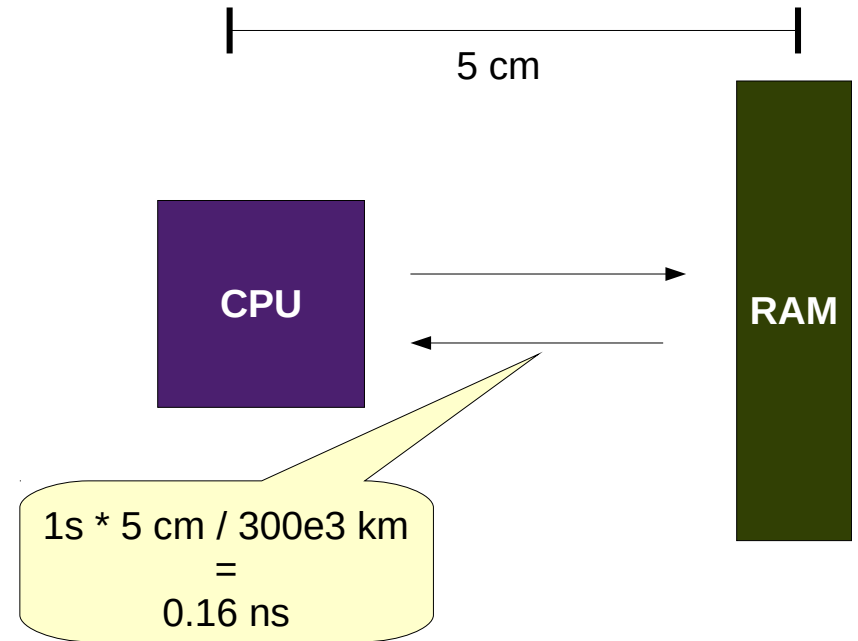


Scalability Minimal Theory

This guide will teach you:

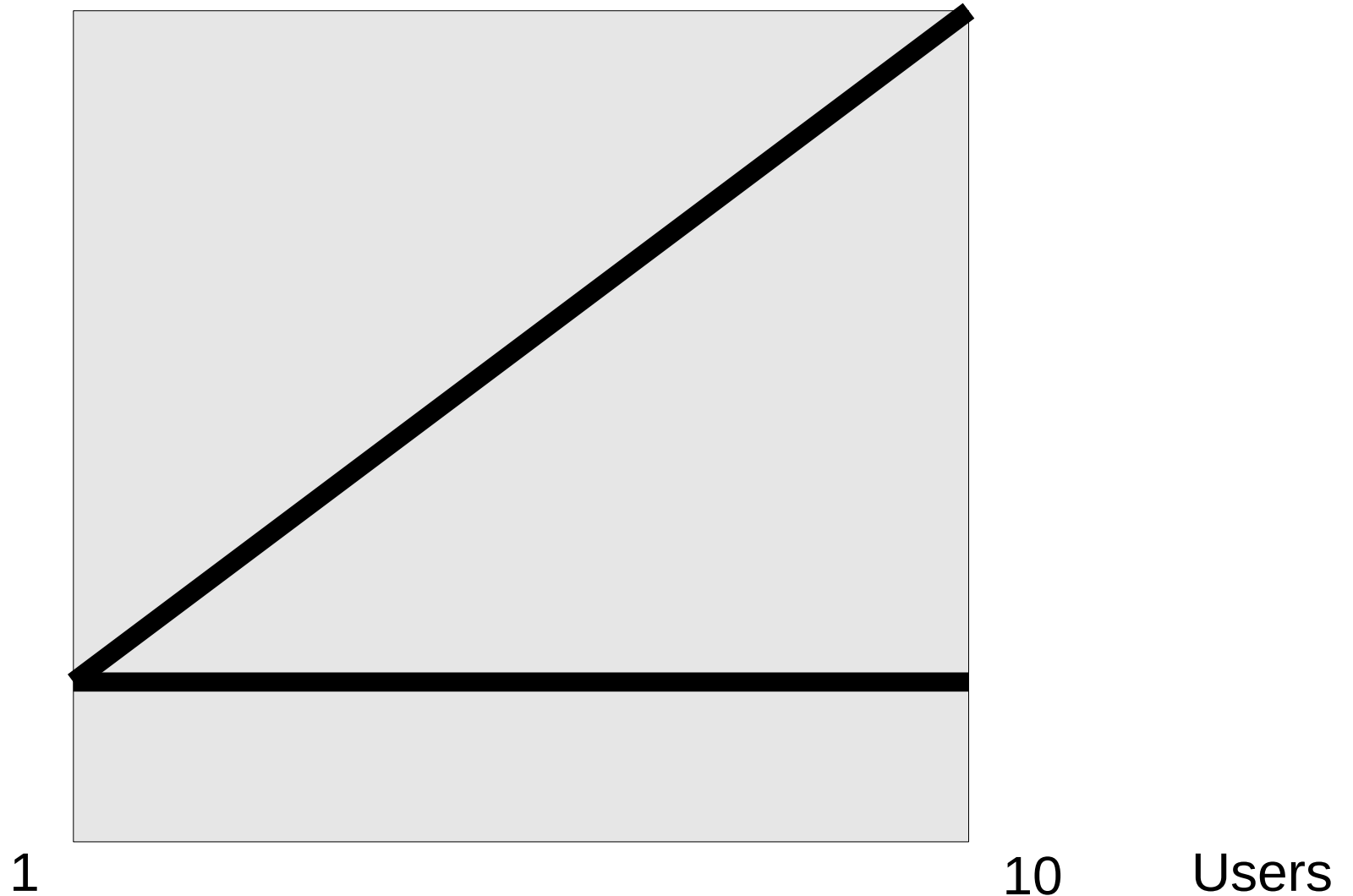
- The Speed of Light
- The Frequency of CPU
- The Latency of Network
- The Speed of Disk
- Solutions to Slowness



More about Vifib
www.vifib.org

10 Users, 10 ZEO Clients

Time
Per
Page



Why are Software Slow ?

```
d = {}  
i = 1e12 # 1 TB  
while i > 0; # at least one CPU instruction  
    d[i] = True # at least one memory access  
    i = i - 1 # at least one CPU instruction
```

How long does it take at least ?

