

Enterprising Moodle 2.0

A Comparison of Architectures

Centre for Teaching and Learning
&
Academic Information and Communication
Technologies
University of Alberta

Outline

- Background
- Requirements and challenges
- Moodle architecture
- Single instance architecture
- Multiple instance architecture
- Multiple instance design
 - Portal instance, caching, satellite data synchronization
- **System infrastructure**
- Review and Questions

Background

- Blackboard Vista (formerly WebCT Vista) has been the current centrally supported LMS at U of A since 1998
- U of A is moving from Blackboard Vista to Moodle
 - Active campus Moodle community since 2005
 - LMS committee has recommended Moodle as a secondary LMS after an LMS review in 2008
 - Support from Blackboard Inc. for Blackboard Vista will terminate as of January 2013
- Further information, go to Patrick Thibaudeau and Scott Delinger's presentation on Thursday morning on ***Transition From Blackboard to Moodle 2.0***

Requirements & Challenges

- Scale: 8,000 course sections in central LMS in 2010, and 50,000 active student users
- Different demands on settings, features and plug-ins
- Functional baseline of the current LMS
 - Popular LMS tools
 - Admin features for certain faculties and departments
- Integration with the Student Information System (SIS)
 - Account management
 - Course creation and enrollment
 - Grade submission to SIS
- Other integration such as media server, Google apps, ePortfolio, etc.

Definitions

- ***Moodle instance***: a separate installation of Moodle
- ***Application architecture***: a logical organization of one or more Moodle instances, the overall control structures, as well as the protocols for communication and synchronization if necessary
 - How many Moodle instances: single vs multiple
 - How they are integrated
- ***System infrastructure***: a conceptual design of the underlying system hardware, both physically and virtually.
 - Hardware choice and virtualization
 - Server configuration

Single Moodle instance

- Pros

- Single set user records on a single database
- Single system to maintain: easy to backup, upgrade, etc
- Enrollments are handled on a single server, no need for synchronization or extra development work
- Defaults at top level cascade to all sub-levels

- Cons

- Hard to provide separate administrative privileges to different Faculties and Departments
- Site settings affect the whole institution
- Unable to accommodate any external Moodle instances on campus
- If the instance fails, no course will be accessible. For example, a poorly designed plug-in could jeopardize the entire system

Multiple Moodle Instances

● Pros

- Administrative privileges can be easily offered at the Faculty level
- Modules and default settings can be customized differently among instances
- More options for scalability (different instances can run on different application servers and database servers)
- Ability to isolate and address individual system bottlenecks

● Cons

- Requires much custom development work to integrate multiple instances to ensure seamless user experience
- Multiple instance might cause some plug-in compatibility issues
- Certain features might not work the same, such as Blog, Note, etc.
- Without extra development work, no cascading defaults from central to other instances

