



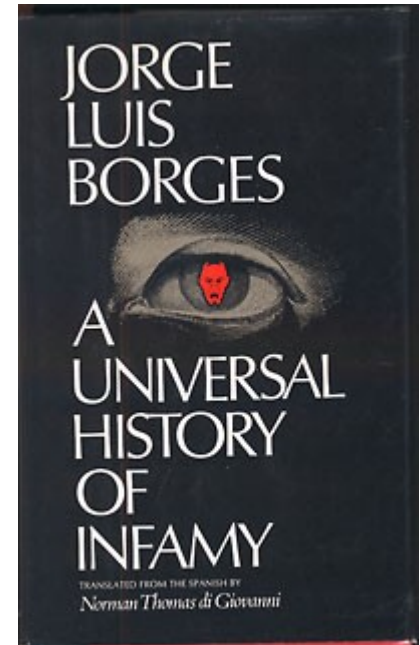
Subject: **Goodle Scales**

From: Martín Langhoff <[martin@catalyst.net.nz](mailto:martin@catalyst.net.nz)>

To: MoodleMoot UK 2007 <[all@moodlemoot.ex](mailto:all@moodlemoot.ex)>

# A universal history of scalability

- 1.3/1.4
- 1.4.3 to 1.6
- 1.7 roles impact
- 1.8 some enhancements
- 1.9 accesslib rewrite
- We are not alone



# Costs

DB > disk > memory

- Work in DB - orders of magnitude cheaper
  - operations (compare, add, etc)
  - data transfer
- Ambiguous ops in PHP
- Latency vs processing time
- Resource locking and contention

# Efficient algorithms

Key step in the design of each module, block, etc.

- The difference is *staggering*.

# Proven scalability



Greater complexity only  
guarantees complexity.

If you cannot prove that the  
new code is significantly  
faster/lighter, under a  
variety of workloads...  
*it isn't*

# Examples

- Data-driven loops
  - Query outside the controlling loop fetches the necessary data
  - Process them within the loop
  - No DB queries within the loop!
- Functions in libraries/classes must declare whether they make any queries. Ideally, they should work on parameters instead.

# Other costs

- Minimise include()/require()d php files
- require libs JIT
- JIT objects



# Caches

## Last resort

- High setup costs
- HTTP is non-persistent
- Hard to design data structures
- No guarantees of locality
- Synchronisation costs in clusters
- `serialize()/unserialize()` costs

# Moore's Law

- ... and lazy developers
- Moodle admin expects new features, and more users/courses/traffic!
- Orders of magnitude between efficient and inefficient
- Impossible vs possible

# Why Google?

- Scales thought impossible
- Simple tools, intellectual discipline
- Publish papers *y some* code
- Any sufficiently advanced technology...

# Divide & Conquer

- Split and parallelise tasks across the data-space -- Map/Reduce
- Split the dataspace according to data locality -- GFS
- Coarse grained, vertical data segments -- BigTable

# Goodle?

- Long road to a pure Google-style design
- Moodle is course and user-centric
- Affinity webserver -> course
- Cache high locality data segments using HD & shmem
- Example: mod/chat

# Summary



- History
- Costs
- Efficient algorithms
- Prove it
- Examples - loops!
- Other costs
- Caches?
- Moore's Law
- Why Google
- Divide & conquer
- Goodle?