Databases & Scalability

Dimitri KRAVTCHUK

Benchmark Team
Paris Sun Solution Center

Before we start...

- Few words about SSC :-)
- Paris <=== 10Mbit, 20ms latency ===> LLG

SSC Locations

- North America
  - USA: Hillsboro, Broomfield, McLean, Menlo Park
- Latin America
  - Sao Paulo, Brazil; Ft. Lauderdale, Florida; Mexico City, Mexico
- Europe
  - Edinburgh, Frankfurt, Madrid, Manchester, Milan, Munich, Paris, Walldorf
- Asia
  - Bangalore, India; Beijing, China; Hong Kong; Seoul, Korea; Singapore; Taïpei, Taiwan; Tokyo, Japan;
- Pacific
  - Sydney, Australia

Sun Solution Center Is Near You

ASSCs in BLUE
SSCs in BLACK

United States
- San Francisco Bay Area, CA
- Hillsboro, OR
- Broomfield, CO
- Mc Lean, VA
- Chicago, IL – Diamond Management
- Plano, TX - EDS
- College Park, MD - Univ of Maryland
- Pittsburgh, PA – Deloitte Consulting

Latin America
Ft. Lauderdale, FL, USA
Mexico City, Mexico
Sao Paulo, Brazil

Europe / Middle East / Africa
Edinburgh, Scotland, UK
Manchester, UK
Warrington, UK - Avnet
Paris, France
Frankfurt, Germany
Munich, Germany
Walldorf, Germany
Milan, Italy
Madrid, Spain
Götegorg, Sweden – Inserve Technology
Helsinki, Finland – ArrowECS
Tallin, Estonia - Microlink
UAE - Tech Access

Asia Pacific
Bangalore, India
Bangalore, India - Wipro
Beijing, China
Hong Kong, China
Shenyang, China - Neusoft
Seoul, Korea
Singapore
Singapore - Ingram Micro
Sydney, Australia
Sydney, Australia – Express Data
Tokyo, Japan

Sun Solution Center

Benchmark and Performance Characterization

- Architecture design
- High-end performance and scalability (servers, storage)
- Performance characterization
- Competitive benchmarks
- Internal product BU benchmarks
- Performance tuning
- Customer/Partner benchmarks
- Customer briefings

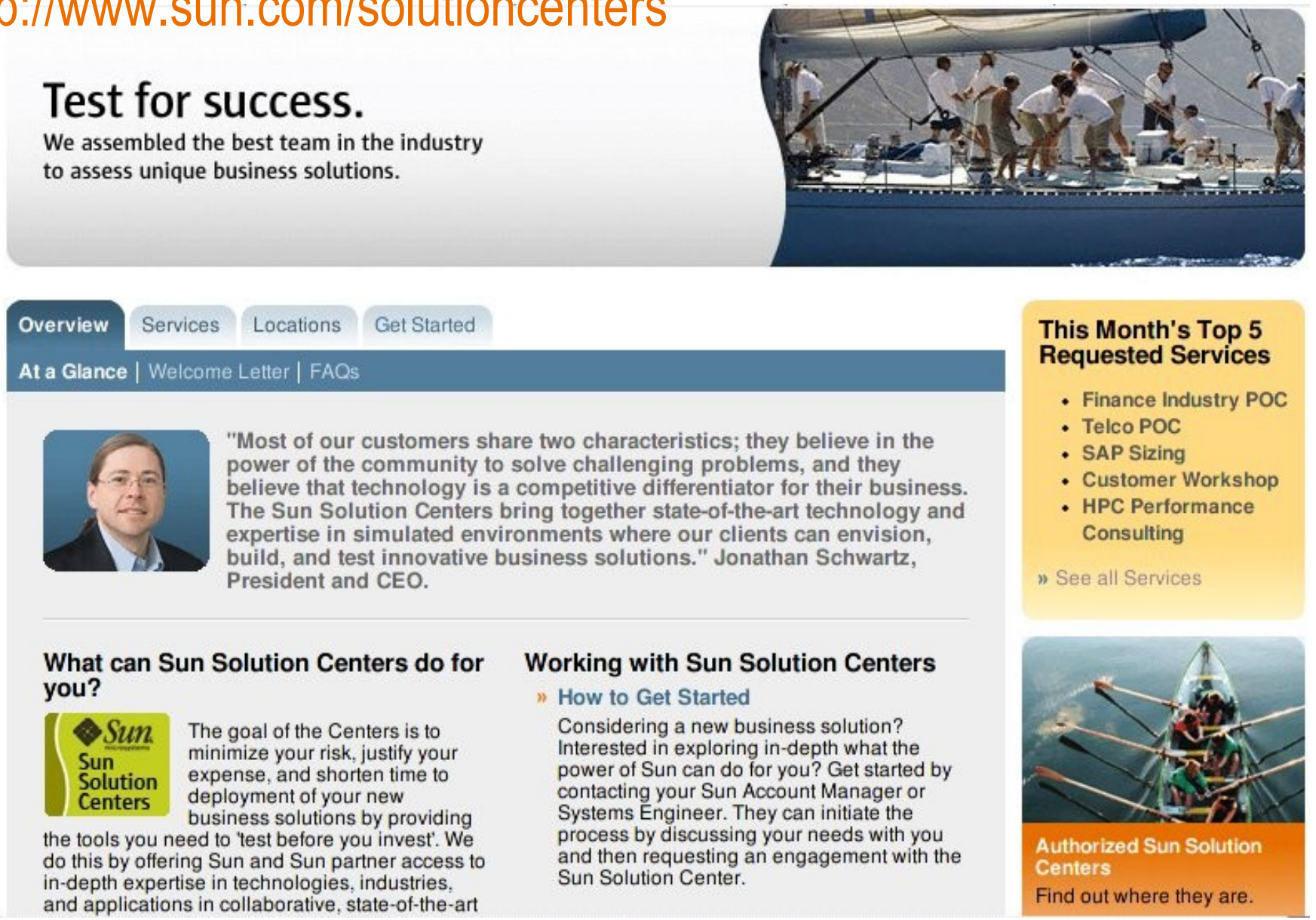
Sun Solution Center

Partner Solution Center

- Architecture design and validation
- Portfolio management and solutions offerings
- Customer/Partner Proof-of-Concepts
- End-to-end software development for live customers
- Industry solutions development and showcase
- Building of horizontal/biz solutions (eg: IdM, Security ... etc.)
- Business innovation and compliance (SOX, HIPA ... etc.)
- Demos, solution showcase

To know more

http://www.sun.com/solutioncenters



Agenda

- Why Scalability?...
- Designed to scale & Solaris performance
- Database Design Overview & Solaris
- Heavy Query: Paralleled or Smart execution
- MySQL Overview
- MySQL Storage Engines
- InnoDB Design
- InnoDB Performance
- Application & MySQL Tuning / Monitoring

Why Scalability?..

- Any answer?.. ; -)

Why Scalability?..

- Any solution may be accepted as just “good enough”...

Why Scalability?..

- Until it did not reach its limit... ; -)

Why Scalability?..

- And even improved solution may be overloaded with a time... ; -)

Why Scalability?..

- And meet the same limit... ; -)

Why Scalability?..

