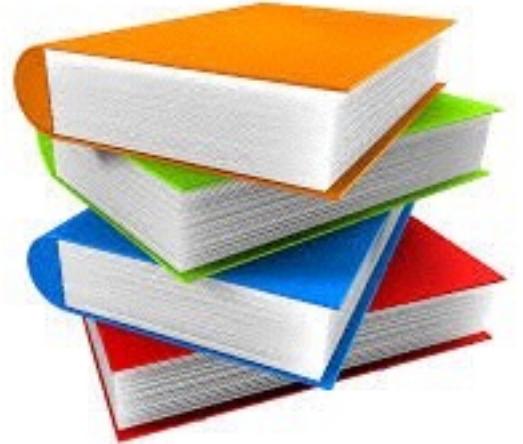


Let's say we want to set up
an e-commerce company

We'll call it "EasyBuy"

Let's see what
products we sell



books

ISBN
Title
Author
Publisher
Edition
Year
Dimensions
...



mobiles

Brand
Model Id
Color
Memory
Primary Camera
Sec Camera
Screen type
Battery Life
Display
Sim Size
Dimensions
...



t-shirts

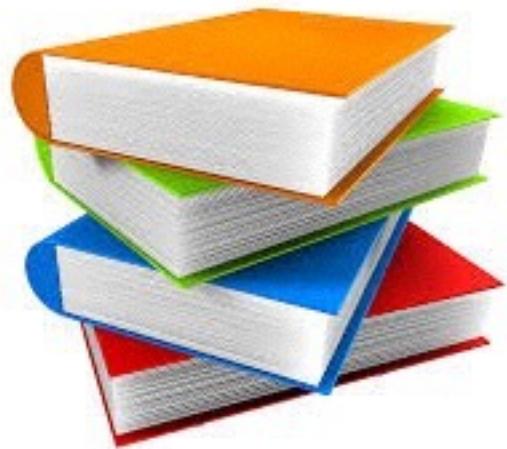
Brand
Model Id
Color
Size
Ideal For
Sleeve Type
Color Type
Euro Size
Fit
...



tv's

Brand
Model Id
Display Size
Resolution
Viewing Angle
Internet
USB Ports
HDMI Ports
Power
Sound Technology
Refresh Rate
...

Categories do not have much in common



books

ISBN
Title
Author
Publisher
Edition
Year

Dimensions

...



t-shirts

Brand
Model Id
Color
Size
Ideal For
Sleeve Type
Color Type
Euro Size
Fit

...



mobiles

Brand
Model Id
Color
Memory
Primary Camera
Sec Camera

Screen type
Battery Life
Display
Sim Size
Dimensions

...

How do we store this
data ?

In a relational
database this might
look something like ...

Products Table

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6	...
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt1	Lee	Red/Green						M	Lee Round shirt	...

id	brand	color	screen type	publishe r	author	model Id	title	size	title
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia
Tshirt1	Lee	Red/Green						M	Lee Round shirt

Notice that there are
big gaps in every row

id	brand	color	screen type	publishe r	author	model Id	title	size	title
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia
Tshirt1	Lee	Red/Green						M	Lee Round shirt

Publisher and author make no sense for a mobile phone

Empty cells occupy space
on disk

This is a major waste of
storage capacity

Especially when you have
data for millions of rows

There should be a
better way to store
this data!

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6	...
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt1	Lee	Red/Green						M	Lee Round shirt	...

id	attribute Name	attribute Value
Mob1	brand	Samsung
Mob1	color	Black
Mob1	model Id	Galaxy S6
...		
Book1	author	Khalid Hosseini
Book1	title	Kite Runner
...		
TV1	brand	Sony
TV1	color	Black
...		

Converting
sparse rows
into
column value
pairs

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6	...
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt1	Lee	Red/Green						M	Lee Round shirt	...

id	attribute Name	attribute Value
Mob1	brand	Samsung
Mob1	color	Black
Mob1	model Id	Galaxy S6
Mob1	title	Samsung Galaxy6
Book1	author	Khalid Hosseini
Book1	title	Kite Runner
...		
TV1	brand	Sony
TV1	color	Black
...		...

Let's see how the data for a mobile phone is stored in the table

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6	...
Book 1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt 1	Lee	Red/Green						M	Lee Round shirt	...

Column "id"
remains as is

id	attribute Name	attribute Value
Mob1	brand	Samsung
Mob1	color	Black
Mob1	model Id	Galaxy S6
Mob1	title	Samsung Galaxy6

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6	...
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt1	Lee	Red/Green						M	Lee Round shirt	...

Attribute names

We create a new row for every attribute of Mob1

id	attribute Name	attribute Value
Mob1	brand	Samsung
Mob1	color	Black
Mob1	model Id	Galaxy S6
Mob1	title	Samsung Galaxy6
Book1	author	Khalid Hosseini

Only those attributes which have values for a mobile phone!

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6	...
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt1	Lee	Red/Green						M	Lee Round shirt	...

Attribute names

id	attribute Name	attribute Value
Mob1	brand	Samsung
Mob1	color	Black
Mob1	model Id	Galaxy S6
Mob1	title	Samsung Galaxy6
Book1	author	Khalid Hosseini

Notice some columns are missing

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy	...
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt1	Lee	Red/Green						M	Lee Round shirt	...

Attribute value

id	attribute Name	attribute Value
Mob1	brand	Samsung
Mob1	color	Black
Mob1	model Id	Galaxy S6
Mob1	title	Samsung Galaxy6
Book1	author	Khalid Hosseini

The “attributeValue” column stores the actual value of each attribute

id	brand	color	screen type	publisher	author	model Id	name	size	title	...
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy6	...
Book1				Riverhead	Khalid Hosseini		Kite Runner		Kite Runner	...
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia	...
Tshirt1	Lee	Red/Green						M	Lee Round shirt	...

Attribute names

id	attribute Name	attribute Value
Mob1	brand	Samsung
Mob1	color	Black
Mob1	model Id	Galaxy S6
Mob1	title	Samsung Galaxy6
Book1	author	Khalid Hosseini

Thus optimizing space!