The Frequency of CPU: 2Ghz

CPU

```
while i > 0;
```

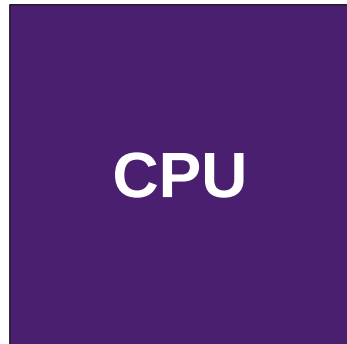
```
d[i] = True
```

```
i = i - 1
```

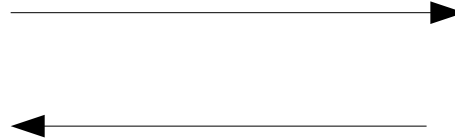
$$\begin{aligned} 3 * 1 / 2\text{Ghz} \\ = \\ 1.5 \text{ ns} \end{aligned}$$

The Speed of Light: 300,000 km/s

|-----|
5 cm



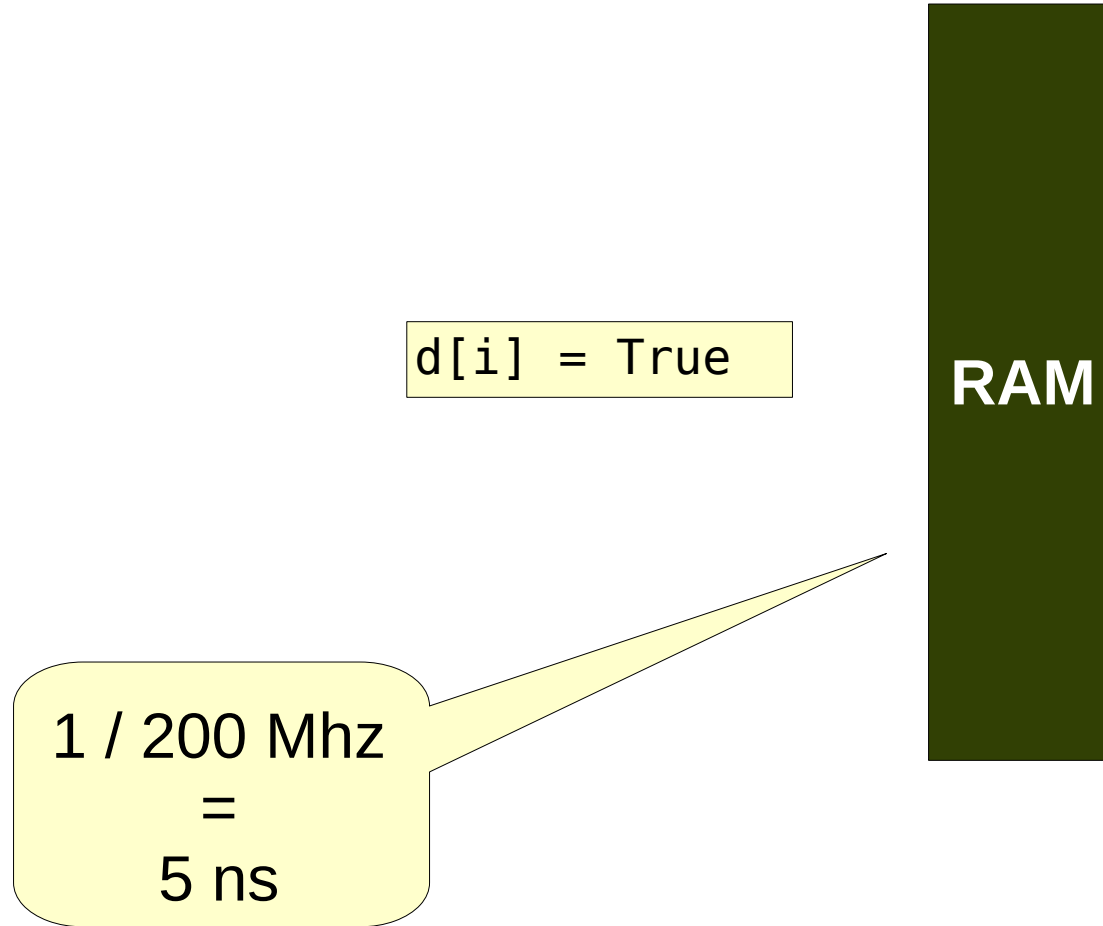
CPU



RAM

$$1s * 5 \text{ cm} / 300e3 \text{ km} = 0.16 \text{ ns}$$

The Frequency of RAM: 200 Mhz



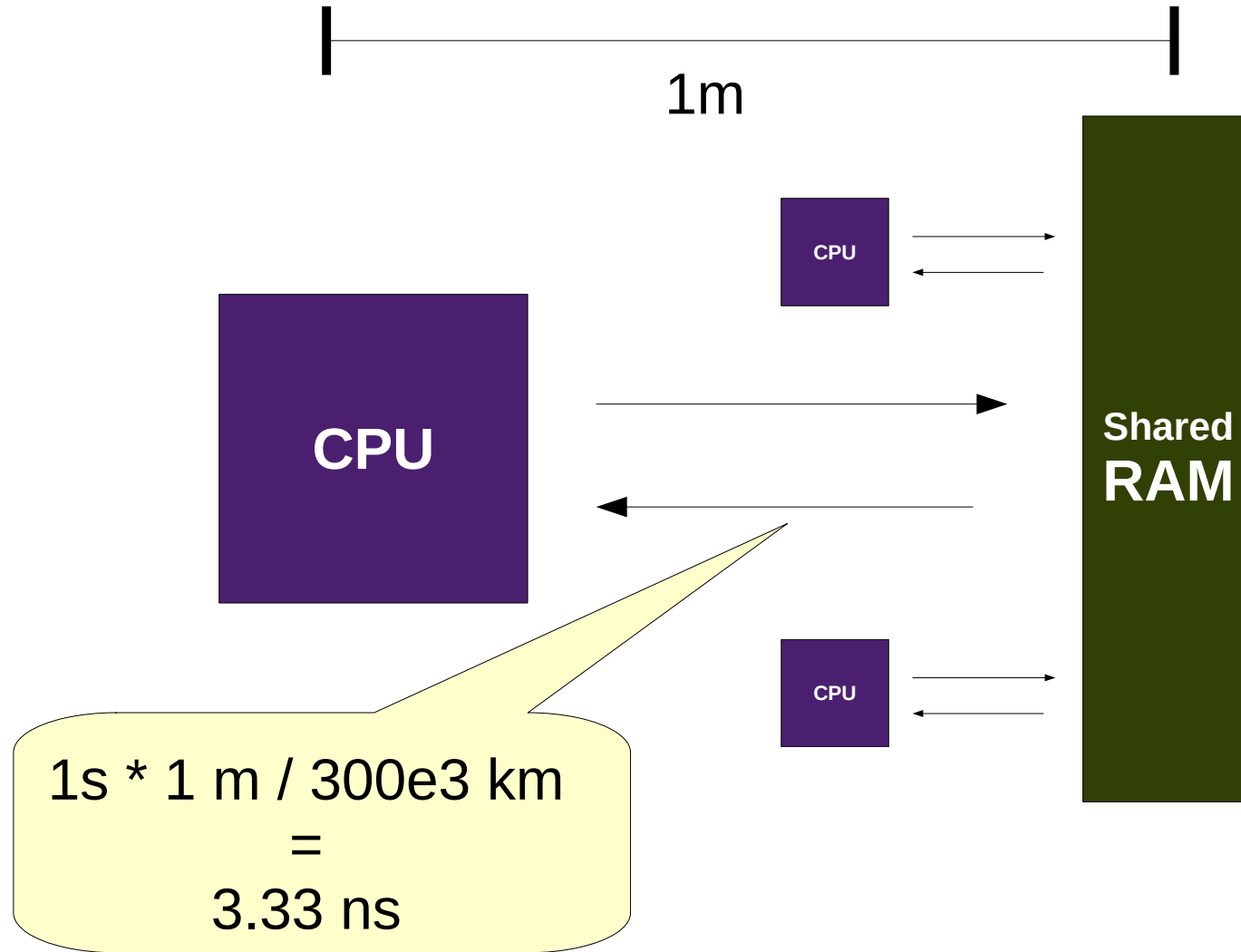
Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 5 + 0.16 * 2 + 0.5 = 5.82 ns  
    i = i - 1 # 0.5 ns  
# total = 6.82 ns
```

$$1e12 * 6.82 \text{ e-9} = 6820 \text{ s} = \text{about 2 hours}$$

$$16e9 * 6.82 \text{ e-9} = 109 \text{ s} = 2 \text{ min}$$

The Speed of Light: 300,000 km/s



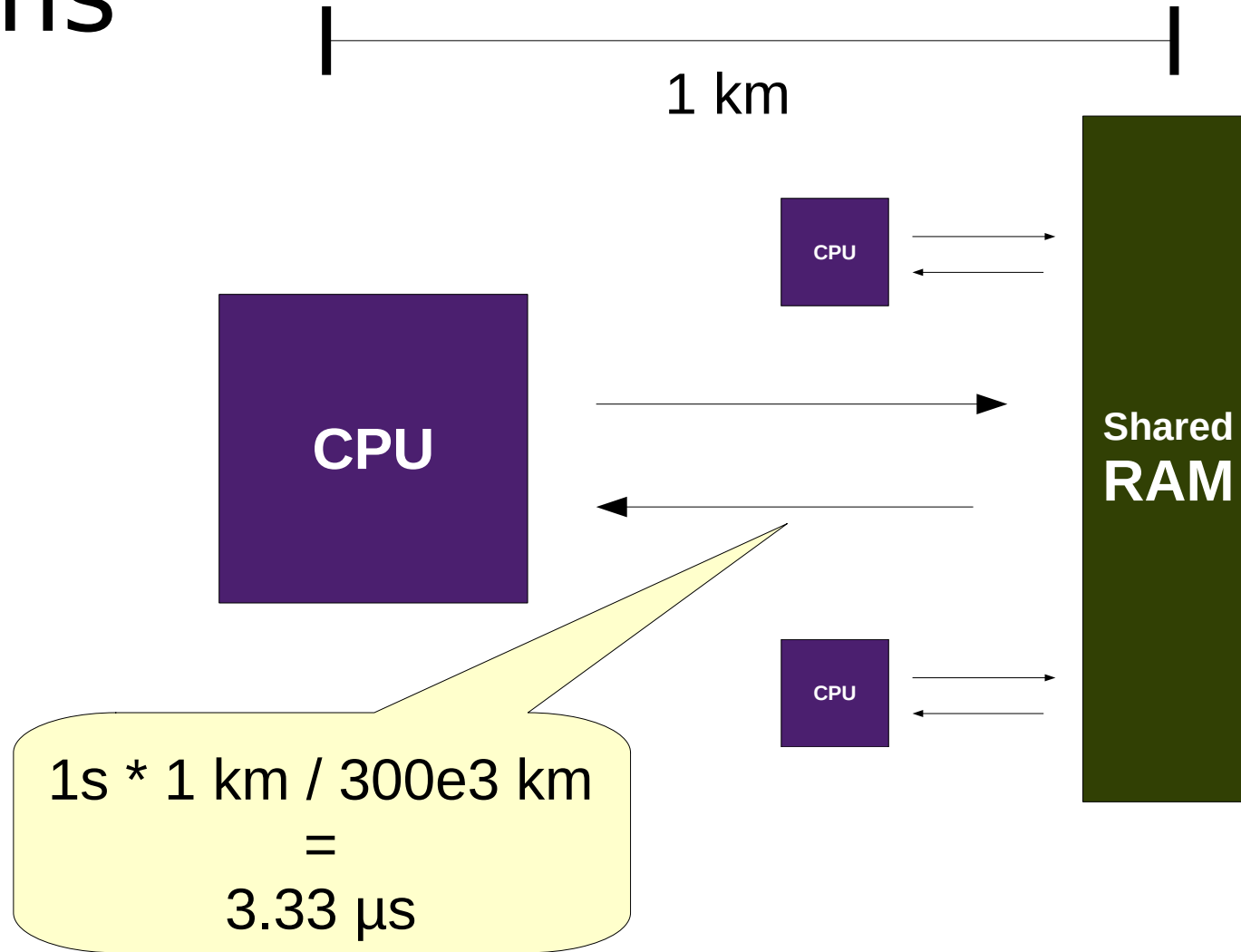
Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*3 + 6.66 = ns  
    i = i - 1 # 0.5 ns  
# total = 23.16 ns
```

