L&B Notification Keeper

Zdeněk Šustr et al.
• **Logging and Bookkeeping (L&B):** job-centric grid monitoring
  - collecting events from grid elements
  - using that information to reconstruct job state
  - presenting a single point of contact for users to...
    - query the state of their jobs
    - register for automated notifications on job state changes
    - register user tags
    - control job info access
L&B Notification Interface

- **L&B notifications** – publish/subscribe model implementation
  - Native (legacy) delivery mechanism (since 2004: L&B 1.0) reusing L&B’s reliable event delivery chain:
    1. Register and specify conditions
    2. Obtain Notification ID
    3. Use Notification ID to listen and receive messages
  - OpenWire/STOMP-based messaging (since 2011: L&B 3.0)
    1. Register, specify conditions + messaging topic
    2. Listen to topic, receive messages

![Diagram of L&B Notification Registration and Reception paths](image)

*Fig. 1: Notification Registration and Reception paths in L&B*
Maintaining Registrations

- Registrations always **temporary**
  - Prevents accumulation of messages if listener disappears
  - Typical 12-h expiry
  - Refresh possible any time within the life span
  - New registration required if left to expire
    - Events arriving in the meantime won’t be reported

- Reliable delivery within registration validity period:
  - Legacy: messages never discarded before delivery
  - Messaging: likewise but message considered delivered if accepted by broker, the rest depends on msg infrastructure

- Keeping registrations fresh
  - Manually by calling `glite-lb-notify refresh`
  - L&B harvester (heavy-weight notification client)
    - Currently only supports legacy notifications
  - Server-side `glite-lb-notif-keeper` **new in L&B 3.2**
    - Run by `cron`
    - Primarily intended for notifications delivered over messaging
Notification Mini How-To

- Registration by `glite-lb-notify new <conditions>`
  - `-j` Specify job ID (report on single job)
  - `-o` Specify owner DN (all jobs for that owner)
  - `-v` Specify VO (all jobs for that VO)
  - `-c` Send message only if job state changes
  - `-T` Send only if job reaches terminal state
  - `-a` Alternative delivery path
    - `-a x-msg://<topic>`
  - `-N` Anonymize user identities in messages
... See documentation for more options

- Refresh by `glite-lb-notify refresh <NotifID>`
- Use NotifID as URL in your browser to view registration details
- Unregister by `glite-lb-notify drop <NotifID>`
  - L&B super users may drop any notification
• Run by cron, reading /etc/glite-lb/site-notif.conf
• Reports to syslog
• Config file, one record per line:
  <handle>  <conditions> -a x-msg://<topic>
• The *keeper* script creates a registration for each record or, if it already exists:
  1. Checks how long ago have messages last been accepted
  2. Checks the size of the message backlog
  3. Refreshes registration unless limits were exceeded
• Stale registrations with excess backlog are left to expire
  – by themselves within 12 hours
  – ...Super Users may drop them immediately if required
• New registrations will be created on the next *keeper* run
  – ...but the backlog will be cleared
  – Comment-out (#) the corresponding line in site-notif.conf to get rid of an unwanted registration for good.
And now...

Demo