This is how a column
oriented database
stores data

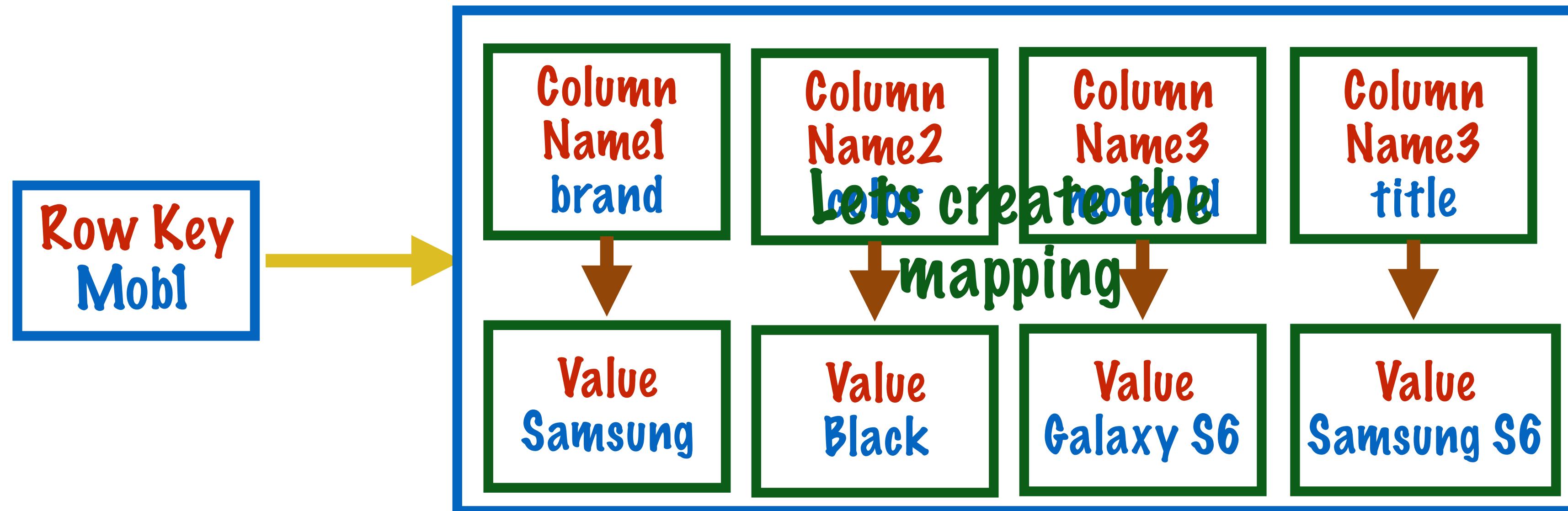
Let's see the layout in
a column oriented
store visually

Row Key



id	brand	color	screen type	publisher	author	model Id	title	size	title
Mob1	Samsung	Black				Galaxy S6			Samsung Galaxy S6
TV1	Sony	Black	Ultra HD			Bravia			Sony Bravia

COLUMN ORIENTED Database



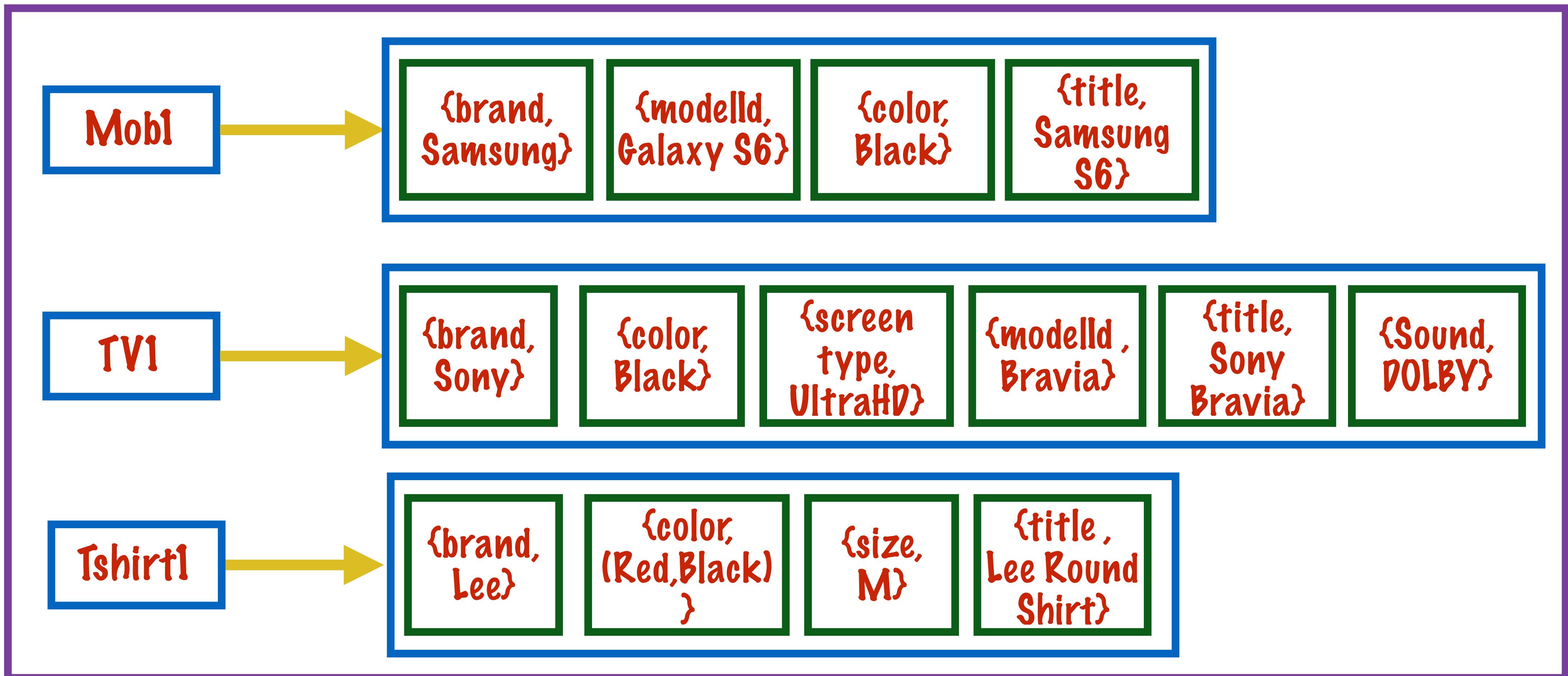
COLUMN ORIENTED Database

What do we gain?

Disk usage is
minimised by not
saving empty cells

Not bound tightly
to table schema
Can add columns
in future

COLUMN ORIENTED Database



Products Table

Requirements from a
data store for a catalog

column oriented

As EasyBuy grows and
becomes popular



PHOTO: DARRIN HAUCK/GETTY

Millions of
customers



Increase In
Orders



**Millions of
products**



Data store size in petabytes

Millions of operations
per second on data store

A single system cannot
cater to this **high demand**



*We need a cluster of
datastore machines*



Where data is
distributed evenly
among the nodes



Requirements from a
data store for a catalog

column oriented

distributed

Evenly distributed data



Which system does the application **interact** with?