$$1e12 * 23.16 = 23160 \text{ s} = 6.4 \text{ hours}$$

$$16e9 * 23.16 = 370 \text{ s} = 6 \text{ minutes}$$

The Latency of Network: 1 μs to 1ms



Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5 + 3300*2 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 3316.5 ns
```

$$1e12 * 3316.5 = 38 \text{ days}$$

$$16e9 * 3316.5 = 14 \text{ hours}$$

The Speed of Disks: 10,000 rpm

`d[i] = True`

Disk

$$\begin{aligned} 60s * 10e3 / 2 \\ = \\ 3 \text{ ms} \end{aligned}$$

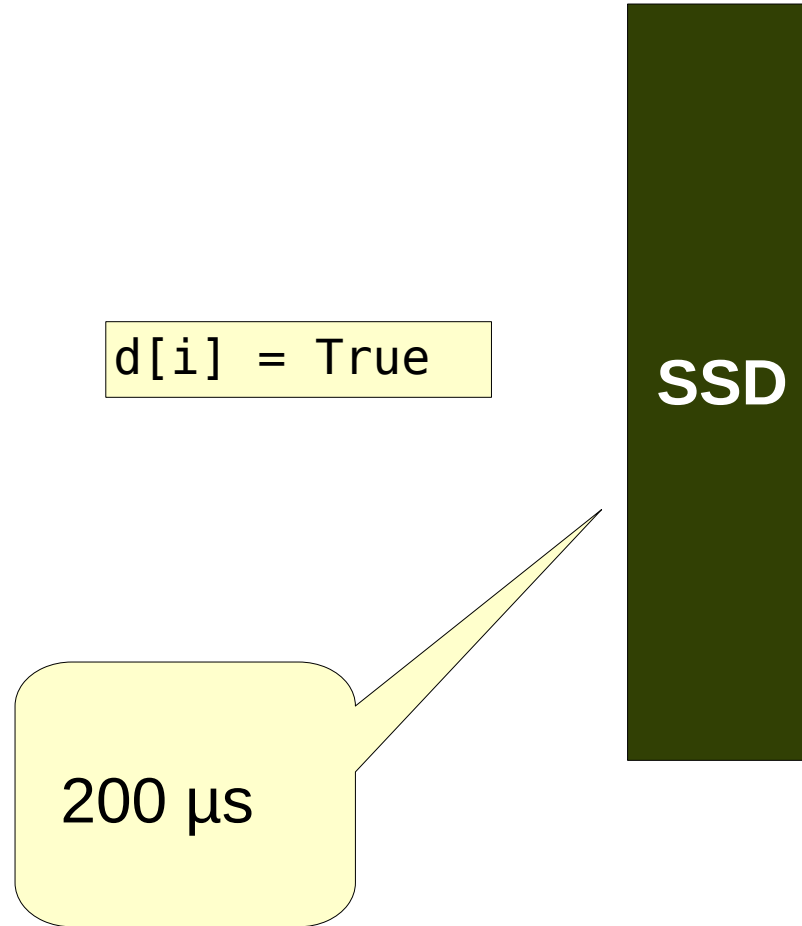
Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5 + 3000000 + 3.33*2 = ns  
    i = i - 1 # 0.5 ns + = ns  
# total = 3000023.16 ns
```

$$1e12 * 3000023.16 = 95 \text{ years}$$

$$16e9 * 3000023.16 = 1.5 \text{ year}$$

The Latency of SSD: 200 μ s or better



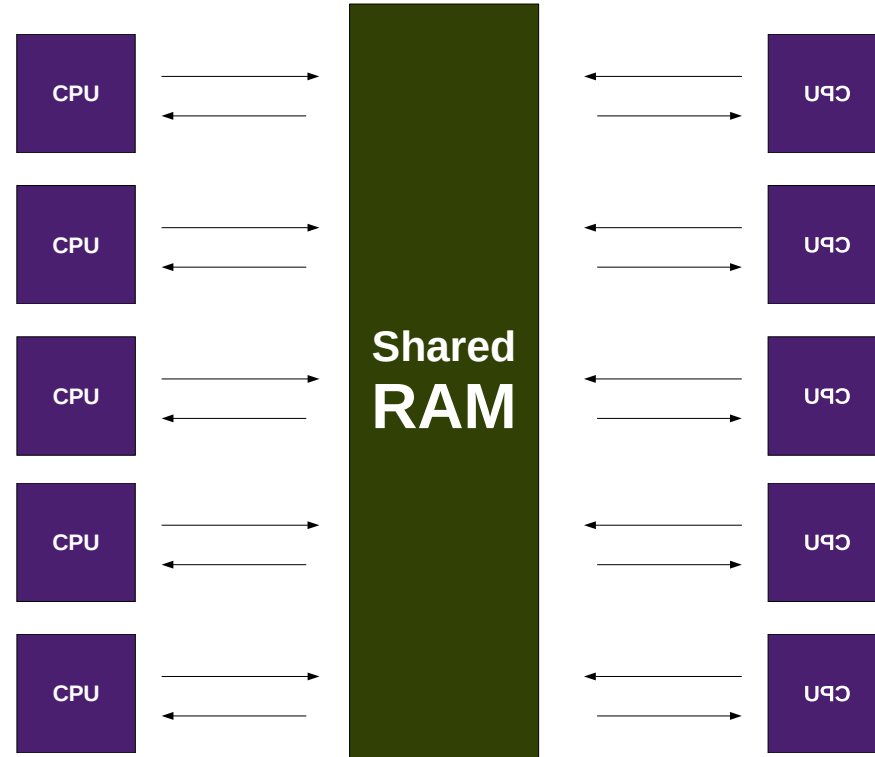
Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5 + 200000 + 3.33*2 = ns  
    i = i - 1 # 0.5 ns + = ns  
# total = 200023.16 ns
```

$$1e12 * 200023.16 = 6 \text{ years}$$

$$16e9 * 200023.16 = 37 \text{ days}$$

Access Conflict (Serialized)



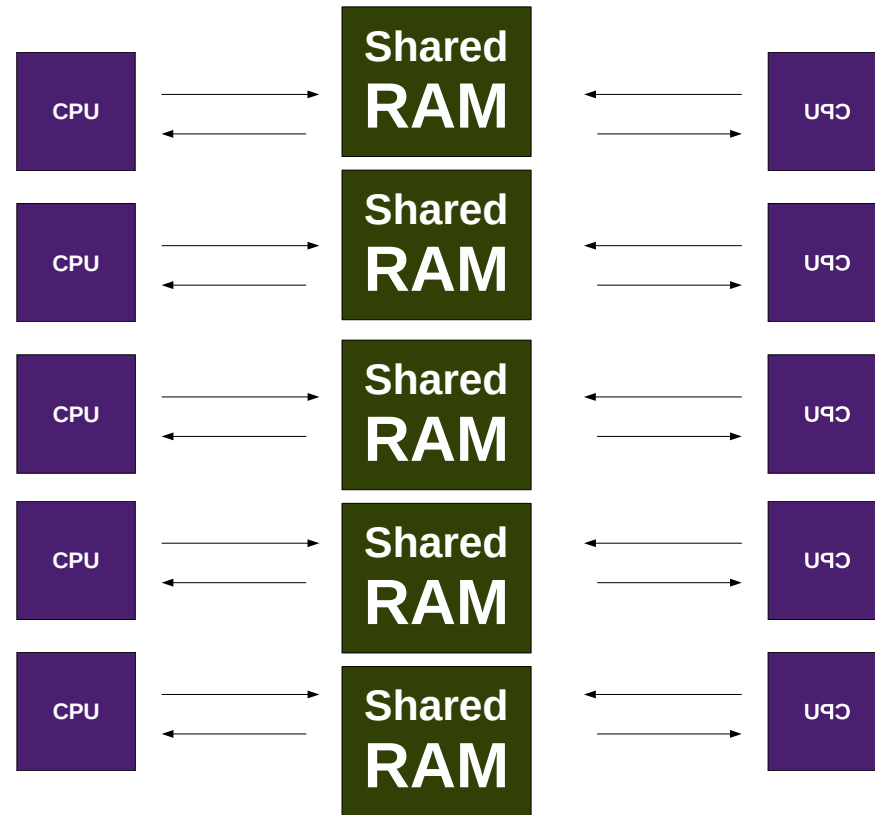
Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 66011.5 ns
```

