



The College at
BROCKPORT
STATE UNIVERSITY OF NEW YORK

HighEdWeb 2010 TPR6: Transitioning to a Multi-tier Web Environment

Steve Lewis
Web Manager

Overview

1. Where we started
2. Transition Process
3. New Production Environment
4. New Pre-production Environment
5. Change Management
6. Future Plans



Spring 2009 Status

- One Sun E450 Server, Two Zones
 - Solaris 10
 - Apache 1.3
 - MySQL 4
 - PHP 4 / mod_perl
- Custom build of Apache



Division of Labor

Flexo

- User accounts / FTP
- Development Web Server, Database

Bender

- Serves www
- Database Server
- Shared wwwroot System



Structural Problem

Development		Production
/web/cake	Project Files	/web/cake
/web/auth	DB Config, etc.	/web/auth
/usr/local/cake	Cake Library	/usr/local/cake
.../apache/htdocs	Production Site	/web/live

Deployment of new projects from development to production was a concern because the environment changed too much.



Systems Challenges

- Afterthought of the Web
- Unclear segregation of duties between system administrators, Web team
- Challenges of keeping custom Apache up-to-date
- Strategic decision to embrace Linux



Beginning the Transition

- Move the databases first
- 72 databases on production server
- Most are public read, admin write



Typical Migration Algorithm

- Disable administrative access
- Move data
- Create accounts
- Move public site to new DB server
- Configure admin to new DB
- Restore administrative access

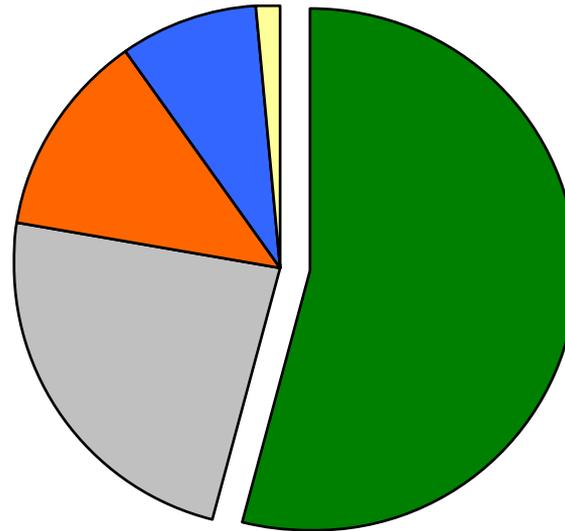


Automated Script

```
mysqldump  
-h www.brockport.edu  
-u xfer  
-p'$REMOTE_PW'  
--create-options  
--databases $DB_TO_MIGRATE  
| mysql  
-u webadmin  
-p'$LOCAL_PW'
```