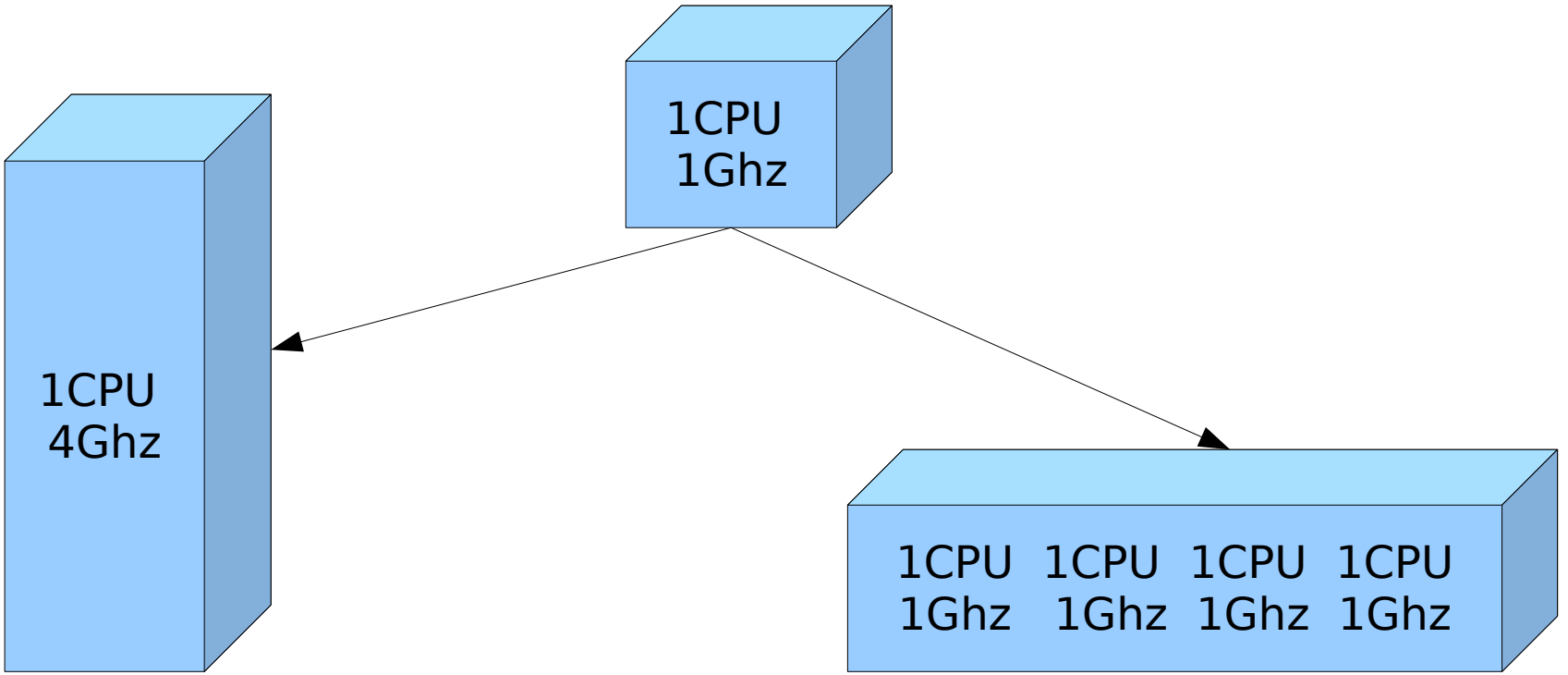
- Eternal goal: “auto”-adaptive to load solution...

Why Scalability?..

- And keep in mind: even very powerful solution but in wrong hands may be easily broken!... : -)

Back to Computers...

- Your application is running well, but you have to increase performance at least by x3 – to which plateform will you move?..



Evolution in development

- 1980:
  - Here is our computer
  - Now let's develop an application which will run well on it!...
- 2000:
  - Here is our application...
  - Let's now try to find on which computer it'll run well!...

“Free lunches are finished!...”

- Article was written by a non-Sun employee!
- How fast was increased CPU frequency during last years?..
- How fast was increased CPU number on a single server?..
- Sun SPARC servers:
  - M9000-32: 256 cores (512 hardware threads), > 2TB RAM
  - T-series: 1CPU= 8cores (64 hardware threads)
    - 1 to 4CPU within a single unit!
- Intel / AMD chips are following the same way
- Parallel processing => is The Answer

How easy is Parallel Processing?

- Be honest – it's hard!
- But on the same time it's one of the main reasons why we still need engineers! - So, be happy! :-))
- Do you mean any processing may be “Paralleled” ?..
  - Probably not all..
  - But rather most of them :-)

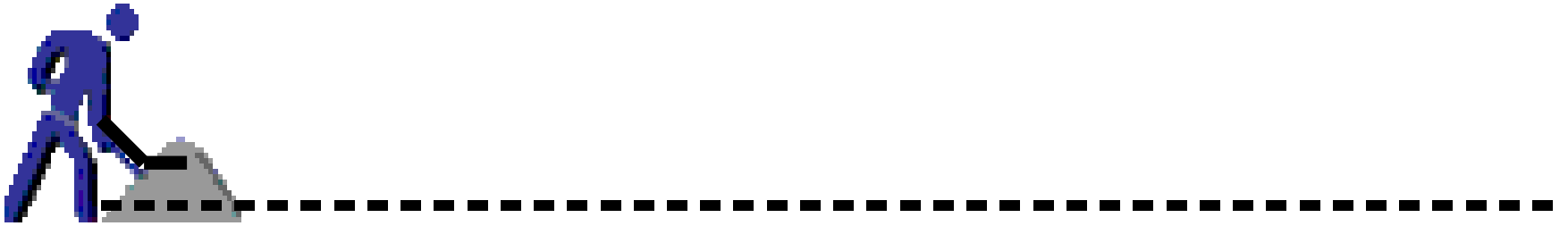
By Stupid Example...

By Stupid Example...

- 1. Adapted size...

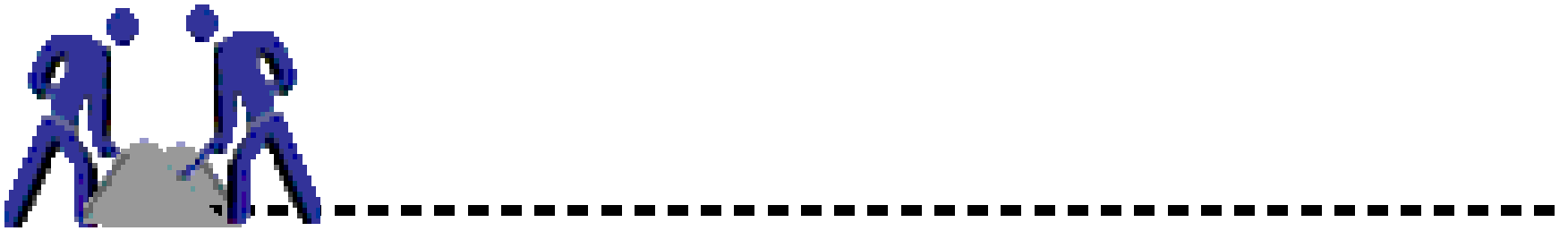
By Stupid Example...

- 1. Adapted size...



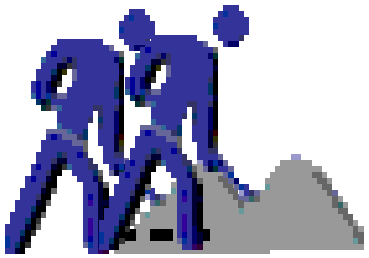
By Stupid Example...