$$1e12 * 66011.5 = 2 \text{ years}$$

$$16e9 * 66011.5 = 12 \text{ days}$$

Solving Access Conflict by Sharding



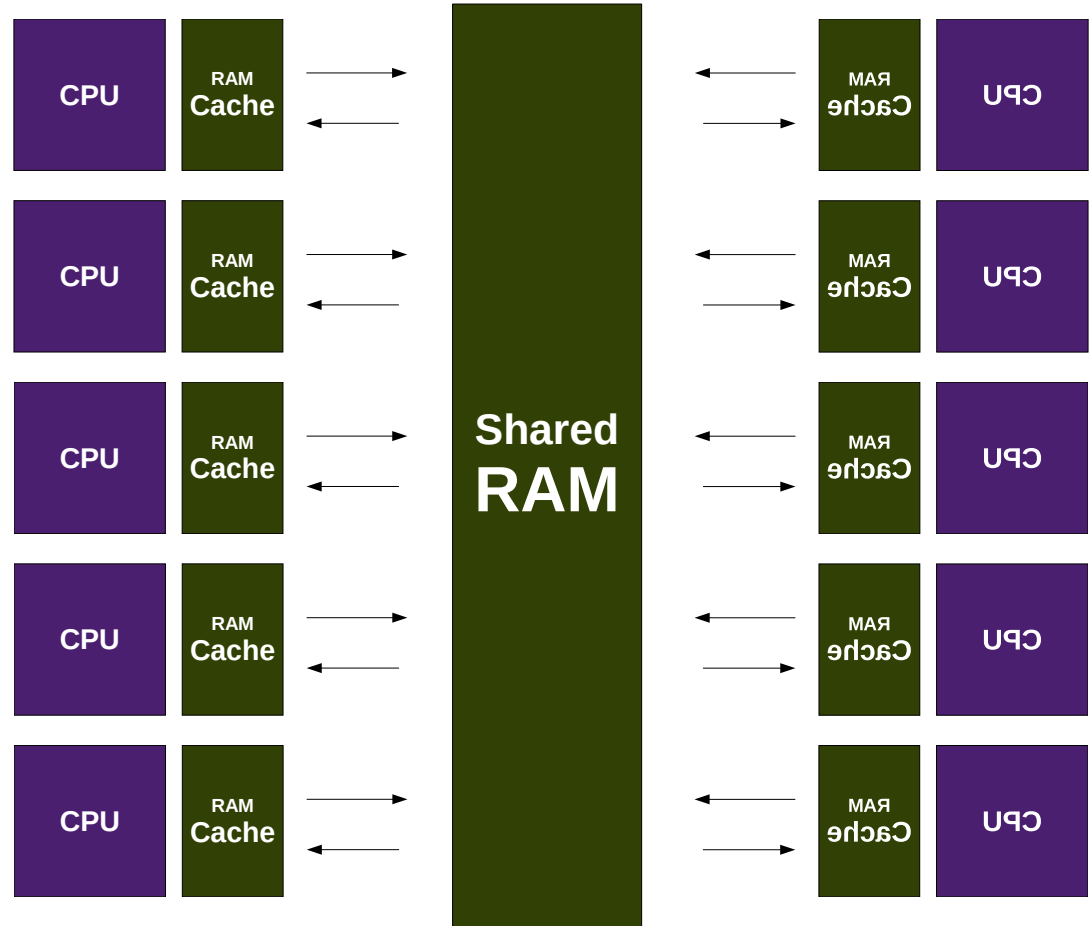
Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*2 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 13211.5 ns
```

$$1e12 * 13211.5 = 152 \text{ days}$$

$$16e9 * 13211.5 = 2.42 \text{ days}$$

Solving Access Conflict by Caching



Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 5 + 0.16 * 2 + 0.5 = 5.82 ns  
    i = i - 1 # 0.5 ns  
# total = 6.82 ns
```

$$1e12 * 6.82 \text{ e-9} = 6820 \text{ s} = \text{about 2 hours}$$

$$16e9 * 6.82 \text{ e-9} = 109 \text{ s} = 2 \text{ min}$$

Solving Access by Beautiful Code

```
d = {}  
i = 1e12 # 1TB or RAM  
while i-- > 0; # 0.5 ns  
    d[i] = True # 5 + 0.16 * 2 + 0.5 = 5.82 ns  
    # total = 6.32 ns
```