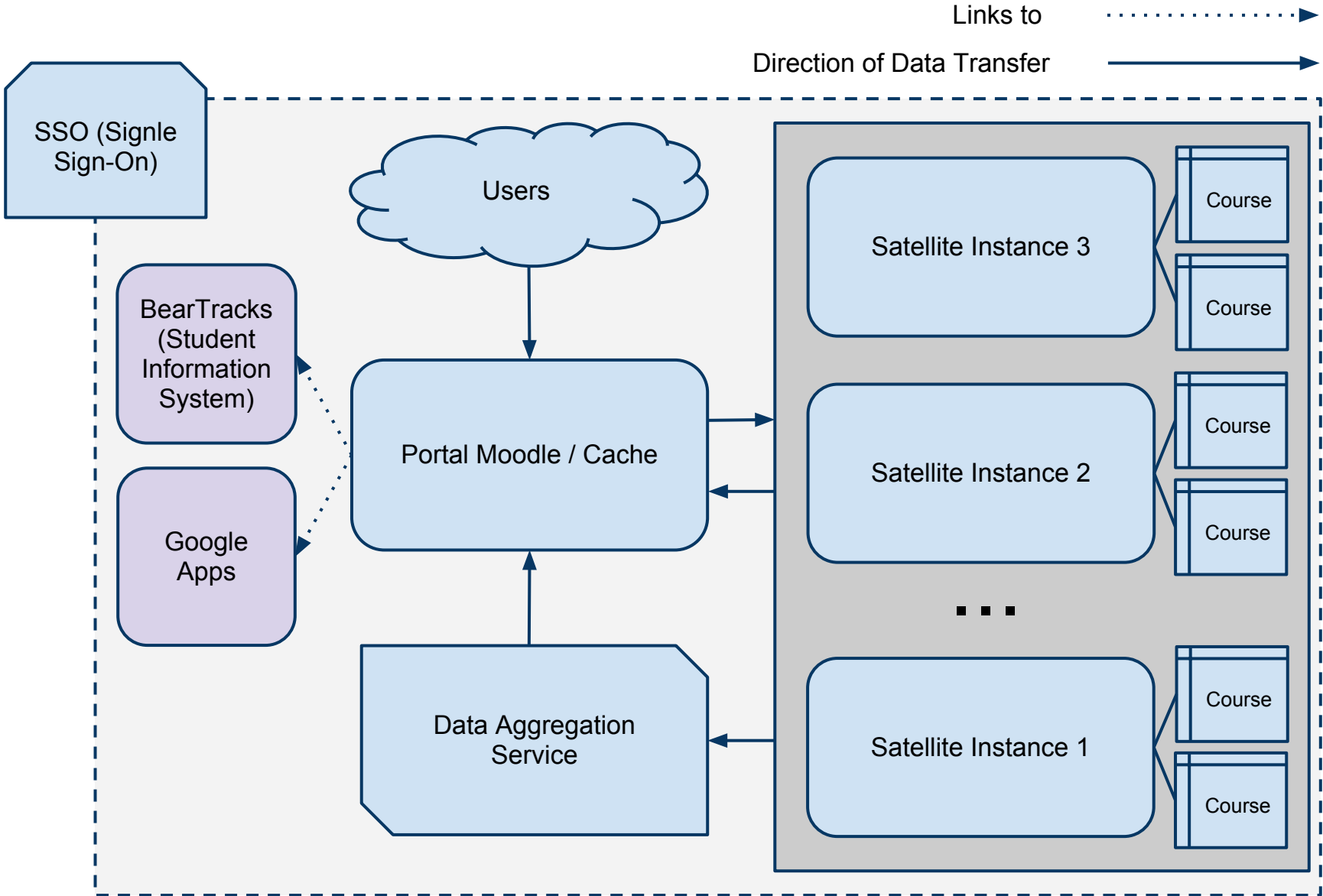
Multiple Instance Prototype

Demo Video

Why Have Multiple Instances?

- Need multiple levels of administration.
- Different module offerings for certain faculties.
 - e.g. Available languages, available themes.
- Different administrative settings.
 - e.g. Course default settings for a unit. Specific tool settings/defaults.
- Administrators for specific groups of courses.

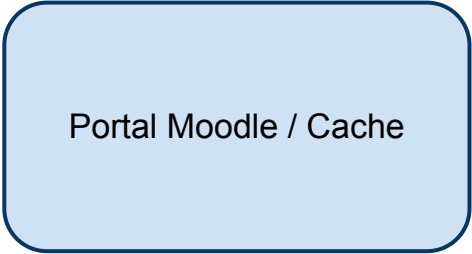
Multiple Instance Design



Requirements

- Authentication
 - Single Sign-On (SSO)
- Consistent Experience for users
- Data Synchronization:
 - Messaging
 - Settings
 - Messages
 - Calendar Events
 - Course Lists
 - Profile Settings
 - Navigation Links

Portal Instance



Portal Moodle / Cache

Role:

- Act as central HUB between all satellite instances.
- Provide and collect data to and from satellites.
- Cache storage

Data Caching on Portal Server

- Data:

- Consolidated

- Course Lists

- Messages

- Calendar Events

- Common

- Blogs

- Profile Data/Settings

- Recent User Logins

- Caching mechanism:

- Custom tables in Moodle DB

- Moodle's messages tables.

