

# **Scaling Zope for ASPN**

Andy McKay, ActiveState

# Agenda



- Specifications
- Building it
- Scaling it
- Lessons learned

# Specifications



- ActiveState Programmers Network (ASPN)
  - Modeled on MSDN
  - ASPN is a subscription model product that provides a bundle of software and services
  - Free level ASPN Open
- ASPN membership included with products
- Data restricted by level

# Features



- First release:
  - Mailing list archives, Cookbook's (Python Cookbook etc)
  - Documentation and books
  - Personalisation based on access
- Second release:
  - Modules (eg PIL, DBI etc)
  - Web Services and Passport interfaces
  - On-line chat, polls, customisable home page

# Side track: current ActiveState site



- Scaling Zope the easy way, "Baking"
- Back end built in Zope
  - Ugly, ugly DTML and ZClasses
- Perl script writes pages out to IIS
- PerlEx to serve out dynamic stuff
- Works great, get to use Zope and keep it very scalable

# Current ActiveState site



- Good bits
  - Fast
  - Zope back end totally isolated
  - Chance to alter content
- Bad bits
  - Administering IIS real pain
  - Perl script
  - URLs not the same in IIS as Zope due to acquisition

# Back to ASPN... Resources



- Given 3 weeks from specification to launch for most of the features
- Resources:
  - 4 Developers with lots of Perl, SQL and a bit of Zope experience
  - 1 HTML / Front end person
  - Average of 75% of time to spend on it



# Choosing a design (1)



Zope was not a shoe in, considered alternatives eg: PHP, mod\_perl, PerlEx etc

- Fastest execution, most scalable, most expensive: Apache, mod\_perl and Oracle
- Easiest to develop: Apache, Zope
- Time to develop was more important
- Ability to integrate with existing site helpful



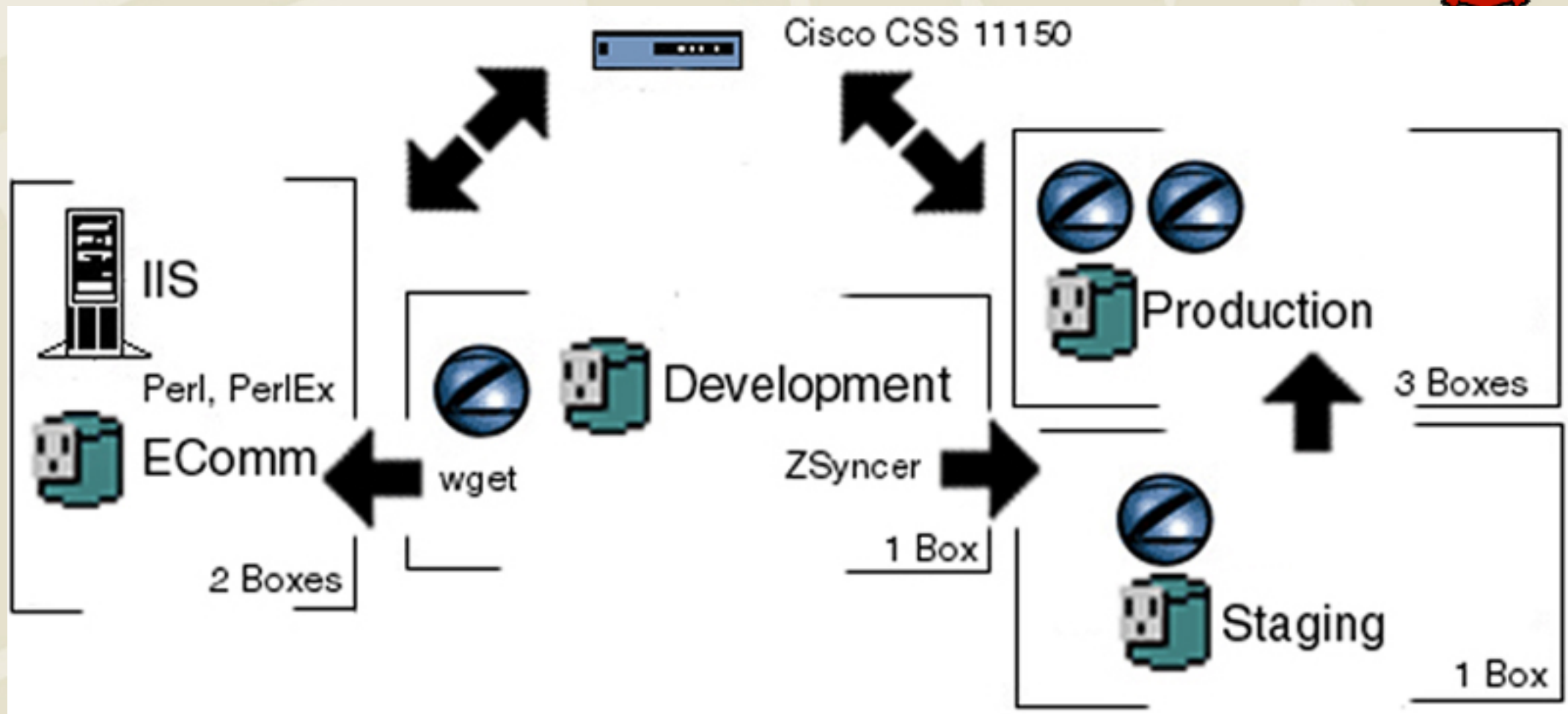
# Choosing a design (2)