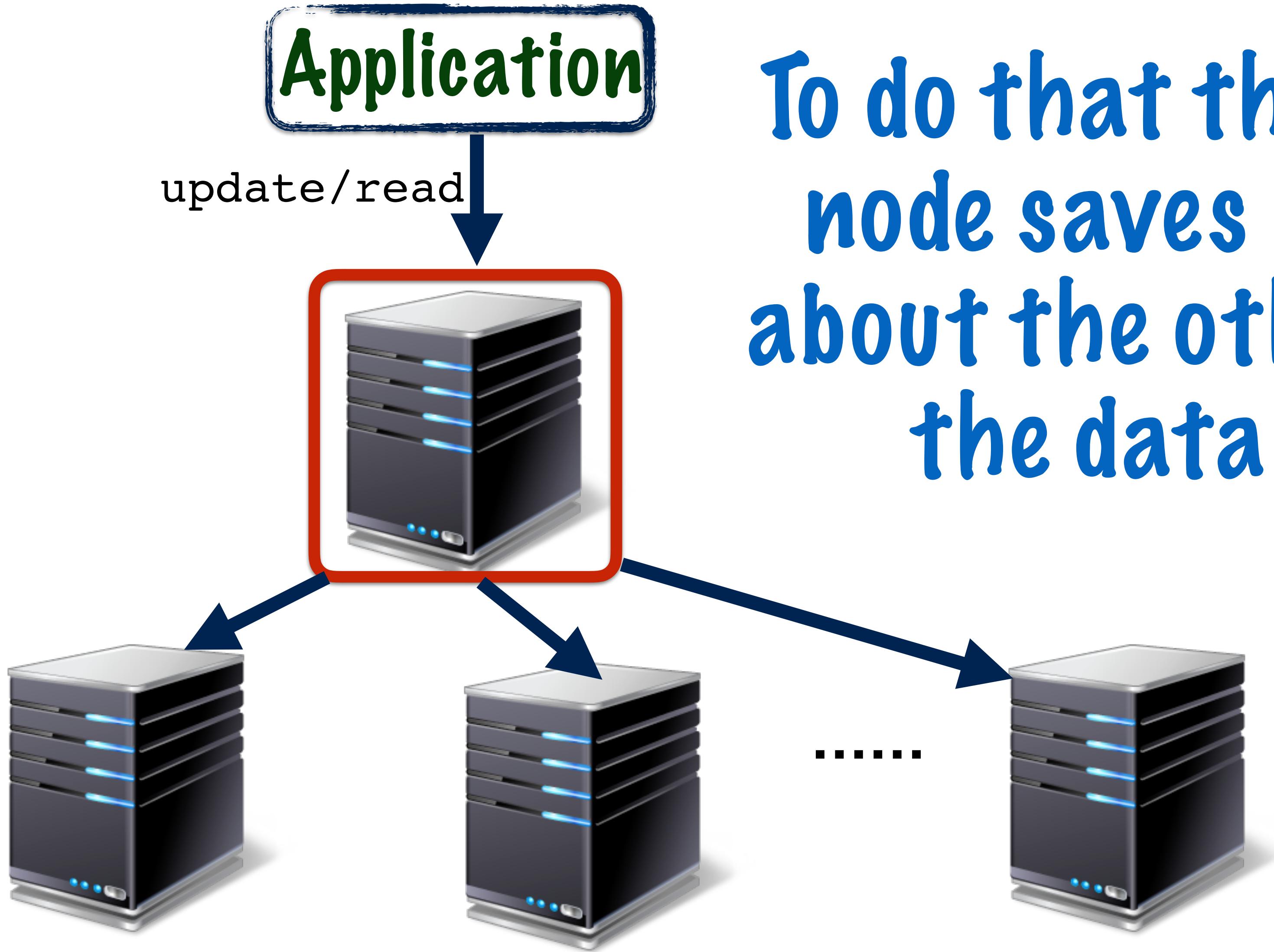
Let's make this node an interface between the cluster and application

Application

update/read



The interface node delegates operations to other nodes



To do that the “interface” node saves information about the other nodes and the data they hold