$$1e12 * 6.32 \text{ e-}9 = 6320 \text{ s} = \text{about 2 hours}$$

$$16e9 * 6.32e-9 = 101 \text{ s} = 2 \text{ min}$$

Solving Access by Better Algorithm

```
d = {}  
i = 1e6 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 99010.5 ns
```

$$1e6 * 99010.5 = 1.65 \text{ minute}$$

Better Algorithm for 90% cases

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 66011.5 ns
```

10%

```
d = {}  
i = 1e6 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 99010.5 ns
```

90%

$$(1.65 \text{ minutes} * 90 + 2 \text{ years} * 10) / 100 = 2.4 \text{ months}$$

Be Radical



0 s

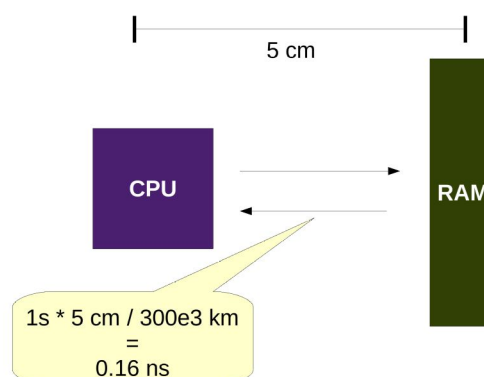
What About ERP5

- Postpone
 - Ajax Menus
 - Ajax Display
 - Activities
- Access Conflict
 - ID Generation
 - Access to Same Btree (HBTree)
- Caching
 - portal_caches
 - HTTP cache
- Sharding
 - Multiple Data.fs
 - NEO
- Algorithm
 - SQL Optimization
 - Catalog vs. ZODB

Scalability Minimal Theory

This guide will teach you:

- The Speed of Light
- The Frequency of CPU
- The Latency of Network
- The Speed of Disk
- Solutions to Slowness

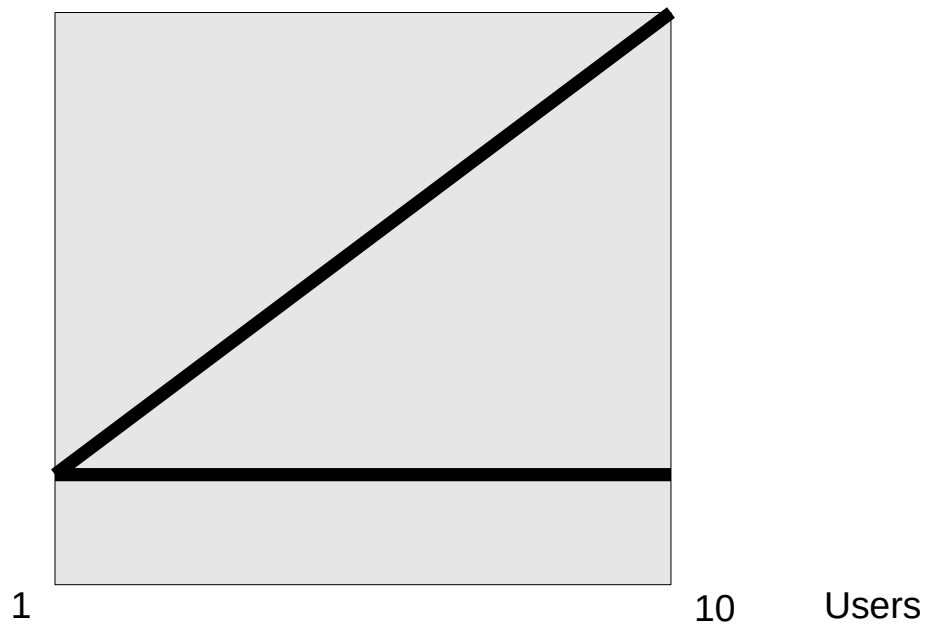


More about Vifib
www.vifib.org

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

10 Users, 10 ZEO Clients

Time
Per
Page



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

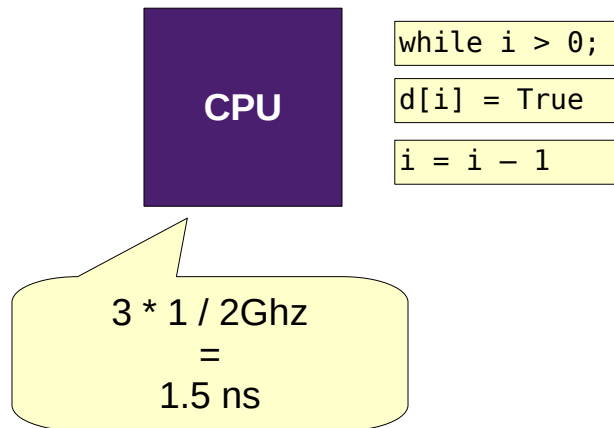
Why are Software Slow ?