- Currently direct db inserts. Ideas?

- Allows us to use Moodle's built-in popups.

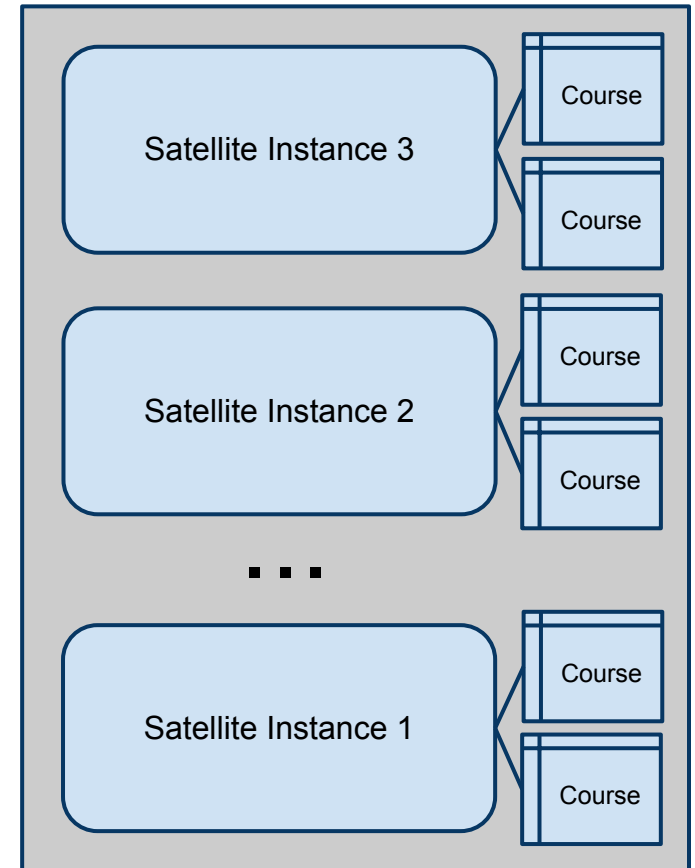
- Updated via separate XML-RPC Web Service server.

- Allows for hardware separation from portal system

Satellite Instance

Role:

- Store and provide courses
- Provide an administrative level for unit administrators
- Allow unique sets of plug-ins to be enabled/disabled for courses
- Transfer relevant data to Portal Cache



Satellite Instance Data

- What to transfer:
 - Messages
 - Calendar Events
 - Profile Data/Settings
 - Recent User Logins on Portal
- Pushes (Messages,Calendar Events) to Data Aggregation Service via XML-RPC client.
- Pulls (Profile Data, Recent User Logins)