- 1. Adapted size...
- 2. “Paralleled”



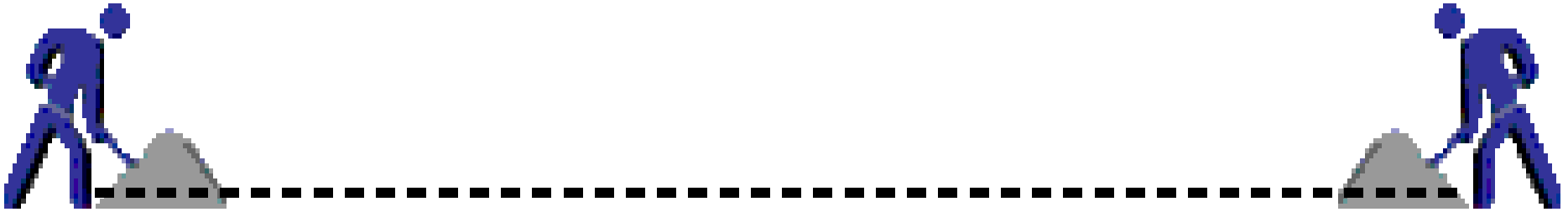
By Stupid Example...

- 1. Adapted size...
- 2. “Paralleled”



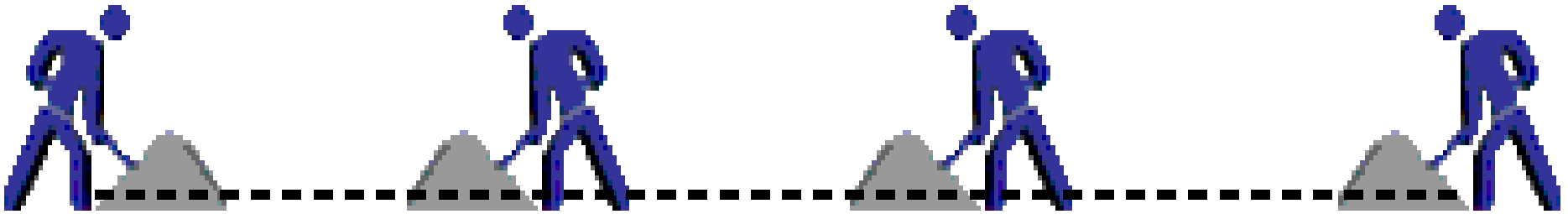
By Stupid Example...

- 1. Adapted size...
- 2. “Paralleled”



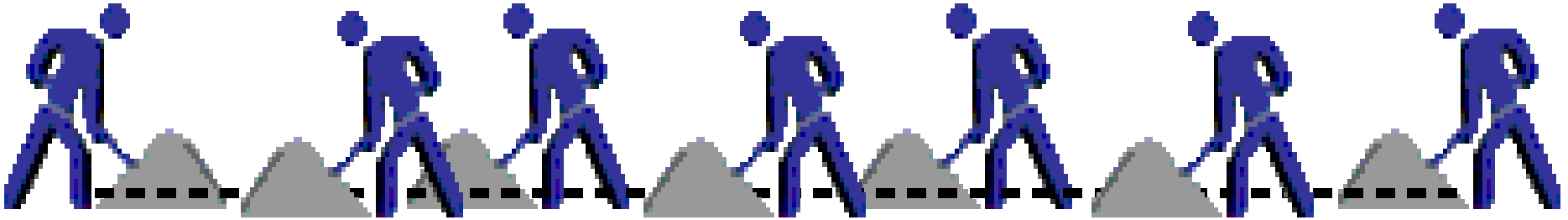
By Stupid Example...

- 1. Adapted size...
- 2. “Paralleled”
- 3. “Parallelization” limits...



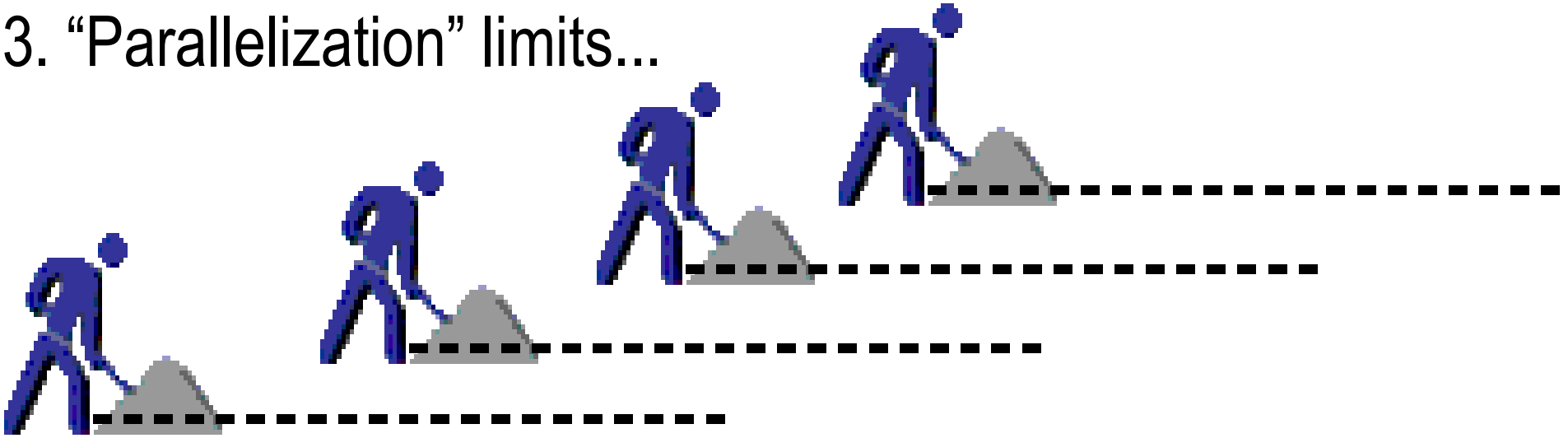
By Stupid Example...

- 1. Adapted size...
- 2. “Paralleled”
- 3. “Parallelization” limits...



By Stupid Example...

- 1. Adapted size...
- 2. “Paralleled”
- 3. “Parallelization” limits...

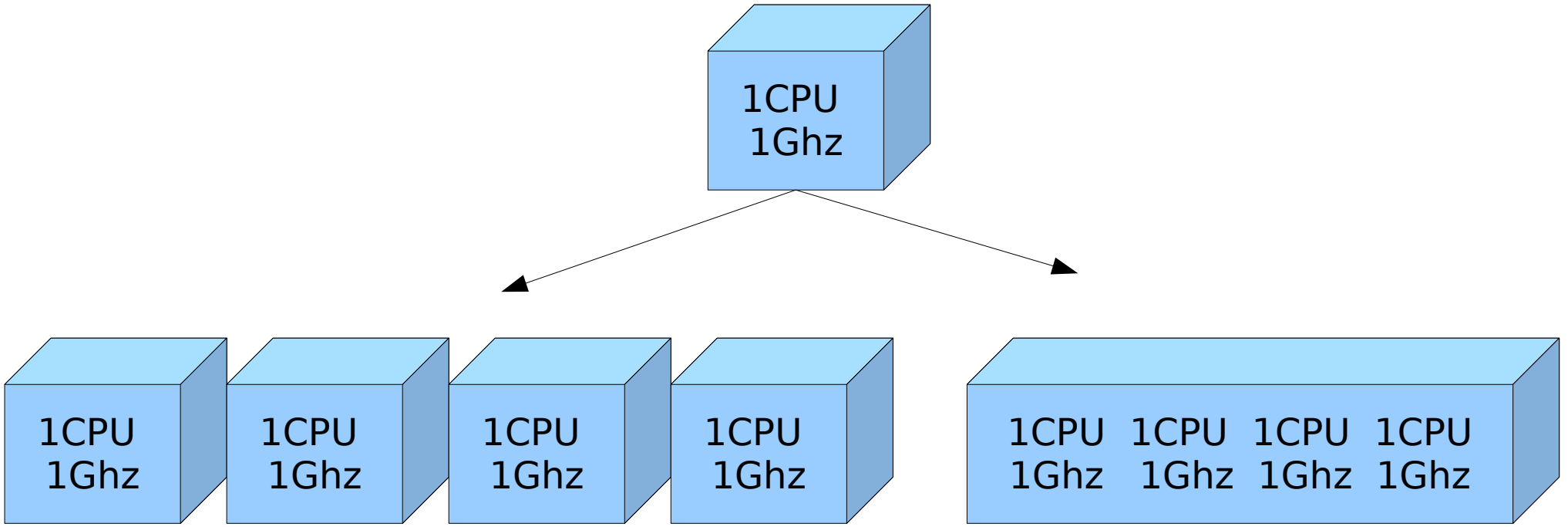


Overhead & Scalability

- Server with 256 CPU
- A given Application supposed to scale
- Amdahl's law:
  - Overhead = 0.1% => 200 CPU
  - Overhead = 10% => 70 CPU (!)
- Reducing Overhead => Improving Scalability!
- Code Instrumentation
  - => Most optimal way to understand Overhead