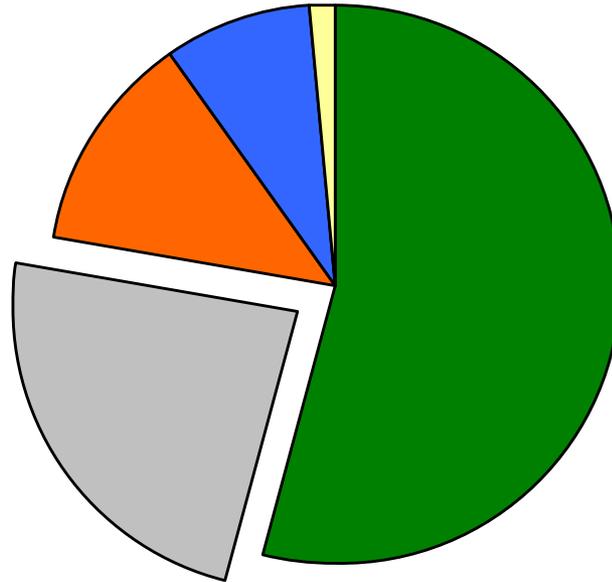
Not Quite as Bad as it Seemed



- 39 of 72 databases moved



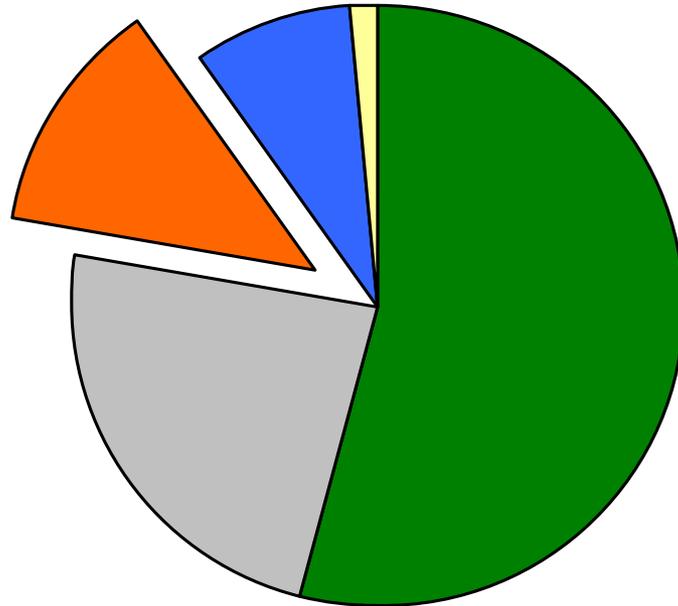
Not Quite as Bad as it Seemed



- 17 outlived their projects



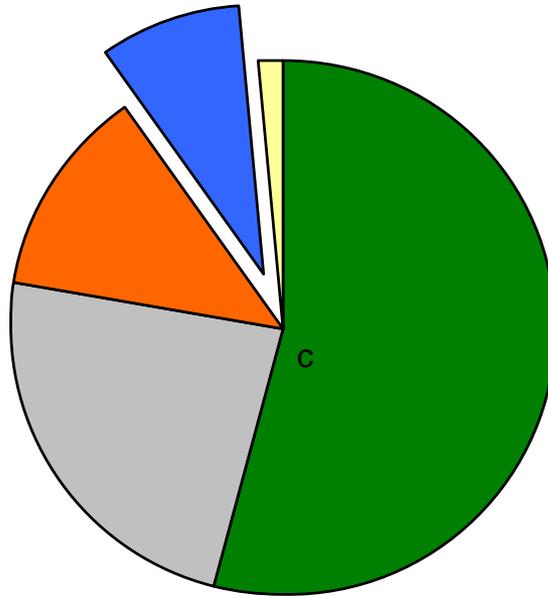
Not Quite as Bad as it Seemed



- 9 abandoned in development



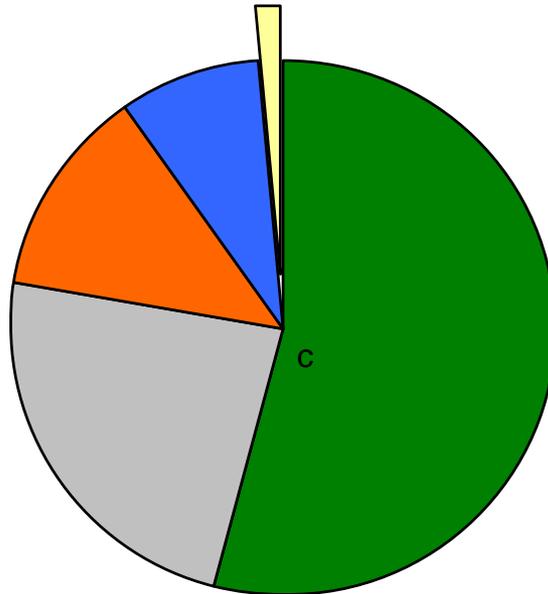
Not Quite as Bad as it Seemed



- 6 related to HighEdWeb – migrated elsewhere



Not Quite as Bad as it Seemed



- 1 system table (migrated by magic)



Old Decision Pays Off

- Most projects use a central config file for access parameters. Adjust server name, username, password and done.



Create the New Web Server

- Need to start testing
 - Mod_perl automated changes
 - DB-interactive programs
- Possible disruption of Live Databases
- Complete copy of the site



Attack of the Clones!

- Clone the database server for testing
- Establish process to synchronize data
- Add custom data to identify test database
- Adjust host fields on mysql database
- /etc/hosts hack on new Web server



Looooooots of Testing

- PHP 4 to PHP 5
- Avoidance of mod_auth_external
- Involve the campus community in review
- Halt development on anything unrelated to compatibility with upgrade
- Daily rsync of files changed in production



Create New User Access Machine

- Old system had separate credentials
 - Migrate to Collegewide NetID (LDAP)
- Requiring SFTP for new connections
- Return to NFS for page updates



The Cut-over

- Communication of upgrade plan, timeline
- Disable user access to update site
- Run final rsync of files
- Duplicate production database host permissions
- Remove /etc/hosts database hack
- Switch DNS for www.brockport.edu
- Switch DNS for user access
- Notify users of status, change to configuration
- Retire former system as hits → 0



But We're not Done Yet

- Recall challenges w/ prior development environment
- We have that testing DB server
- Stand up a development box



Send in the Clones (or not)

- Could have cloned
 - Didn't want GBs of stale HTML, images
- Needed:
 - Apache configuration
 - CGI
 - Cake MVC files
 - Other projects as needed



Changes to Testing DB

- Continue to update host permissions
- New development gets discrete DB name
 - Convention of _pprd to name
 - Need to recreate permissions daily
- Modifications to existing projects' DB schemas get tested **every day** until rolled into production



Example: Quickreg

- Copy production to webdev
- Confirm DB credentials to testing DB
- Program
- User Preview on webdev (rinse, repeat)
- Deploy updates to production



Update Process

1. Backups (PHP, mysqldump)
2. Temporarily disable application
3. Run SQL updates on production
4. Copy application update to production server (preserving public content)
5. Test to confirm it all works
6. Remove SQL updates on development (or restore from backups)



Brockport's Future

- Remain on VMWare
- Establish Versioning for Web Projects
- Add new Web servers/proxy



The College at
BROCKPORT
STATE UNIVERSITY OF NEW YORK

Evals: TPR6

Questions

Steve Lewis

@stebert

slewis@brockport.edu



The College at
BROCKPORT
STATE UNIVERSITY OF NEW YORK