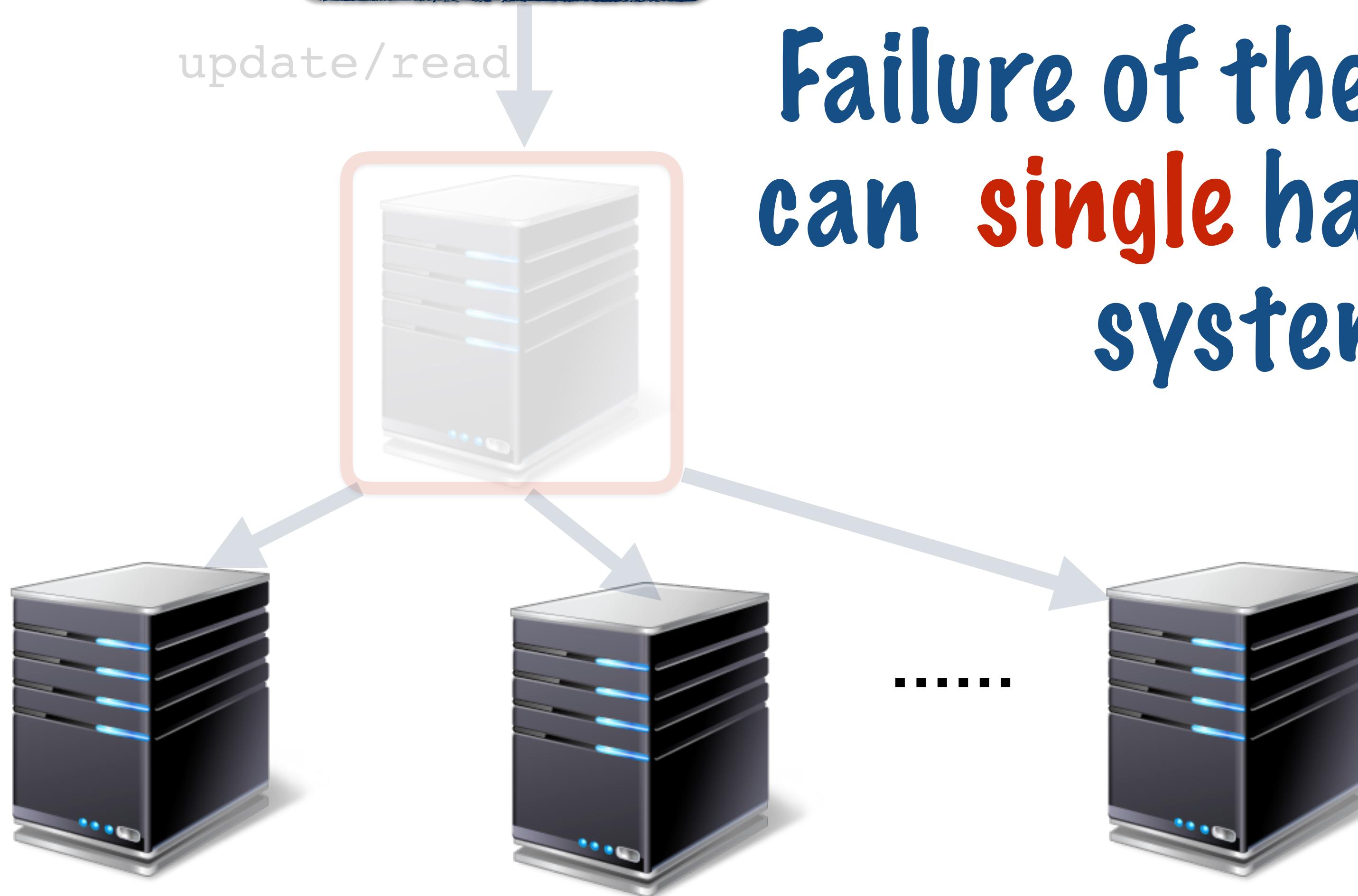
Application

update/read

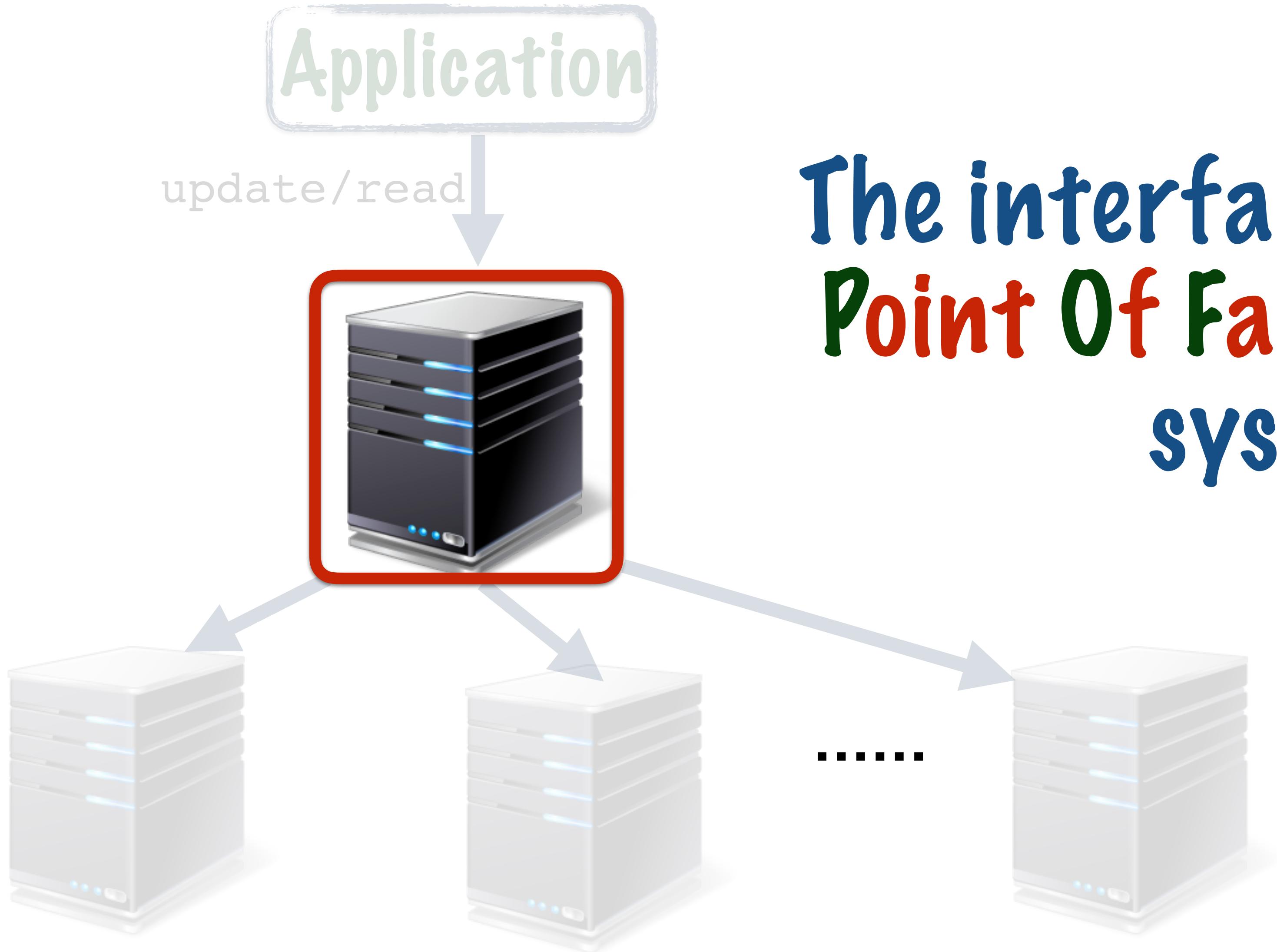


What if the interface node goes down ?

Application



Failure of the interface node
can **single handedly** result in
system failure



The interface is a **Single Point Of Failure** for the system

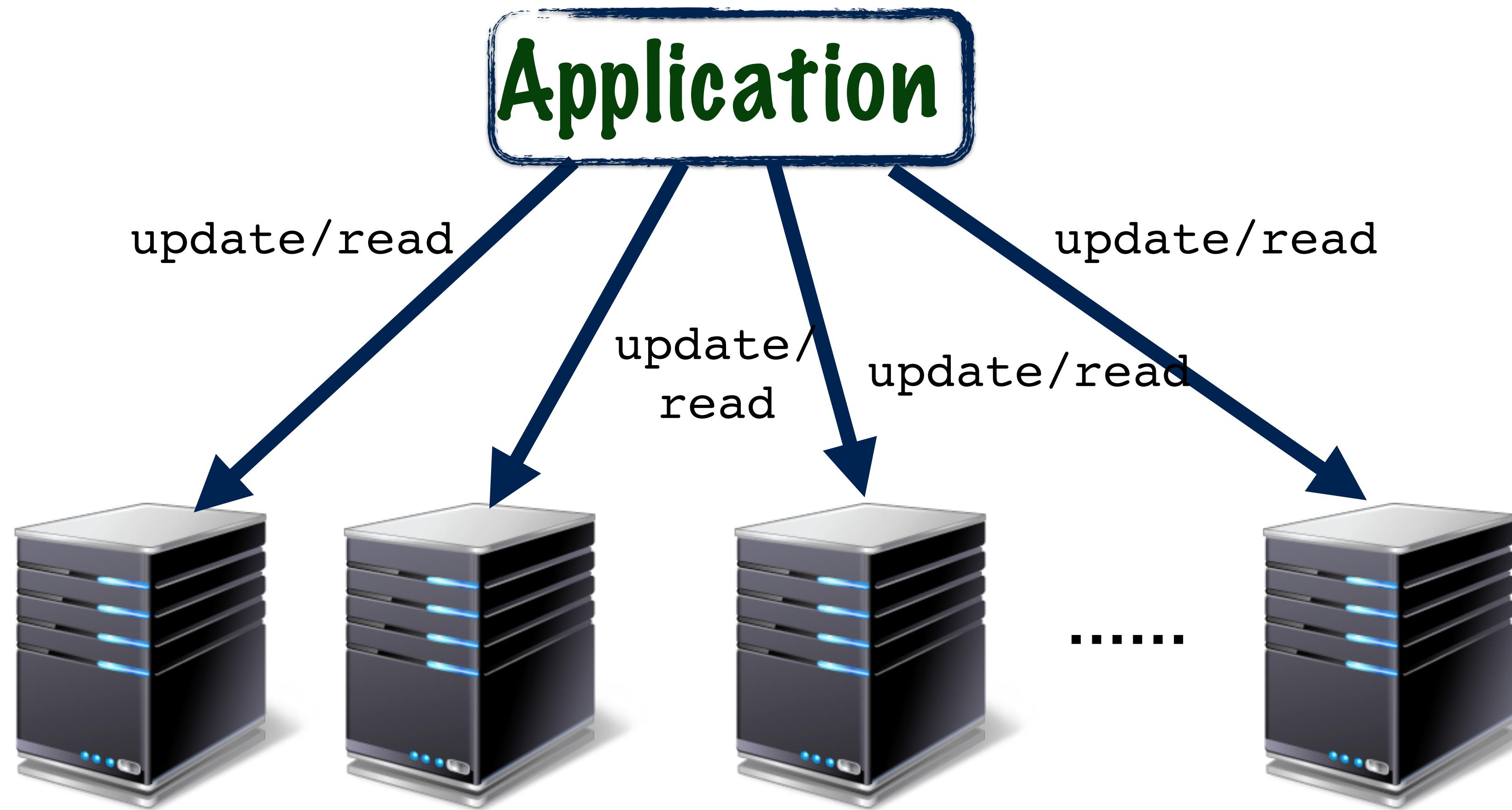
How do we avoid SPOF?