After much debating we chose:

- Multiple Zope front end
- MS SQL Server back end
  - More faith in SQL than ZCatalog
  - Already been burned by ZODB and ZCatalog in Zope 2.1.6
  - ZEO not needed
- Cisco router for load balancing

# Design



# Testing



After some rapid development and one Python conference...

With two days left we were able to devote some time to meaningful testing unfortunately

- It wasn't very fast but we had no time left to do major overhauls
- Wrote some simple testing tools using
  - xmlrpc to automate some unit tests
  - webchat to automate testing html

# Traffic (approx)



- 10 mn hits per month
- 500,000 visits
- 2 ms to serve a page (?)
- about 76 Gigs a month of traffic
- Similar size to Zope.org
- Not as much as we had hoped

# Database Size



As of 30 Jan 2002 there are:

- 800k e-mails from 142 mailing lists
  - growing at 10,000 a day
  - indexing takes about 1 hour on our servers (750 PIII, 512MB)
- 40k pages of other content
- 34k registered users

# Performance Issues



- ZODBCA single threaded
- Too many requests coming through to Zope
  - Some images
- Some sloppy coding
  - eg: cyclic reference (adding REQUEST to REQUEST)
  - returning large results to DTML rather than a good SQL statement
- Memory leaks



# Lessons Learned (SQL)



- MS SQL is not as great as we thought
  - Slow to index content
  - More problems with SQL and connection than anything else
  - Properly used ZEO and ZCatalog is probably better for full text
- Tight control, unfortunately people write stuff in Zope because its easier then it doesn't get indexed



# Lessons Learned (Caching)



- Redesign the front end to make caching easier
  - Unfortunately lots of ASPN is designed to be dynamic
  - The most popular pages (Home, mailing list are dynamic)
- Cache as much as possible
  - Start with the easy things: images, css, static content
- Caching per thread is nowhere near as useful as caching across all the threads (especially if you are running with lots of threads)

# Things we got right



- Separating development and production totally
- Trying to avoid DTML when possible (not enough)
  - Writing Python products is good, anything else is bad
- Having separate database that can be written to through many different ways
- Using a SQL back end
- Once the content switch was stable it rocked

# Improvements



- Waiting for stable Zope (2.4.4 looking good so far)
  - ZmxODBCA working plus caching SQL statements
  - RAM cache manager and Accelerated cache manager
  - Consolidating DTML
  - Removing Site Access
- Performance result:
  - Approx 2x performance
  - Hard to measure until into production

# Questions?



- ASPN: <http://aspn.ActiveState.com>
- WWW: <http://www.ActiveState.com>