

InC-Atlassian

InCommon Atlassian

This is the home page for the InC-Atlassian space. This page contains comments and thoughts, posted from various campuses.

A summary of the issues can be found [here](#). **NOTE: These issues are not in any particular order...**

Topics:

1) SSL acceleration.

- This was in our original design, fortunately, so we currently run an SSL offload and load-balancer appliance (a Citrix NetScaler) in front of the Confluence instance. This provides some firewall-like protection and performance enhancements. It would be great to be able to wrangle multiple Confluence machines via the load-balancing feature, but we'll talk a bit more about that in the "Clustering" section below. (SC)

2) User account and group management.

- Currently, we take advantage of our campus-wide LDAP resource for user authentication. That is great (users can leverage a password they are used to using, not generate/maintain something new), but without LDAP group integration for content authorization, it only takes us so far. So, we've been experimenting quite a lot with Atlassian's identity management tool, Crowd. The 1.0 - 1.1.1 releases couldn't handle large LDAP directories like our own (~500,000 user and group objects), but today saw the release of 1.1.2 and Atlassian is hopeful that it will take care of most (if not all) of our user and group management scalability problems. I'll be digging into that shortly. (SC)

3) Back-end storage (database, filesystem).

- Confluence taps into our Oracle infrastructure for a database back-end. Because of performance concerns in the initial design, we opted to leave attachments outside of the database and on a filesystem. For that, and for the nightly backup XML files, we mounted a volume provided by our NAS cluster. This seems to work well, but we really wish we could turn it into a clustered and load-balanced Confluence configuration. More on that next...

4) Clustering.

- When Confluence became available, it required that attachments be stored in your database, something we had not done (for performance reasons, as mentioned above). It would be nice if Atlassian supported an alternate, filesystem-based (via NAS, for example), mechanism for cluster configuration. Without that, we would be looking at migrating all of the current attachments into the database and that process, while we haven't examined it, doesn't sound like fun. With it, we'd be ready to cluster immediately and could leverage our external load-balancing appliance.

5) Mirroring.

- We realized some time ago that Confluence had become an integral part of our daily work and that, when it is unavailable, life becomes difficult. To combat this, we devised a plan to erect a secondary Confluence (built as a standalone entity - no reliance on NAS, external databases, etc) and populate it daily with the previous night's backup of the primary system. This would provide emergency access to the Confluence content, especially for those who might need it to rebuild services in the event of a disaster. We haven't built this yet, but it looks quite doable with a bit of scripting. It would be very handy, though, if Atlassian somehow provided Confluence-to-Confluence mirror functionality out-of-the-box.

6) Migration and consolidation.

- We've found that other units who have been running their own wikis are, in many cases, quite eager to move all of their content into a central Confluence service so that they can retire their own wiki and make use of Confluence's considerable advantages. Atlassian has provided some really good tools, which we have utilized, for merging the content of other wiki tools (SnipSnap, etc) into a Confluence instance, but merging all or part of a different Confluence instance is not as smooth.

7) Backup and restore.

- I love the simplicity and ease-of-use when it comes to Confluence's backup and restore procedure. It's one of my favorites, really. Unfortunately, though, it only allows you to restore the whole Confluence instance (wiping out everything that was there before the restoration procedure began). This is great for disaster recovery, but not very practical for fixing user accidents ("Oops! I deleted my space!" or "Oops! I deleted her space!"). It would be very good to have partial restore (space-by-space and page-by-page) capabilities.

8) page-watch email bug.

- Any user can set a watch on a space s/he is able to view (to be notified by e-mail of changes), even if s/he doesn't have a valid e-mail address in his/her profile. If a user without an e-mail address has a watch set on a space, any attempts to edit a page in that space results in the resulting attempt to send the watch mail failing and this in turn causes the edit attempts to fail with a very unhelpful/misleading generic error message. Thus a careless action by any one user renders an entire space uneditable.
- There is apparently no way to review which users have watches set on a space. This seems to be a glaring omission that makes managing the platform much more difficult (due to the bug noted above) and eliminates a very valuable piece of information about usage patterns.
- For many reasons it would be much better to at least have the admin option of requiring a valid e-mail address in a profile before a user is able to complete the registration step.