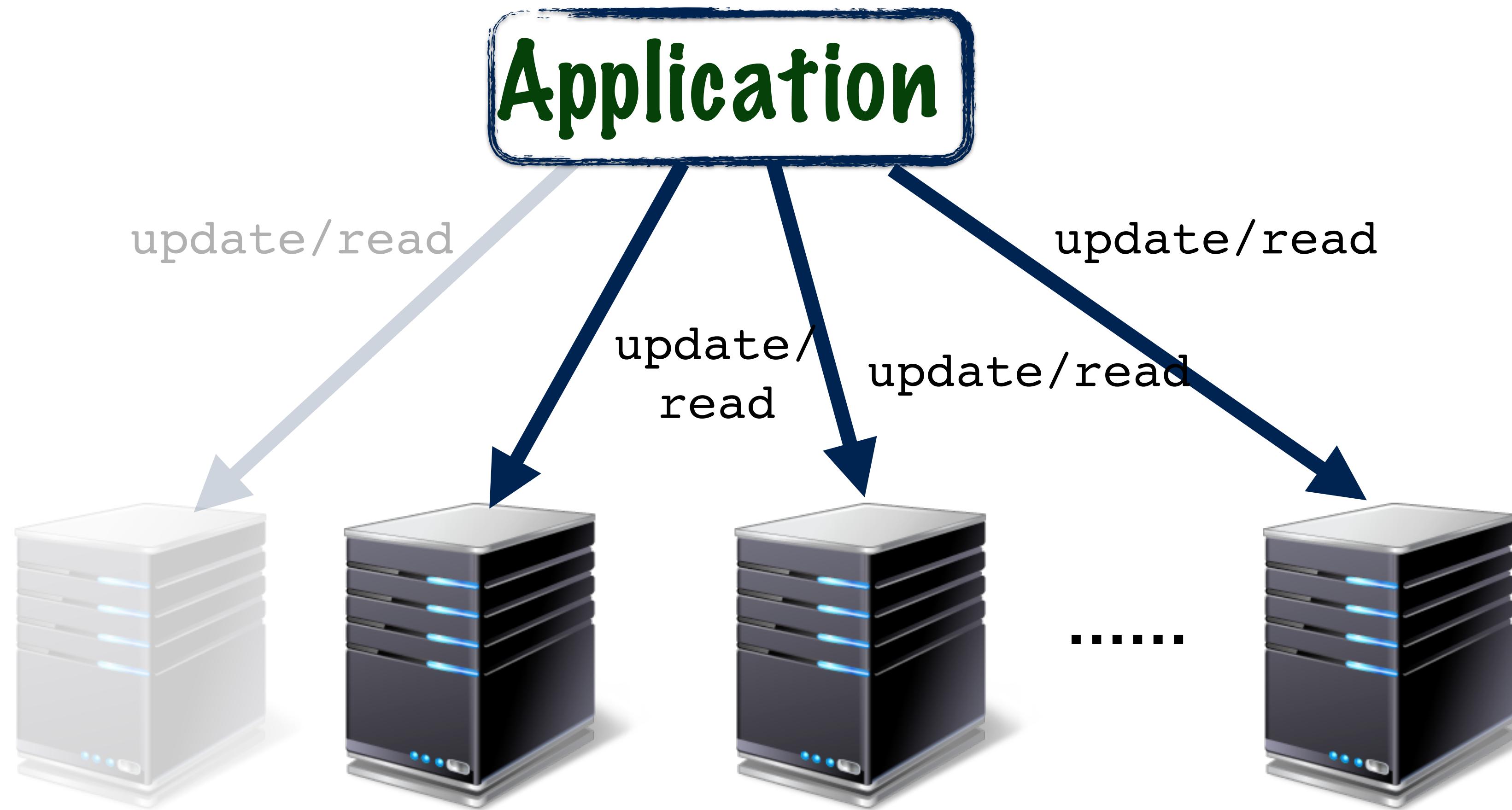
Coming back to our cluster



Let the application select
any node as the “interface”