```
d = {}  
i = 1e12 # 1 TB  
while i > 0; # at least one CPU instruction  
    d[i] = True # at least one memory access  
    i = i - 1 # at least one CPU instruction
```

How long does it take at least ?

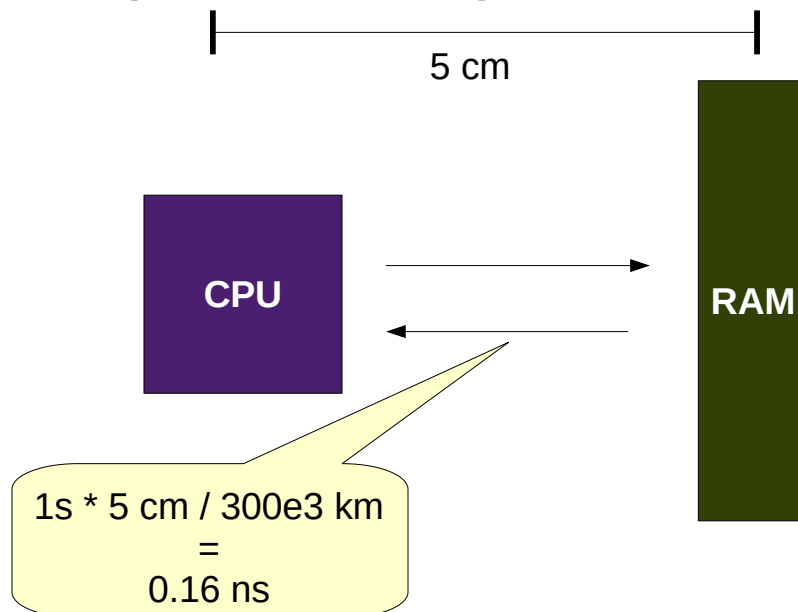
© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

The Frequency of CPU: 2Ghz



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

The Speed of Light: 300,000 km/s



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

The Frequency of RAM: 200 Mhz

`d[i] = True`

RAM

$$\begin{aligned} 1 / 200 \text{ Mhz} \\ = \\ 5 \text{ ns} \end{aligned}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

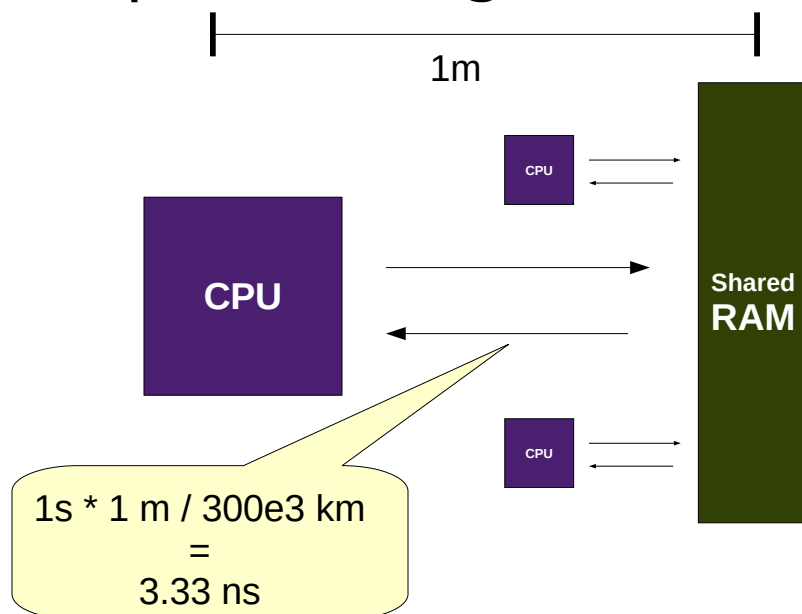
```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 5 + 0.16 * 2 + 0.5 = 5.82 ns  
    i = i - 1 # 0.5 ns  
# total = 6.82 ns
```

$1e12 * 6.82 \text{ e-9} = 6820 \text{ s} = \text{about 2 hours}$

$16e9 * 6.82 \text{ e-9} = 109 \text{ s} = 2 \text{ min}$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

The Speed of Light: 300,000 km/s



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

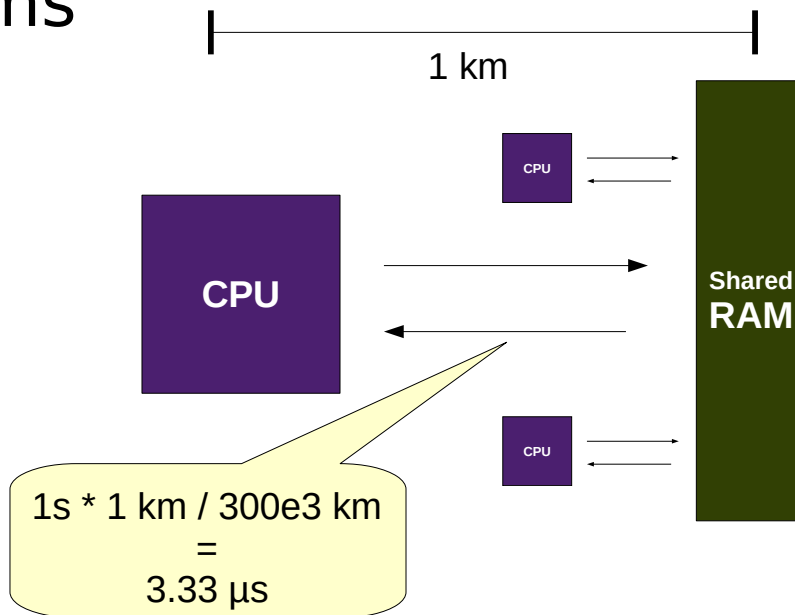
```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*3 + 6.66 = ns  
    i = i - 1 # 0.5 ns  
# total = 23.16 ns
```

$$1e12 * 23.16 = 23160 \text{ s} = 6.4 \text{ hours}$$

$$16e9 * 23.16 = 370 \text{ s} = 6 \text{ minutes}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

The Latency of Network: 1 μ s to 1ms



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
  d[i] = True # 0.5 + 5 + 3300*2 = ns  
  i = i - 1 # 0.5 ns = ns  
# total = 3316.5 ns
```

$$1e12 * 3316.5 = 38 \text{ days}$$

$$16e9 * 3316.5 = 14 \text{ hours}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

The Speed of Disks: 10,000 rpm

d[i] = True

Disk

$$\begin{aligned} 60s * 10e3 / 2 \\ = \\ 3 \text{ ms} \end{aligned}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

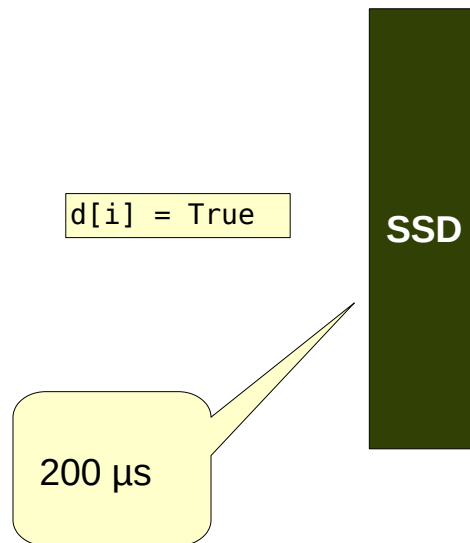
```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5 + 30000000 + 3.33*2 = ns  
    i = i - 1 # 0.5 ns + = ns  
# total = 3000023.16 ns
```

$$1e12 * 3000023.16 = 95 \text{ years}$$

$$16e9 * 3000023.16 = 1.5 \text{ year}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