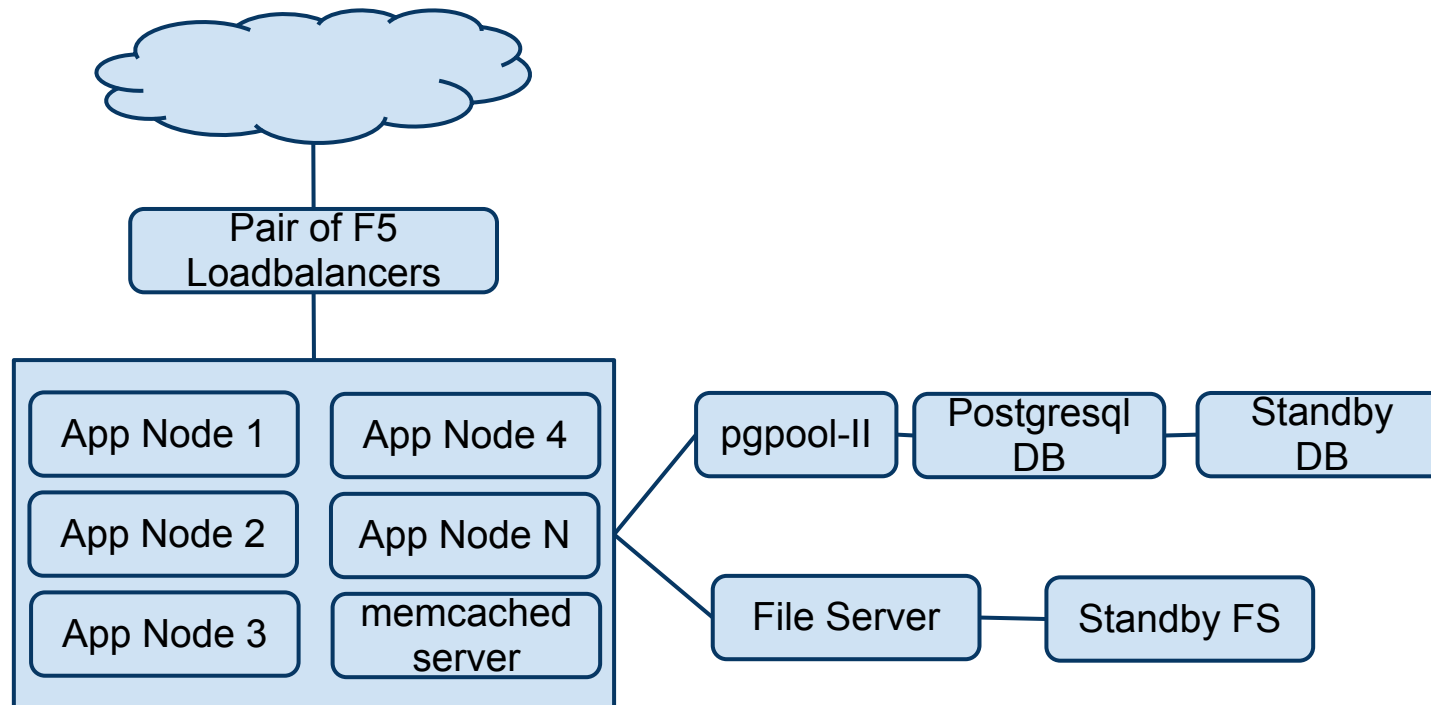
How Satellite Instances Sync

- Priority Queue of users (time-stamp)
- Process:
 1. N users are pulled from top of Queue
 2. User data is extracted for those users
 3. Data is encoded and transferred to Portal Cache via Data Aggregation Service
 4. User profile information/settings pulled from Data Aggregation Service
 5. List of recent user logins, newer than a stored time-stamp are retrieved.
 6. Users older than a threshold, who exist in recent user logins are re-prioritized to top of queue.