9) We were running version 2.2.6a.

- Last week, we upgraded to 2.5.3. However as part of their "method 1" (which we had to do because our database is larger than 2 GB (who's isn't???)), we had to point the confluence.home variable at the 2.26a directory to make the 2.5.3 instance functional. I'm told by their support that deleting the 2.2.6a directory would kill 2.5.3. They haven't been totally clear on what I have to do to eventually get rid of 2.2.6a, but this is a bit silly that a new install depends on an older version's directory. If there's info they need in there, why isn't it simply copied over at upgrade time?

10) Managing Users and Groups.

- (Brown) We are currently running Confluence 2.5.3, wired to our central ldap server. Ldap currently has about 18K user objects and 55K group objects. We use Ldap for user authN (passwords are actually in Kerberos) and for authZ. We currently centrally manage the processes for creating new spaces and specifying which users and groups can access a given space; we *really* want to be able to further delegate this functionality (to departments and groups). To do the AuthZ functionality, we rely on the current Confluence pages; there are several issues:
 - a) (Use case CONF-8662) Administration : Manage Groups : View Members of Group : Browse Pagination -- Pagination does not adequately support thousands of members
 - b) (CONF-8675) Administration : Manage Groups : Browse Pagination -- LDAP queries select every group name for each page load (potentially thousands); Pagination does not adequately support thousands of groups
 - c) (CONF-8663) Add Page : Restrictions : Choose Users -- Browse Group Membership form violates information security policy by exposing group membership to end users
- Suggestion -- Provide APIs to make it easier to
- a) use an external tool to manage spaces and manage permissions on those spaces; this tool might work better in environments with tens of thousands of users and hundreds of thousands of groups,
- b) allow a site to configure Confluence to disable the use of Confluence's Manage Groups pages.

11) Federated Access.

- *(Michael... can you fill this in?)*

Hi Chris Phillips (chris dot phillips at Queens u dot ca) from Queen's University in Kingston Ontario here. I figure I would jump in and throw some info out there about what we are doing.

We use confluence but are working toward an implementation to integrate with the Canadian Federation that we are involved in. We have a draft implementation page on the go at <http://wiki.its.queensu.ca/display/heidm/HowTo+Building+a+Shib+SP+in+Solaris+10+zone+on+Sparc+for+Confluence+Wiki>.

Much of the insight was provided by the internet2 reference created by Gary Weaver, but we had to use the regular user objects which I think may cause us some issues in the future.

From a federated standpoint, it's great, especially when you couple the Custom Space Usergroups Management Plugin with it. This at least permits you to push the group management into the space owner which I see as a significant shortcoming of the native confluence product itself (ie. lack of ability to delegate group mngmt out of the box that is).

We are at the point where we are engaging with our partners in the federation to deal with release of appropriate attributes.

We do see a challenge coming down the pipe of for privacy elements and would like to hear how others dealt with this. (what do you do when someone declares that their records should be hidden? Do you replace their common name with 'PRIVACY REQUESTED' ??? what about email addresses?

12) Confluence as Content Management System

- from Vince @ UMich) One item that I would add is that we are currently looking at Confluence as a possible Content Management System that would replace our existing web presence. This would allow content to be created and maintained by staff without web programming experience. It would also offer many other benefits including versioning. Once concern we have is how to manage public spaces along with internal spaces containing sensitive data. We are currently experimenting with this.

13) CROWD

(from Wayne at Umich) Having just got my OpenId to access this wiki, I wonder about the future role of CROWD as an integral part of large scale implementation. CROWD is openid enabled and when we asked about doing user/group management using an external LDAP server we were told that CROWD is where the functionality lies.

14) Policy

We use Confluence as our IT wiki supporting around 200 users. One of the main issues we ran into was developing good policy. Here are some example policies:

- <https://depts.washington.edu/wiki/>
- <http://wiki.geneseo.edu/display/pol/Home>
- http://wiki.spsu.edu/index.php/Minutes_of_Coordinating_Meeting
- <https://wiki.brown.edu/confluence/display/BrownWiki/Policy+and+Documentation>