The Latency of SSD: 200 μ s or better



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

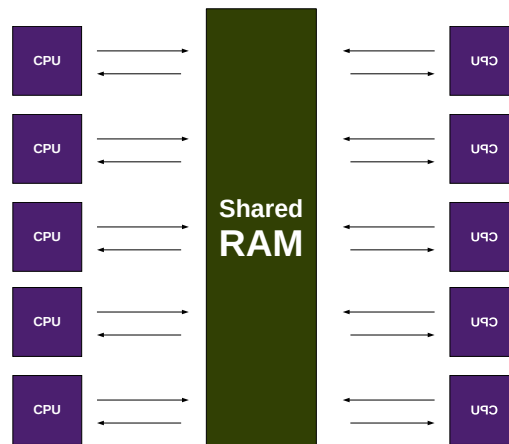
```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
  d[i] = True # 0.5 + 5 + 200000 + 3.33*2 = ns  
  i = i - 1 # 0.5 ns + = ns  
# total = 200023.16 ns
```

$1e12 * 200023.16 = 6 \text{ years}$

$16e9 * 200023.16 = 37 \text{ days}$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Access Conflict (Serialized)



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

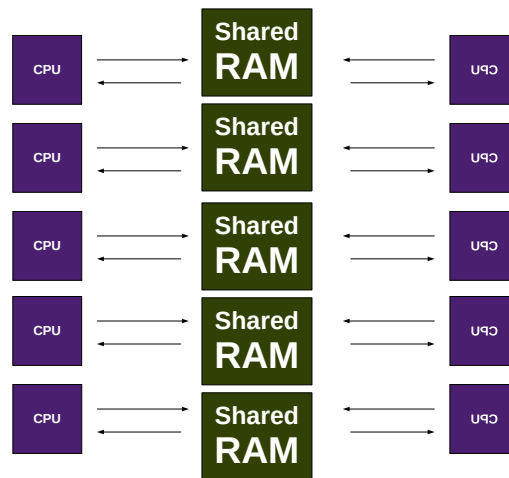
```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 66011.5 ns
```

$$1e12 * 66011.5 = 2 \text{ years}$$

$$16e9 * 66011.5 = 12 \text{ days}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Solving Access Conflict by Sharding



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

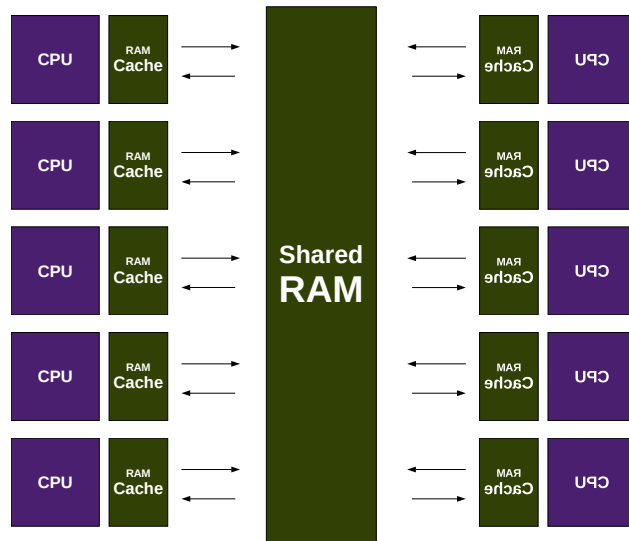
```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*2 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 13211.5 ns
```

$$1e12 * 13211.5 = 152 \text{ days}$$

$$16e9 * 13211.5 = 2.42 \text{ days}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Solving Access Conflict by Caching



© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Why are Software Slow ?

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 5 + 0.16 * 2 + 0.5 = 5.82 ns  
    i = i - 1 # 0.5 ns  
# total = 6.82 ns
```

$$1e12 * 6.82 \text{ e-9} = 6820 \text{ s} = \text{about 2 hours}$$

$$16e9 * 6.82 \text{ e-9} = 109 \text{ s} = 2 \text{ min}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Solving Access by Beautiful Code

```
d = {}  
i = 1e12 # 1TB or RAM  
while i-- > 0; # 0.5 ns  
    d[i] = True # 5 + 0.16 * 2 + 0.5 = 5.82 ns  
    # total = 6.32 ns
```

$$1e12 * 6.32 \text{ e-9} = 6320 \text{ s} = \text{about 2 hours}$$

$$16e9 * 6.32e-9 = 101 \text{ s} = 2 \text{ min}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Solving Access by Better Algorithm

```
d = {}  
i = 1e6 # 1TB or RAM  
while i > 0; # 0.5 ns  
  d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
  i = i - 1 # 0.5 ns = ns  
# total = 99010.5 ns
```

$$1e6 * 99010.5 = 1.65 \text{ minute}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Better Algorithm for 90% cases

```
d = {}  
i = 1e12 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 66011.5 ns
```

10%

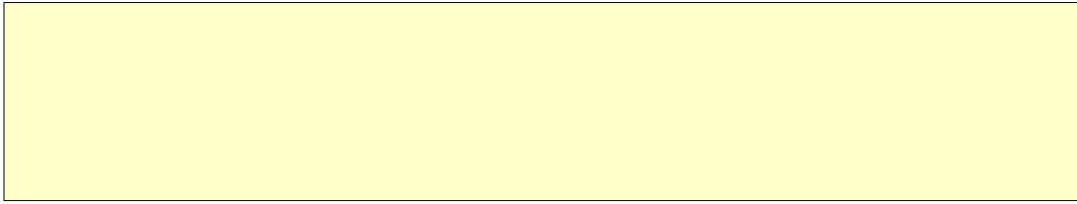
```
d = {}  
i = 1e6 # 1TB or RAM  
while i > 0; # 0.5 ns  
    d[i] = True # 0.5 + 5*2 + 3300*2*10 = ns  
    i = i - 1 # 0.5 ns = ns  
# total = 99010.5 ns
```

90%

$$(1.65 \text{ minutes} * 90 + 2 \text{ years} * 10) / 100 = 2.4 \text{ months}$$

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

Be Radical



0 s

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial

What About ERP5

- Postpone
 - Ajax Menus
 - Ajax Display
 - Activities
- Access Conflict
 - ID Generation
 - Access to Same Btree (HBTree)
- Caching
 - portal_caches
 - HTTP cache
- Sharding
 - Multiple Data.fs
 - NEO
- Algorithm
 - SQL Optimization
 - Catalog vs. ZODB

© Vifib SARL 2010 – All rights reserved – Creative Commons Shared Alike Non Commercial