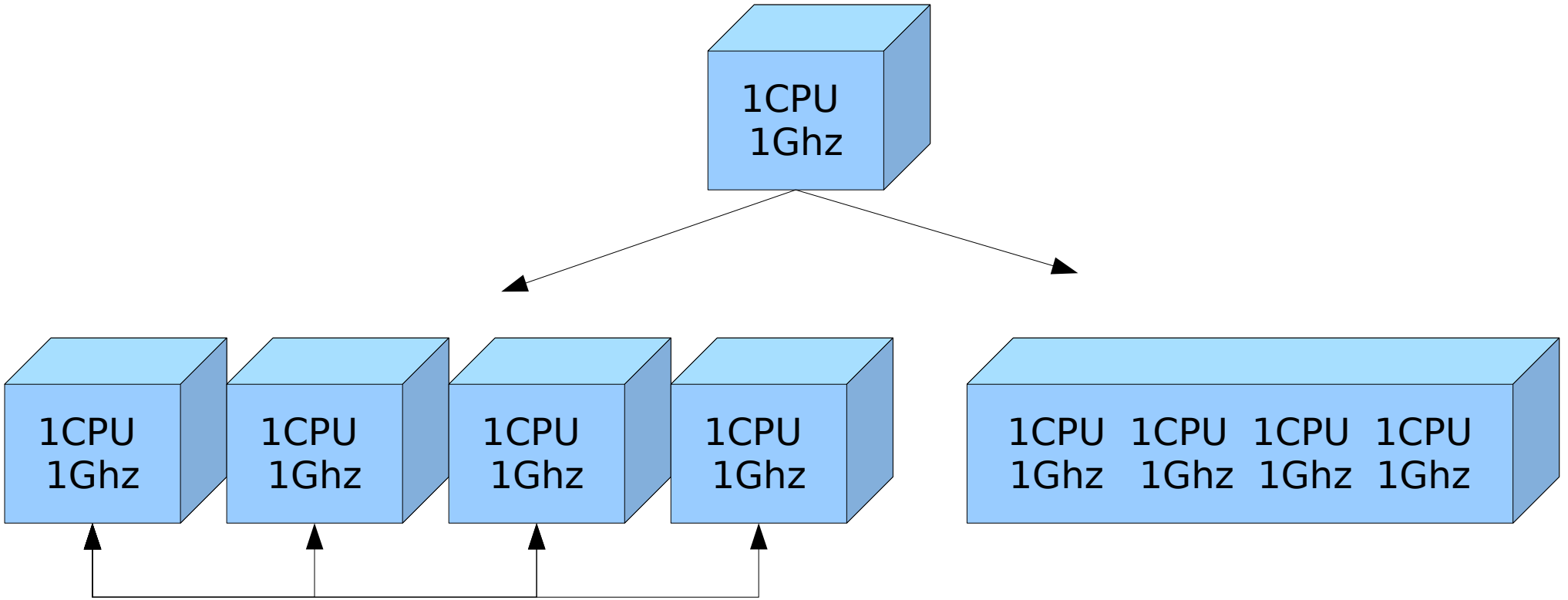
Scalability: Vertical or Horizontal

- Main Problem = Serialization / Contention...
- Vertical or Horizontal?..



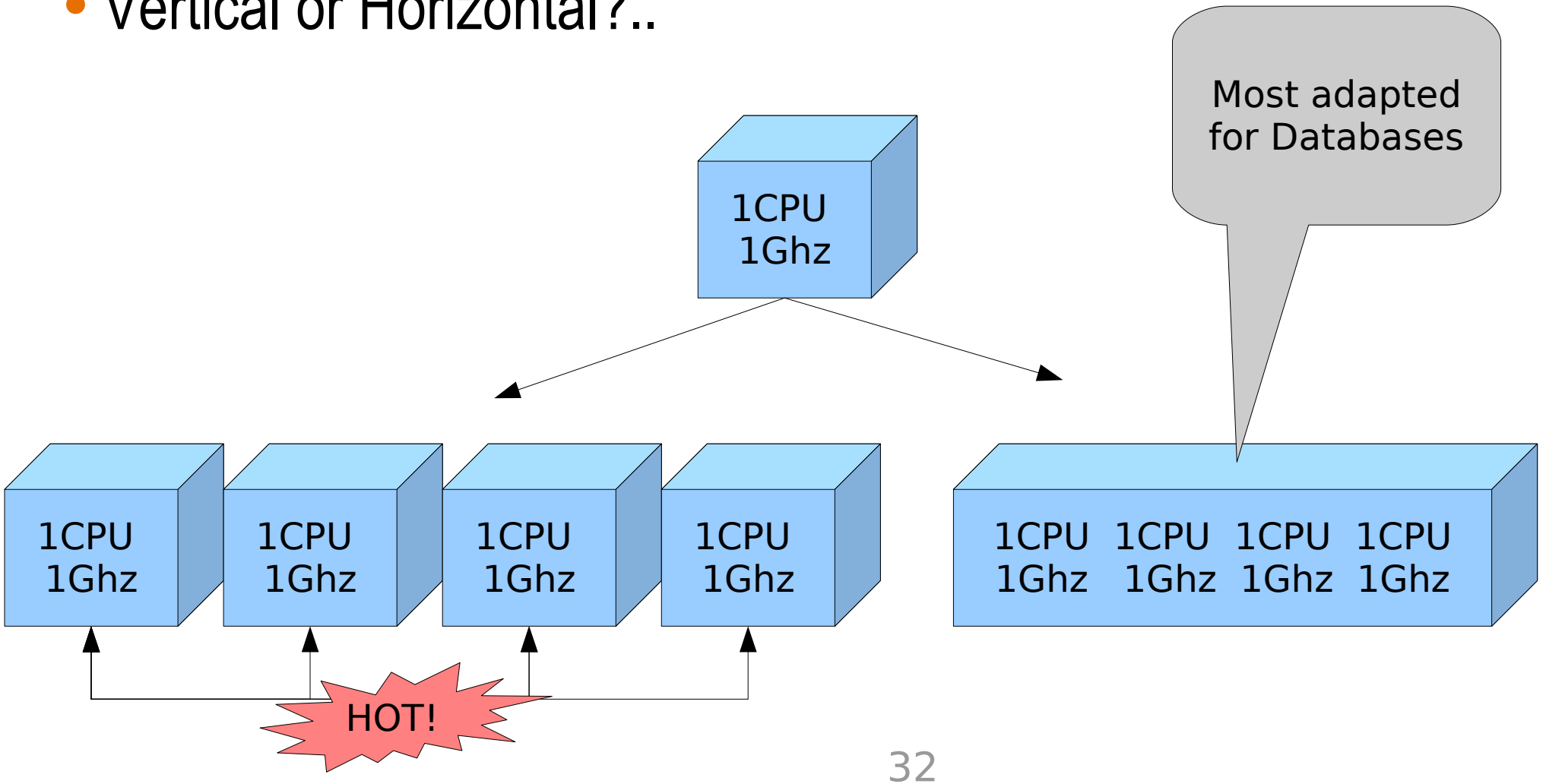
Scalability: Vertical or Horizontal

- Main Problem = Serialization / Contention...
- Vertical or Horizontal?..