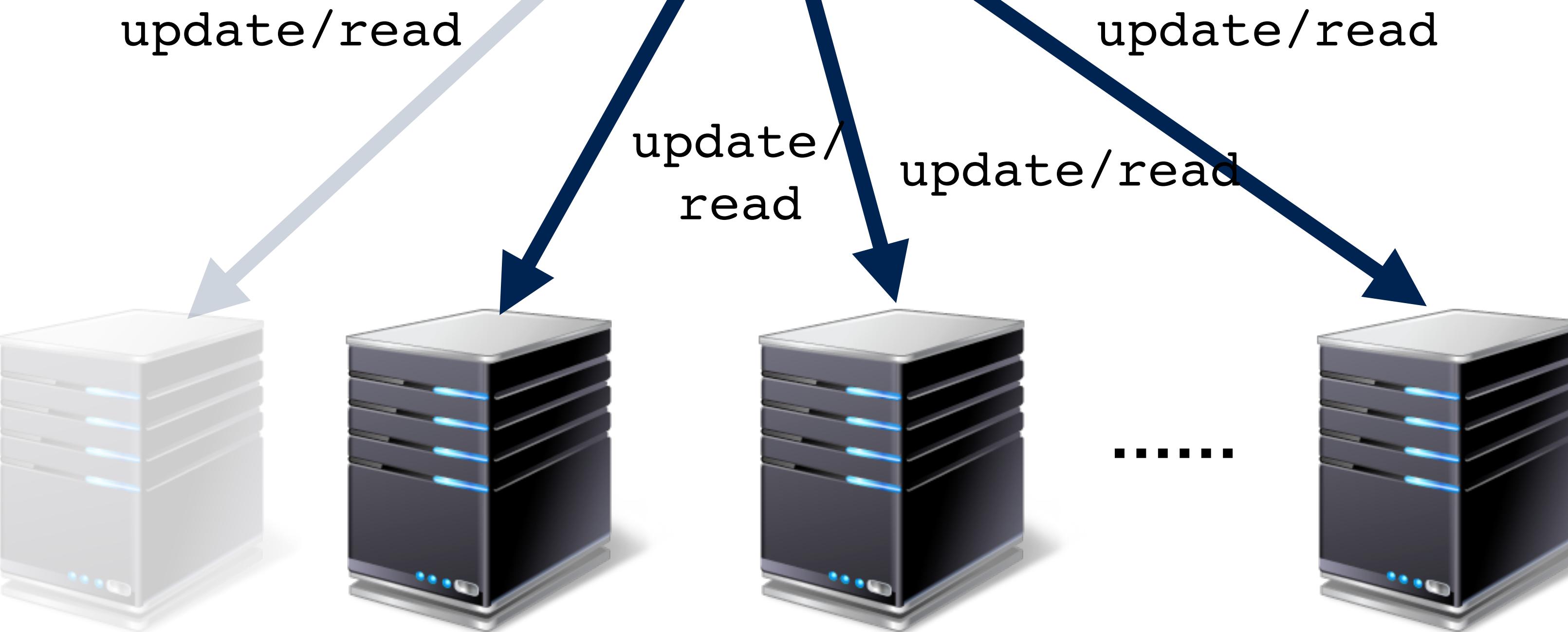




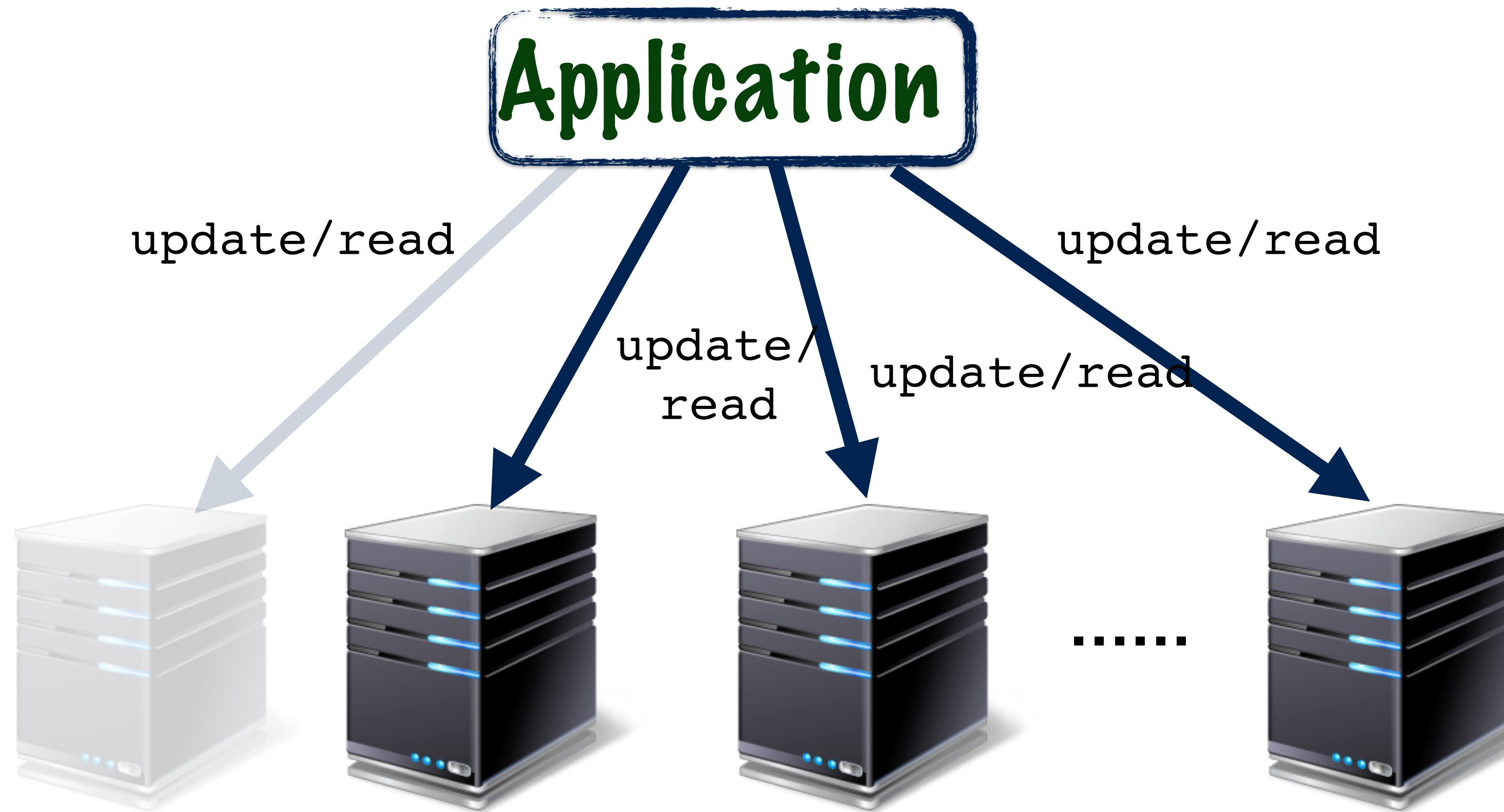


**Let's see what happens when
a node fails**

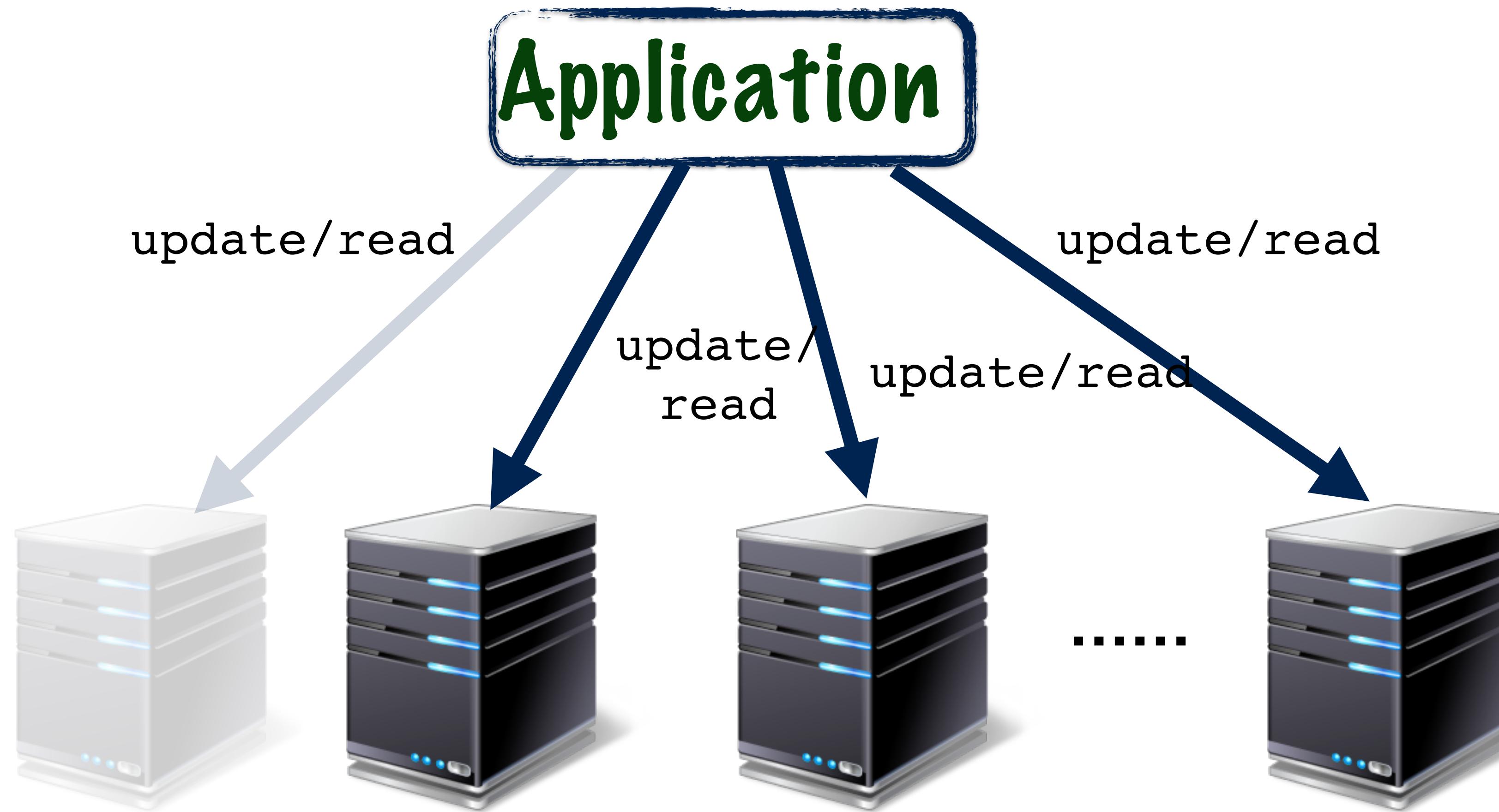
Application



The application can still
continue running!



**The system is not reliant
on a single node**



This architecture is called
de-centralized

Requirements from a data store for a catalog

column oriented

distributed

de-centralized

For any system where we
store huge amounts of
data

What options do we have?

distributed

column oriented

de-centralized

HBase

MongoDB

Cassandra

distributed

column oriented

de-centralized

HBase

MongoDB

Cassandra

distributed

HBase

column oriented

de-centralized

Cassandra

distributed

HBase

column oriented

de-centralized

Cassandra

distributed

column oriented

de-centralized

Cassandra

What is Cassandra?

Cassandra is a database

