	1
Q	Data Object	57	3
W	Data Object	15	2
1	Machine 1	17	E B N
2	Machine 2	5	C W
3	Machine 3	51	J K Q



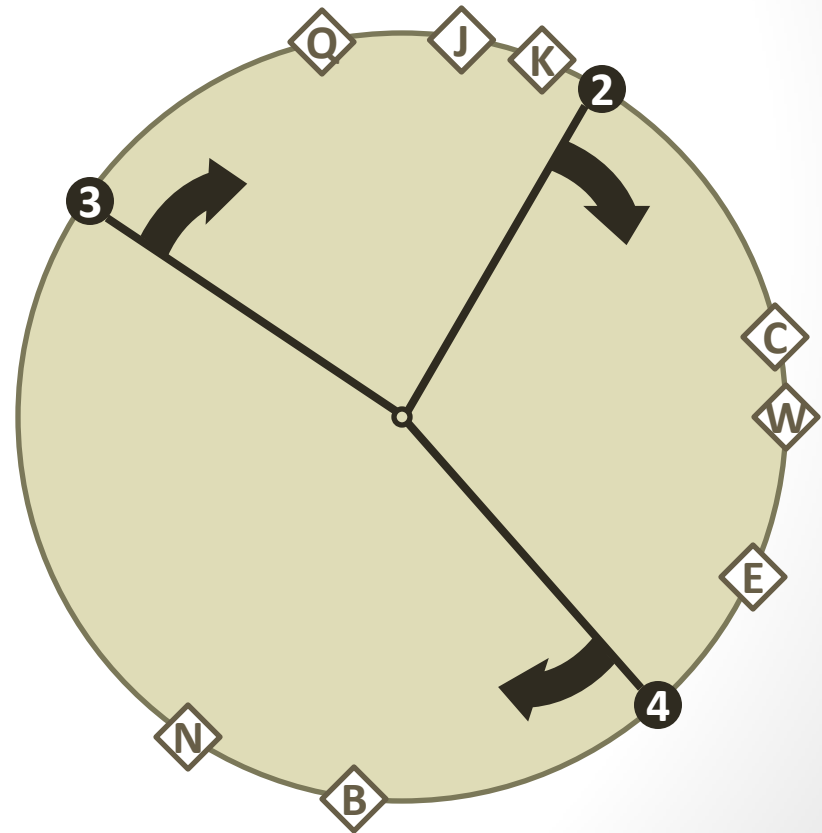
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	4
C	Data Object	14	2
E	Data Object	18	1
J	Data Object	2	3
K	Data Object	4	3
N	Data Object	35	4
Q	Data Object	57	3
W	Data Object	15	2
1	Machine 1	17	E
2	Machine 2	5	C W
3	Machine 3	51	J K Q
4	Machine 4	23	B N



Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	4
C	Data Object	14	2
E	Data Object	18	2
J	Data Object	2	3
K	Data Object	4	3
N	Data Object	35	4
Q	Data Object	57	3
W	Data Object	15	2
2	Machine 2	5	C W E
3	Machine 3	51	J K Q
4	Machine 4	23	B N



What does Scale-Out have to do with NOSQL?

- Traditional Relational Database Management Systems (RDBMS) have problems with scale-out.
- Therefore, new data base management schemes were desired.

NOSQL: Scale-out