Scalability: Vertical or Horizontal

- Main Problem = Serialization / Contention...
- Vertical or Horizontal?..

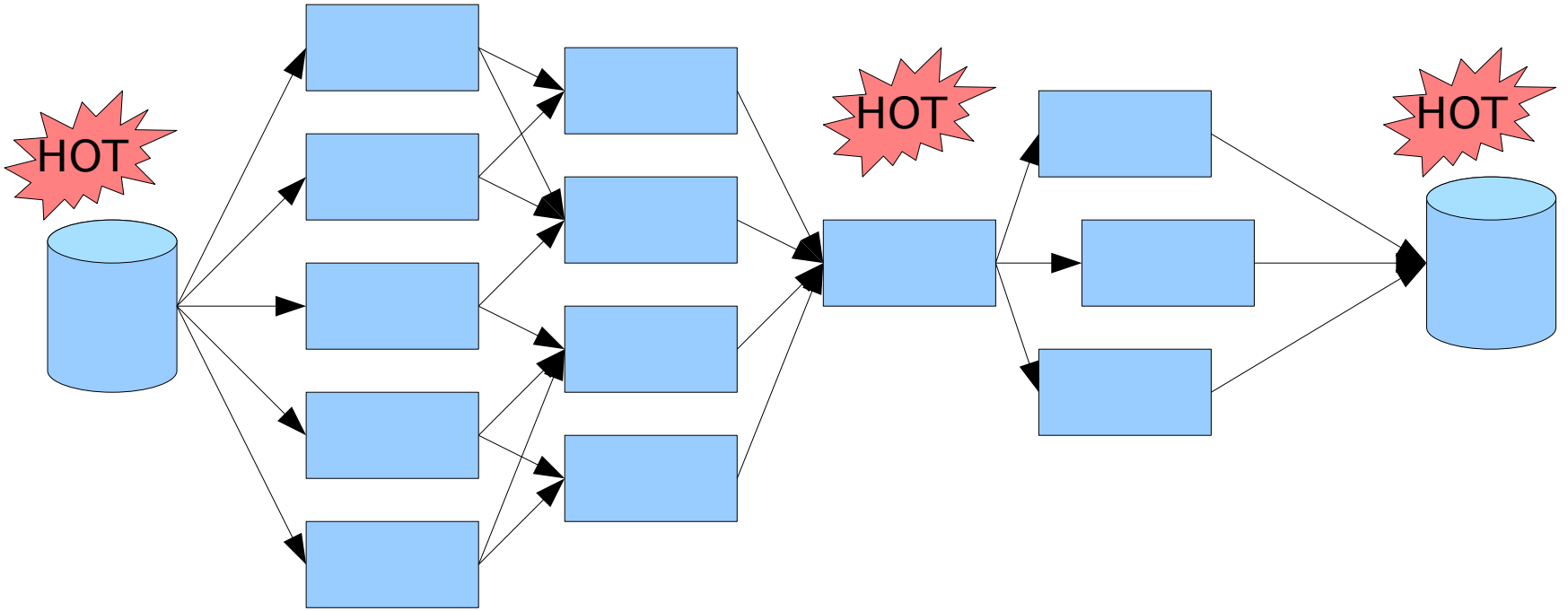


Parallel & Parallel :-)

- You have to process 2 tasks on 1CPU:
  - Each uses 100% CPU
  - Each takes 5 min if runs alone
- Executing 2 tasks in parallel:
  - Total time: 10min
  - Task1 time: 10min
  - Task2 time: 10min
- Executing 2 tasks sequentially:
  - Total time: 10min
  - Task1 time: **5min**
  - Task2 time: 10min

Processing Model Design

- Avoid bottlenecks since your Model Design!

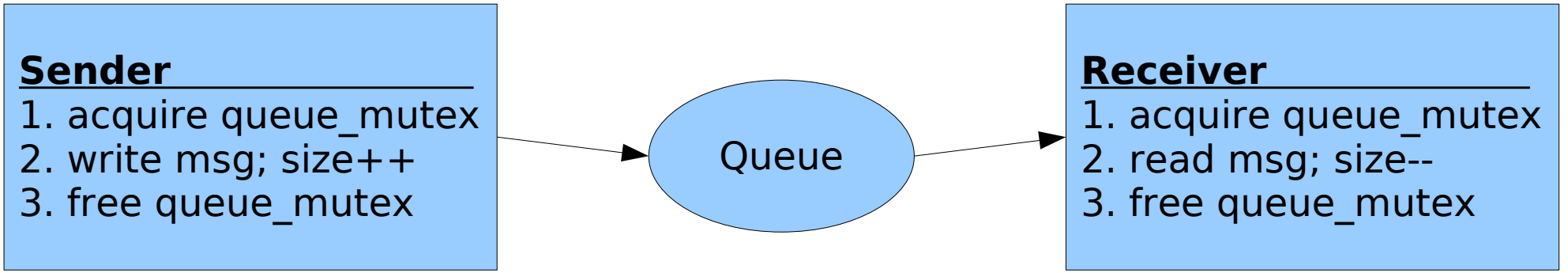


Processing Model Implementation

- Multi-process:
  - Context switch may cost
  - SHM, SEM, MSG
- Multi-thread:
  - Low cost context switch
  - All data seen by all threads
  - Mutex / Atomic operations to manage concurrent access
- Eternal main bottleneck: Locks!