Making things easier

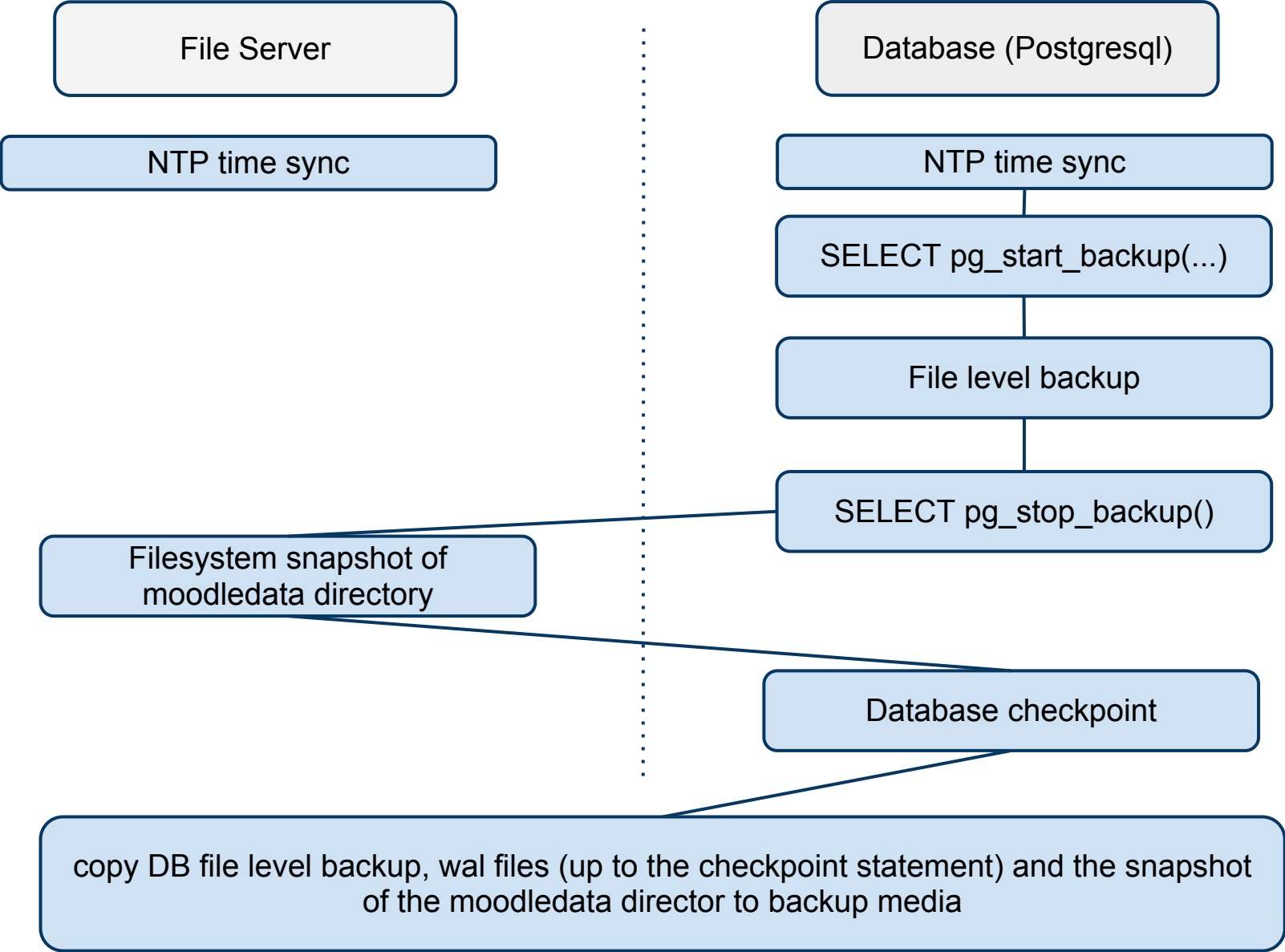
- Implement arbitrary levels of administration. (e.g. Categories)
 - Tool Settings
 - Enabled/Disabled modules
 - Category associated roles (category admins).
- Increase core event system usage.

System infrastructure



- VMWare cluster made up of 2U servers, 2CPU / 12 Cores, 72 GB RAM.
- Database and File Server nodes are pinned to dedicated hardware.
- 6.5 TB Raw FC and 30 TB Raw Sata

Consistent online backups



Discussion - Questions

- How do you manage the Learning Context Hierarchy
- Settings at site level
- How to do online backups
- How do you address different groups' needs on the central LMS
- How do you handle plug-ins (policy and process) on a central LMS
- What is your process to handle feature requests from end-users

Notes to remember to add:

- Image for Portal storage structure
- Slide for processes (pushing, events framework not used enough in the core to make module communication efficient, eg. event triggers would allow dynamic data syncing between instances alot easier.)
- Communication Methods - Web Services (XML-RPC), AJAX
-

Data Merging

- Need access to data on Portal
- Calendar Merging - requires some code additions to the calendar module. Displays events from local database as well as the local "remote" database.
- Course List Merge - AJAX...
- Message Merge - Messages are slightly different than the other data merging processes since they are moved rather than copied to the central...

Challenges

- Scale:
 - about 8,000 courses offered in Central LMS in 2010
 - over 50,000 students take courses in the LMS
 - about 80 departments from 20 faculties uses the LMS
- Centralized and decentralized support
- Levels of administrative access
- Different demands on settings, features and plug-ins
- Addressing the above challenges and minimizing core changes is challenging
 - Create new blocks instead of changing core provided blocks.
 - Restricting core changes to "Central/Portal" Moodle instance.