column-oriented

decentralised
architecture

high
availability

elastically
scalable
tunable
consistency

Cassandra

decentralization and replication
make it fault-tolerant

supports secondary indexes

has a sql-like query language
called **CQL**

Cassandra is a great fit

When you have data in petabytes

data is columnar

eg Catalog data, notification data

But Cassandra has a
few **limitations** as
compared with
RDBMS

Limitations

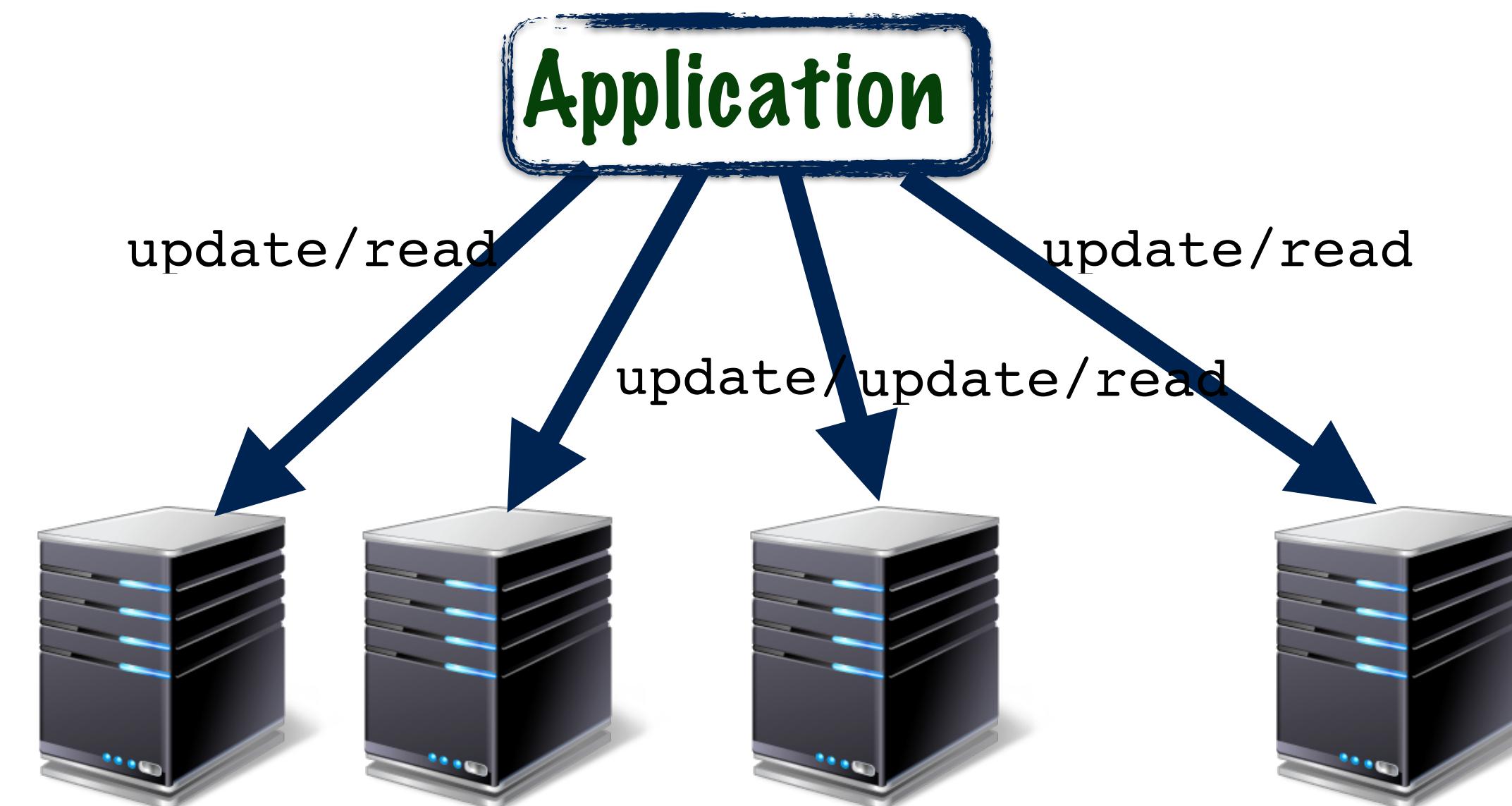
NOT ACID compliant

Cassandra provides tunable consistency

You can have stronger consistency at the expense of availability

Limitations

NOT ACID compliant



To ensure **fault-tolerance**, multiple copies of data are made and stored on different nodes



Mob1 Data



Mob1 Data

.....



Mob1 Data

Data of Mob1 is stored on 3 nodes

Application

update Mob1



Mob1 Data



.....