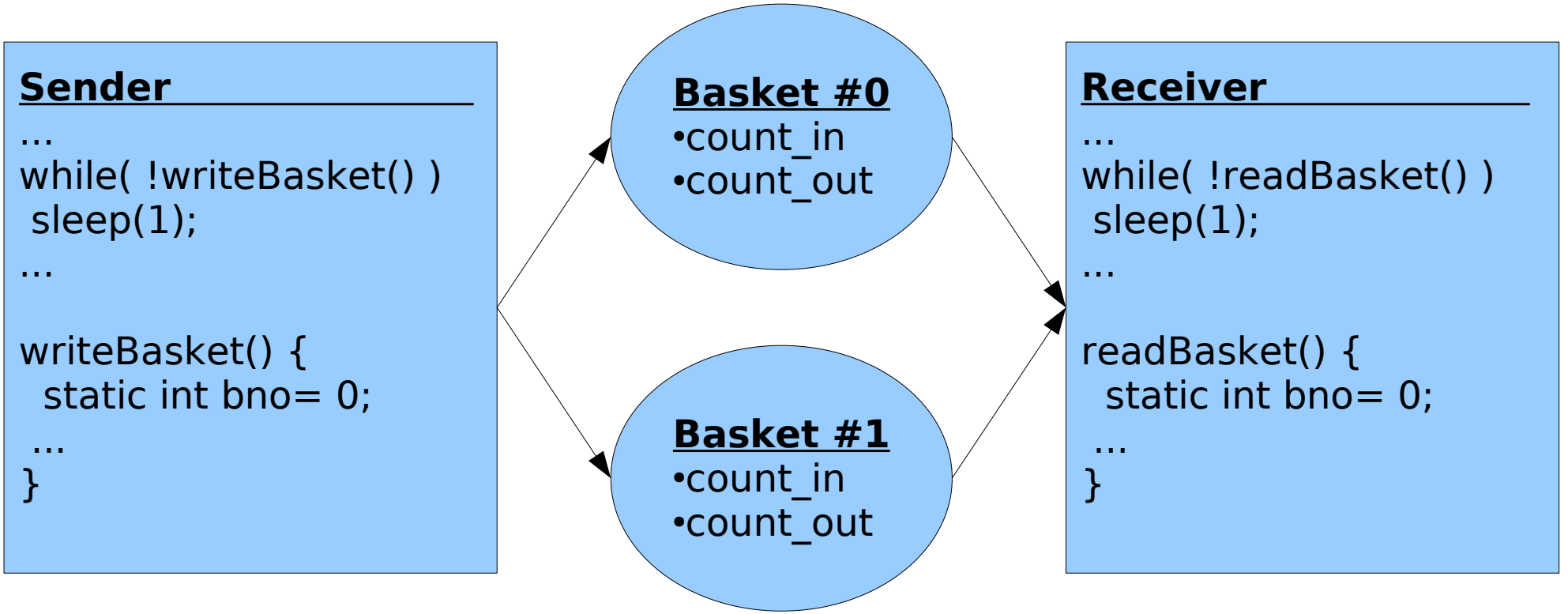
Example: Queue Management

- Classic error: single mutex protected
  - > More processing become fast => More contention become high
  - > Spin locks feature



Example: Queue Management

- Improving Performance: Double-basket queue
  - > Free of locks



Choice of Operating System

- Solaris
  - CPU scalability: proven to scale over 256CPU
  - I/O Level: no limitation
  - Network: throughput is ok, latency need to be improved
  - DTrace!
- Linux
  - CPU scalability: need to prove yet...
  - I/O Level: limited or very limited..
  - Network: throughput is ok, latency is ok
- AIX, HP/UX, FreeBSD, etc..

Choice of Platform

- Intel Server
  - Very fast on CPU, may not scale, but constantly improved
  - May be limited on I/O
  - But don't forget – it's just a big PC !
- SPARC Server
  - Fast enough on CPU, scales very well
  - I/O level is great
  - Very secure, H/W redundancy, Dynamic reconfiguration, etc.
- other..

Main bottlenecks

- 95% => Application itself!
- Network
  - > Packets (latency) vs Throughput (MB/s), interrupt mode
- I/O level
  - > Operations/sec vs Throughput(MB/s), I/O nature
- Locks
  - > Mutex, atomic operation, RW-lock, spin
- Memory management
  - > NUMA, TLB-miss, ISM, DISM
- Communication
  - > Sockets, SHM, MSG, pipes

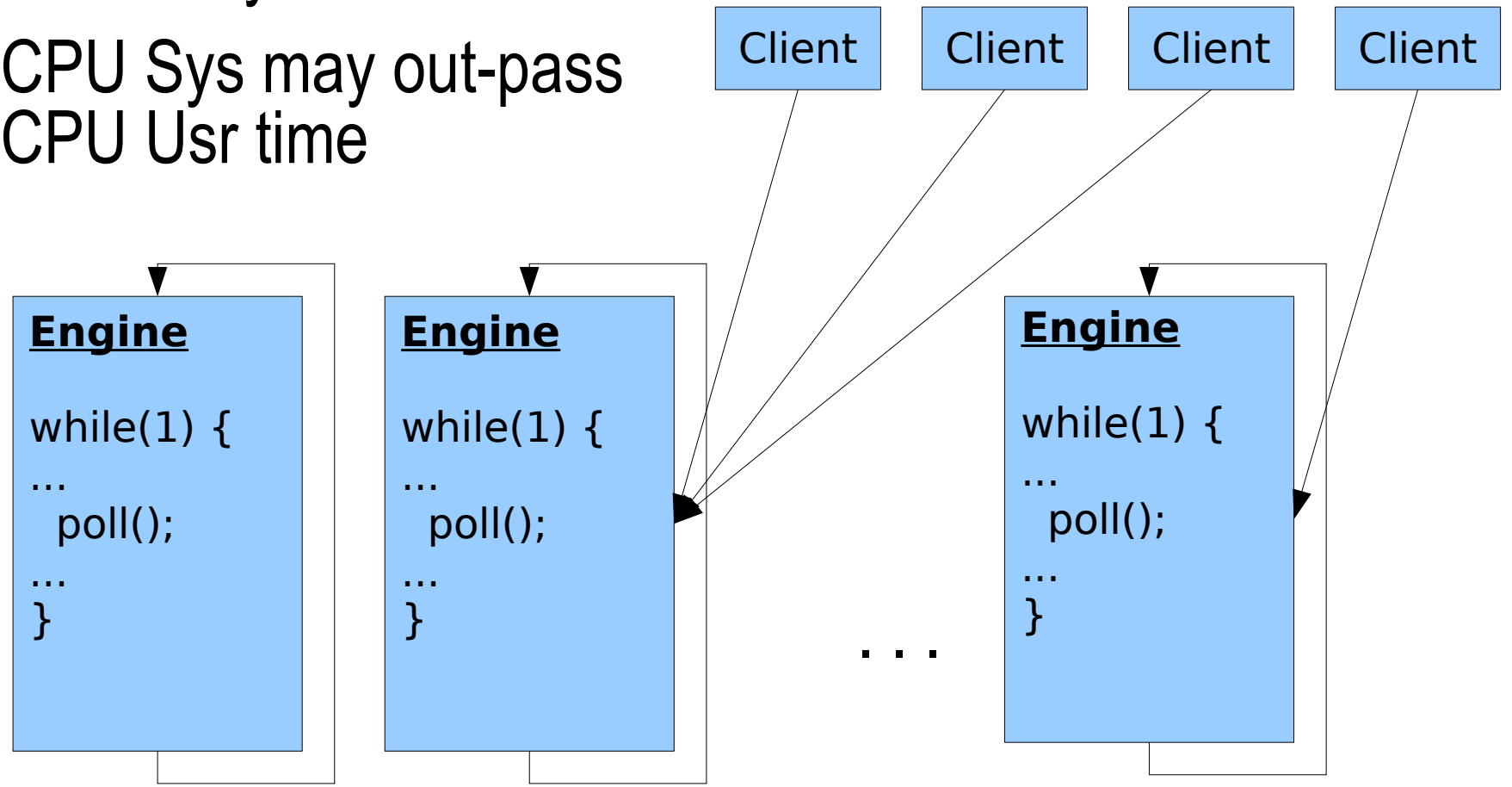
Databases

- Do I have an idea?.. I have so many ideas!!!!!

