



Tema: **La Scala di Moodle**
De: Martín Langhoff <martin@laptop.org>
A: MoodleMoot CO 2010

Historia universal de la escalabilidad



- Elecciones en NZ
- 1.4.3 a 1.6
- 1.9 reimplementación de roles
- Trabajo con grandes instalaciones (NZ, UK, Australia, España)
- No estamos solos

Entender y Medir

- Protocolo HTTP
 - Usuarios simultáneos vs Hits/Pageviews
- Foco: Tajadas 5s
- Patrones de uso y carga
 - 100 usuarios en 5s
 - proyectar o medir



Proyectando

- Login
- Homepage
- Página curso
- Módulo
- 'Hub' distribuidor
- Modelar en base a pruebas

Medir en producción

- Perfstats (al log!)
- “Top Ten” de la semana!
- Usuarios distintos por hora
- Picos de pageviews / hits

Configuraciones

- Servidor único
- Cluster
- VMs

Servidor Unico

- Memoria > Discos > CPU
- *Asignar* RAM: 30% Pg, 60% Apache/PHP
- Kernel / OS
 - Sin GUI
 - Sysstat
 - Parametros sysctl (vm.dirty_*, kernel.shmmax)
 - Formato y montaje de discos (journaling, noatime)

Servidor Unico

- Apache
 - MaxRequestsPerChild
 - MaxClients
 - KeepAliveTimeout
 - Memoria: ps_mem.py, top

Servidor Unico

- PostgreSQL
 - Unix Socket!
 - shmem
 - Configurar para más memoria
 - maxconnections = Apache MaxClients
(solo si hay una sola DB y usuario)
 - Disco txlog (sin journalling)
 - Porqué Pg?

ONE LAPTOP PER CHILD

1



This work is licenced under a Creative Commons Attribution 2.5 Licence.

Ejemplos de costos y algoritmos

- Loops controlado por los datos de un query
 - Query afuera del loop obtiene los datos necesarios
 - El loop los procesa
 - El loop no hace ningun query!
- Funciones en librerias tienen que declarar si hacen queries -- idealmente, deben actuar sobre los parametros

Algoritmos y costos

- Minimizar los archivos cargados / interpretados
- librerías requeridas "justo a tiempo"
- clases instanciadas "justo a tiempo"



Caches

Ultimo recurso

- Altos costos
- HTTP es no-persistente
- Diseñar estructuras de datos
- Pocas garantias de localidad
- Costos de sincr en clusters
- Costos en serialize/unserialize

La ley de Moore

- programador: quiere ser mas perezoso
- dueño del servidor quiere mas prestaciones Y mas usuarios/cursos/trafico!
- La diferencia entre eficiente y no eficiente esta en magnitudes
- Puedes hacer cosas imposibles

Porque Google?

- Hacen lo previamente pensado imposible
- Herramientas simples + disciplina intelectual
- Publican papers y código
- Revolucionario

Divide y conquista

- Divide en tareas que dividen el espacio de datos y son paralelizables
- Segmentar el espacio, aumentar "localidad" de los datos
- Bloques grandes de datos - corte vertical
- Map/Reduce
- BigTable
- GFS

Goodle?

- Largo camino al modelo puro de Google, diseño puramente distribuido
- Moodle esta centrado en el curso y el usuario
- Afinidad webserver -> curso
- Cache local fuerte -- HD, shmem
- ejemplo: mod/chat

Resumiendo



- Ojectivos del diseño
- Probarlo
- Ejemplos de algoritmos y costos
- Algoritmos y costos
- Caches?
- La ley de Moore
- Porqué Google
- Divide y conquista
- Goodle