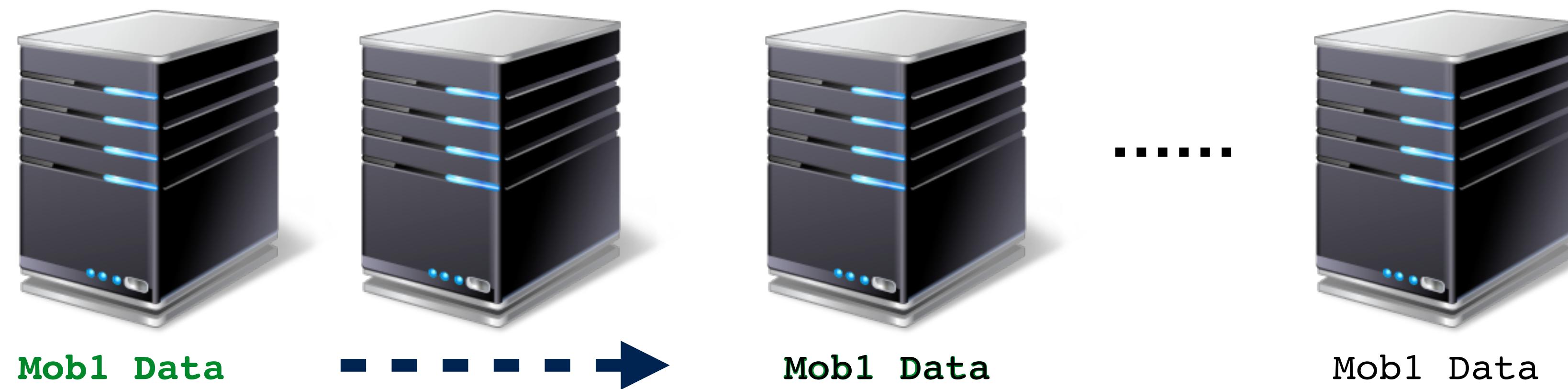


Mob1 Data

Application updates Mob1 data

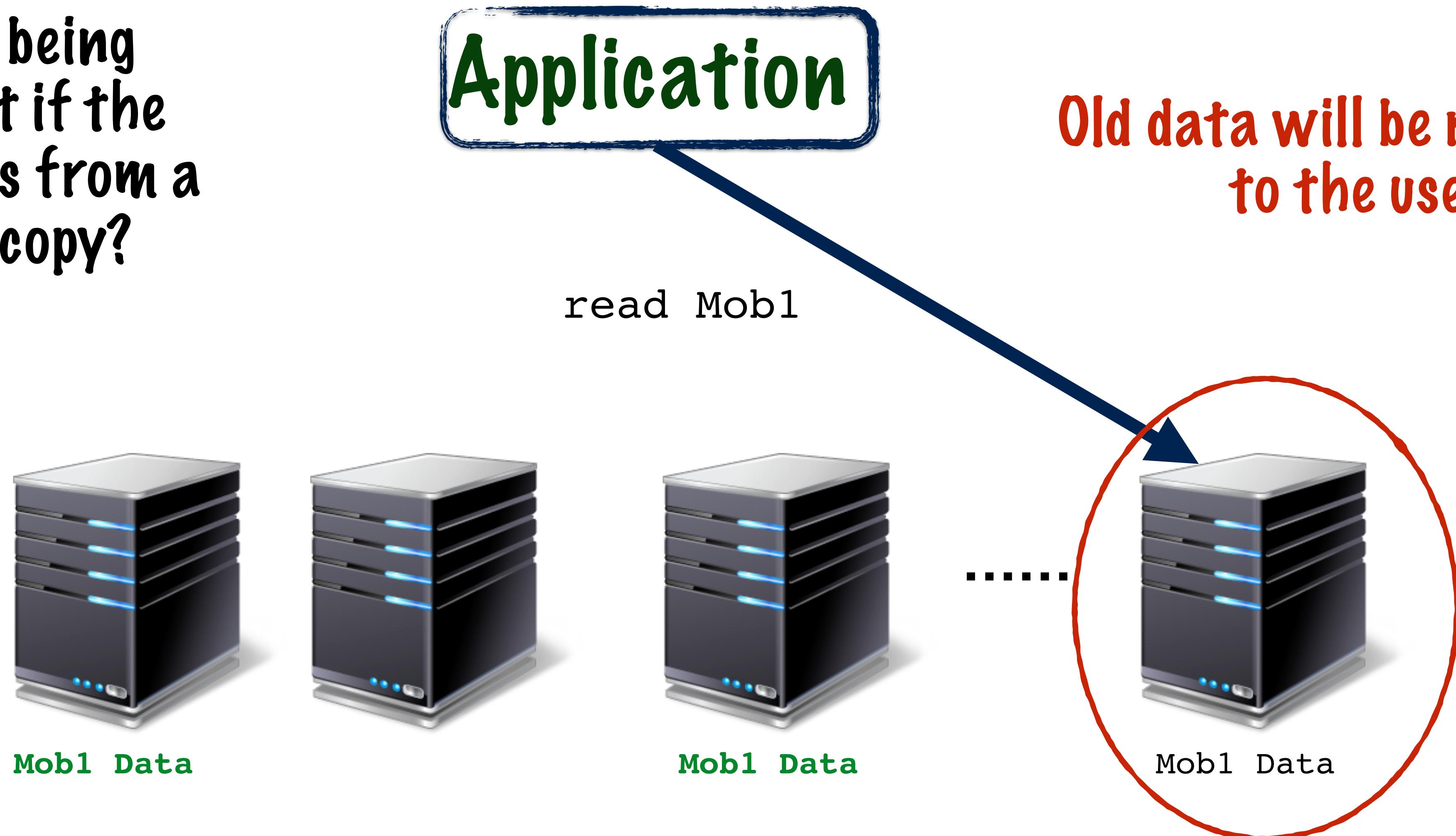
By default, Cassandra
returns success for
update when data on 1
node is updated

Application



Data is then updated on remaining
nodes

While data is being updated - what if the application reads from a non-updated copy?



So the data is **not** consistent

To avoid this, Cassandra provides tunable consistency

Consistency can be configured so that Cassandra can return success for a write after

all nodes have updated the data

a set of nodes have updated the data

Any of these consistency levels can be set as per application requirements

Cassandra should not be used

if data can be managed on a single system

if we want to instantly read what
is written

eg Bank accounts, product orders

So.. Where does
Cassandra fit with
Hadoop?

HADOOP

is a distributed computing framework
developed and maintained by
the Apache Software Foundation

HADOOP

Applications where Hadoop is Used

rendering a
3D film

Product
Recommendations

Applications that involve
similar computations
performed over huge data

HADOOP

distributed

Has a distributed file
system **HDFS**

Has a distributed
data base **HBase**

computing
framework

parallel
processing on
the stored data
MapReduce

HADOOP

distributed

Has a distributed file
system **HDFS**

Has a distributed
data base

HBase

computing
framework

framework to
process the
stored data

Cassandra
MapReduce



So can I use
Cassandra in
Hadoop?

YES

DataStax has a distribution **Brisk**
which combines Cassandra and Hadoop

In Brisk, the map-reduce jobs are run
over the cassandra database

BUT..

The underlying filesystem for Hbase is
HDFS

So it is very convenient to use Hbase if
you already have a Hadoop stack

BUT..

Cassandra is mostly preferred for real-time very fast lookups

HBase is mostly preferred for heavy ETL batch jobs with lower latency requirements

BUT..

You should consider in depth the
requirements of your application
before going ahead

So.. Who is using
Cassandra?