- Well... Even if the initial idea is good
  - > Let's see the implementation! :-))

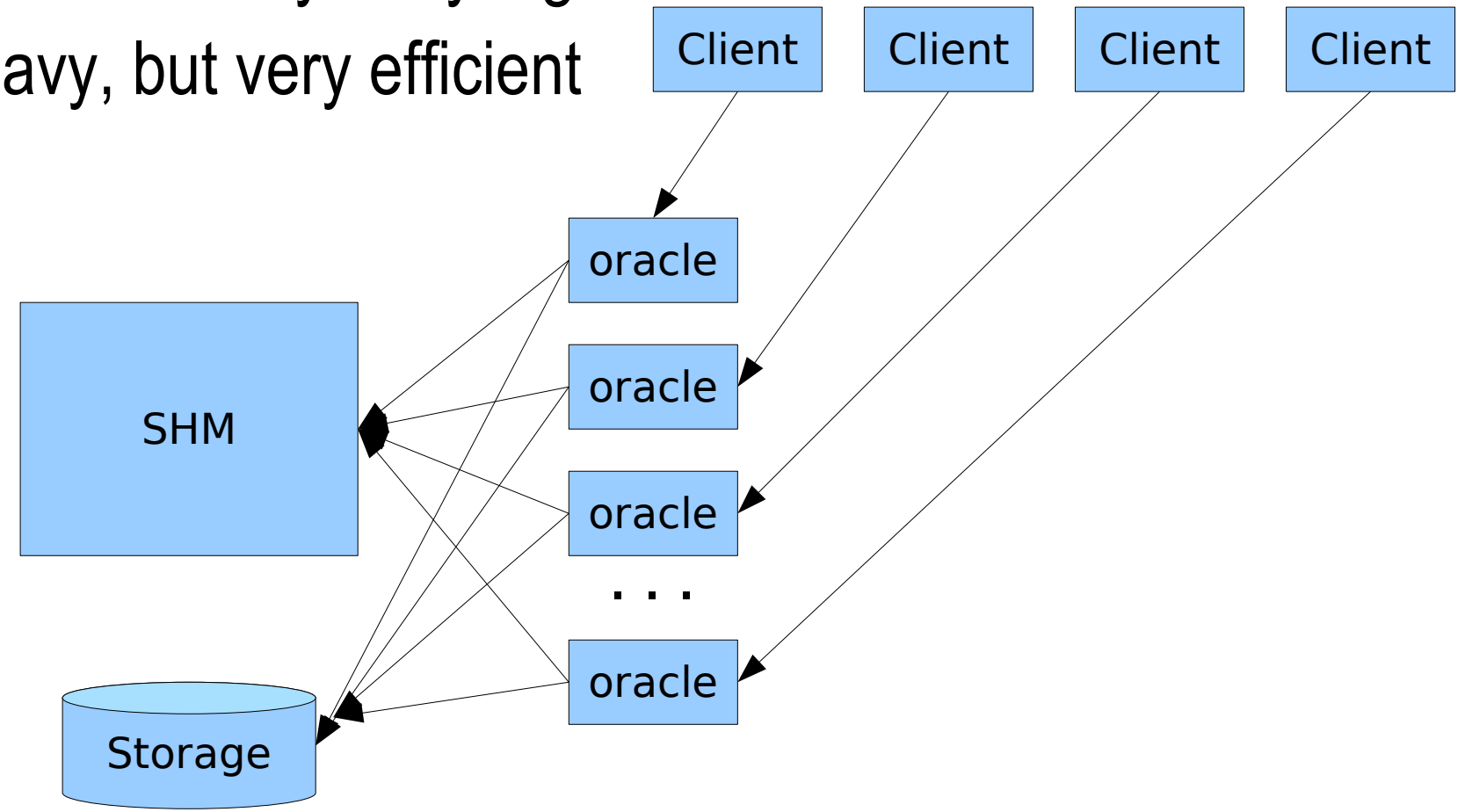
DB Implementations: Sybase

- Bound on poll()
- Scalability issues: ~16CPU
- CPU Sys may out-pass CPU Usr time



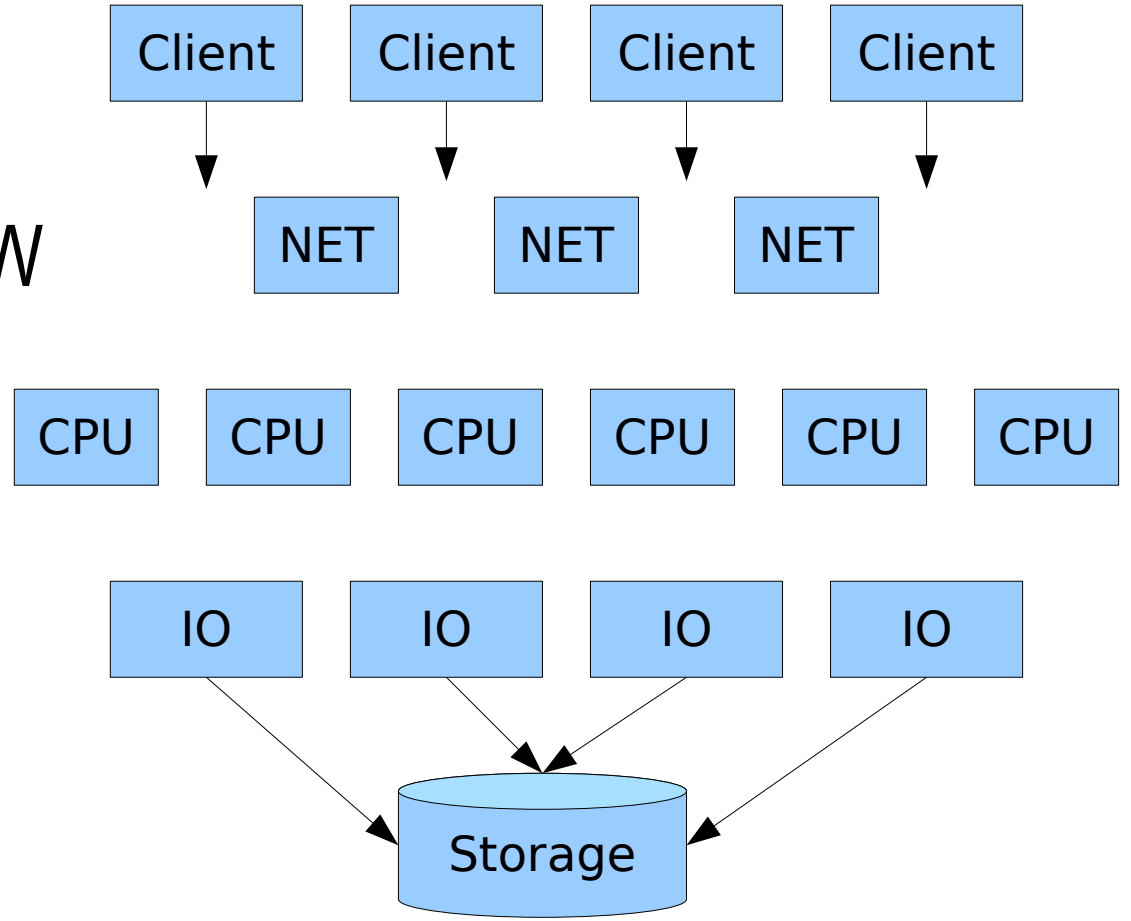
DB Implementations: Oracle

- Dedicated server process for each client
- Max scalability: very high!
- Heavy, but very efficient



