How to design and build a scalable registry

Eduardo Sztokbant – Registro.br
eduardo@registro.br

ccTLD Workshop – Bangkok, Thailand
October 10th, 2004

How to design and build a scalable registry
TOC

- Design for read load
- Design for extensibility
- Exhaustive testing
- Third-party software security

How to design and build a scalable registry
Design for read load

- Most of the queries to be run are read queries (whois).

- Bottleneck isn't the CPU, but hard disk overloading.

- At Registro.br we have 2 main database boxes, a master one and another one basically to answer whois queries.

How to design and build a scalable registry
Design for read load – optimized queries

- Optimized queries chosing good indexes.

- A deep study of the database will be really valuable to know when a full-table scan should be replaced by an index.

- Indexes should be integer whenever possible.

How to design and build a scalable registry
Design for read load – queues scale well

- Registro.br's model is based on a first-come-first-served ticket queue processed by a daemon periodically.

- Average of 1,000 new domain registrations/day.

- System is robust enough to deal with up to 50,000/day.

- queues+daemons scale well.

How to design and build a scalable registry
Design for extensibility

- The system must be flexible enough to have new features incorporated without the necessity to change the design.

- A good object-oriented design is worthy of consideration.

- Registro.br's software is written in C++ and also makes extensive use of C++ frameworks.

How to design and build a scalable registry
Exhaustive testing

- One of the most important tasks for achieving stability and preventing unpleasant surprises.

- Every single new piece of software should be as tested as possible.
e.g.: from simple new features added to the code to third-party application upgrades.

How to design and build a scalable registry

Exhaustive testing

At Registro.br:

- 2 independent development environments to write/test code.

- MySQL had been tested for about 1 year before Solid was discarded.

- Registro.br's system is still being tested over FreeBSD 5 and GCC 3. Production boxes run FreeBSD 4 and code build with GCC 2.95.

- CppUnit is being adapted with the intent of testing the behaviour of the code after new code is written.

How to design and build a scalable registry
Third-party software security

- Security patches for software being run should be applied as soon as they are available.

- It is recommended to daily monitor specific mailing lists/websites such as bugtrack.com

Questions??

?

Eduardo Sztokbant - Registro.br
eduardo@registro.br