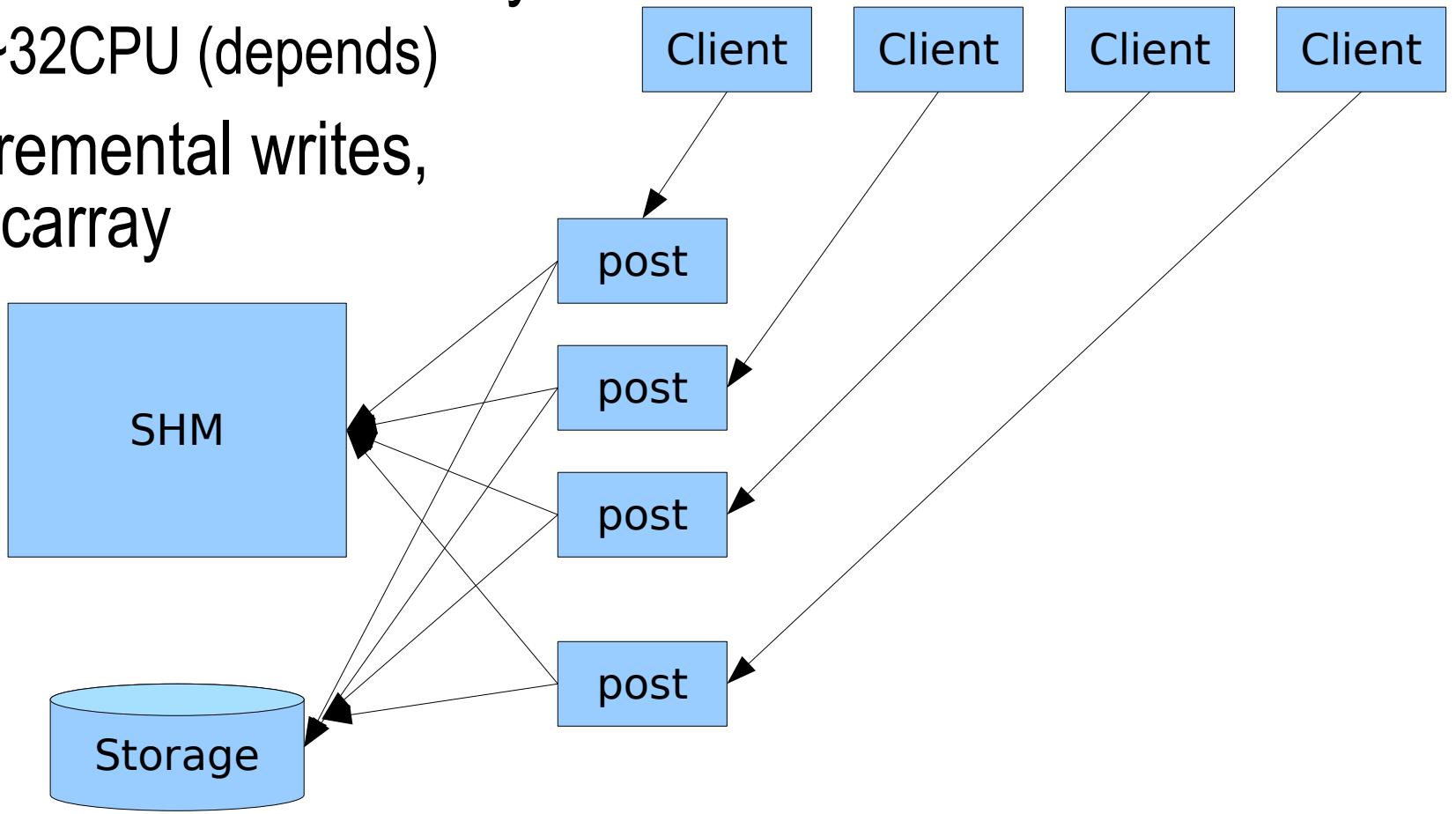
DB Implementations: Informix

- Pool of threads (configurable):
  - > Net, CPU, IO
- Max scalability:
  - > very high!
- Most optimal use of H/W



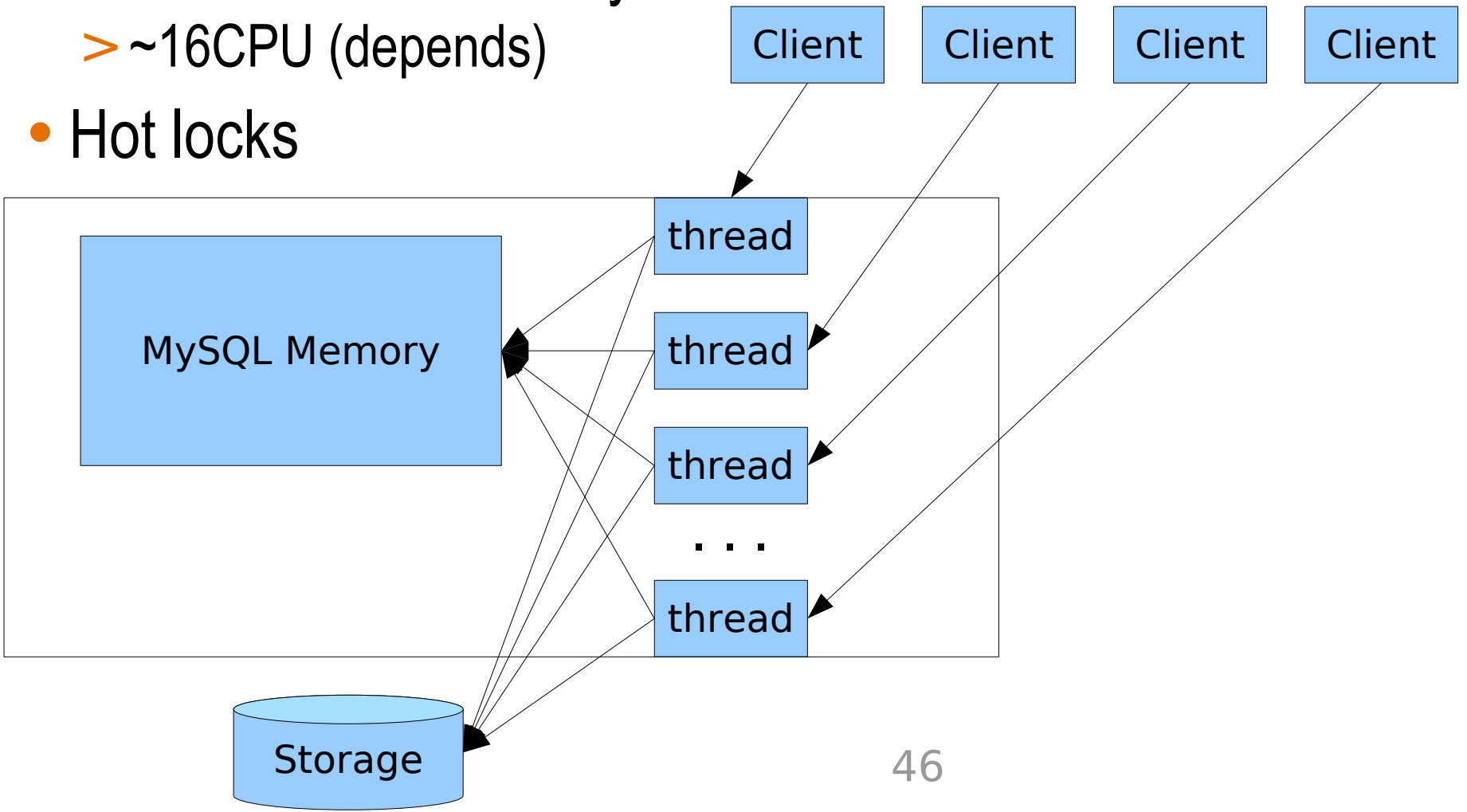
DB Implementations: PostgreSQL

- Dedicated server process for each client
- Current max scalability:
  - > ~32CPU (depends)
- Incremental writes, procarray



DB Implementations: MySQL

- Dedicated server thread for each client
- Current max scalability:
  - > ~16CPU (depends)
- Hot locks



Heavy Query

- Paralleled Execution:
  - Oracle
  - Informix, Informix XPS
  - Terradata
  - Greenplum(PostgreSQL)
- Smart Execution:
  - Sybase IQ
  - Infobright (MySQL)

Scalable DB Application

- Right platform + OS + Database vendor
- Scalable data model
  - Table lock, page lock, row lock, serial/sequence, etc.
- Scalable code
  - Paralleled
  - Free of locks ;-)
    - Note: CPU cache & data arrays
  - Efficient
- Just do it! :-)

Q & A

=> next slides...