



La journalisation

Principes et implémentation

Sébastien LAMPS

IR3 Groupe 2

Exposé de Système

Ingénieurs 2000, Université de Marne la vallée

Plan



- ✧ La journalisation, c'est quoi ?
- ✧ La journalisation, à quoi ça sert ?
- ✧ Une application répandue: Syslog
- ✧ La journalisation en Java:
 - ❖ Jakarta Commons Logging
 - ❖ Log4J



Le principe de la journalisation



Définition:

Inscrire dans un journal, ou *Log*, les évènements du système au cours de son activité.



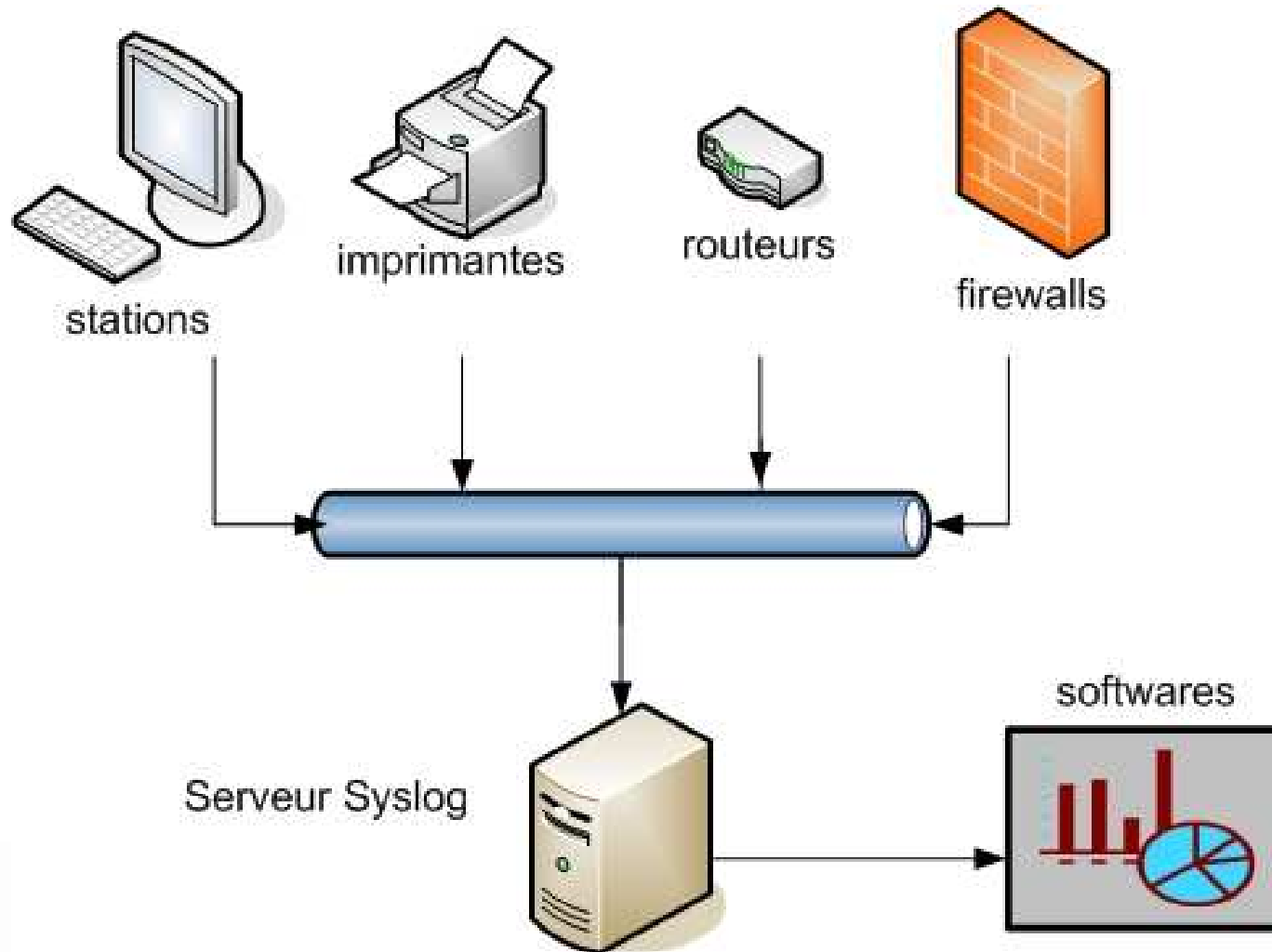
Les utilisations de la journalisation



- ✧ La sécurité et le contrôle
- ✧ Le recouvrement des données
- ✧ Le debugging
- ✧ La génération de rapport d'activité
- ✧ La garantie juridique
- ✧ La rapidité des sauvegardes
- ✧ Etc



Un exemple: Syslog



Exemple de journal Syslog



- ✧ **MySelf:~# cat /var/log/syslog**
- ✧ Dec 7 06:25:05 localhost syslogd 1.4.1#17: restart.
- ✧ Dec 7 06:39:01 localhost /USR/SBIN/CRON[928]: (root) CMD ([-d /var/lib/php4] && find /var/lib/php4/ -type f -cmin +\$(/usr/lib/php4/maxlifetime) -print0 | xargs -r -0 rm)
- ✧ Dec 7 07:03:07 localhost -- MARK --
- ✧ Dec 7 07:17:01 localhost /USR/SBIN/CRON[950]: (root) CMD (run-parts --report /etc/cron.hourly)
- ✧ Dec 7 15:02:11 localhost proftpd[1182]: 192.168.0.3 (84.90.94.31[84.90.94.31]) - FTP session opened.
- ✧ Dec 7 15:02:12 localhost proftpd[1182]: 192.168.0.3 (84.90.94.31[84.90.94.31]) - mod_delay/0.4: delaying for 64 usecs
- ✧ Dec 7 15:02:12 localhost proftpd[1182]: 192.168.0.3 (84.90.94.31[84.90.94.31]) - FTP session closed.
- ✧ Dec 7 15:17:01 localhost /USR/SBIN/CRON[1194]: (root) CMD (run-parts --report /etc/cron.hourly)
- ✧ Dec 7 15:39:01 localhost kernel: USB Mass Storage device found at 2
- ✧ Dec 7 16:03:08 localhost -- MARK --

proftpd[1182]: 84.90.94.31 - FTP session opened

kernel: USB Mass Storage device found at 2

La journalisation en Java



- ✧ Des journaux, sous plusieurs formes possibles
- ✧ Des méthodes d'ajout d'évènements
- ✧ Des niveaux de criticité:
 - ❖ TRACE
 - ❖ DEBUG
 - ❖ INFO
 - ❖ WARN
 - ❖ ERROR
 - ❖ FATAL



Jakarta Commons Logging



- ❖ API ultra simple
- ❖ Le journal: une interface Log
- ❖ Une factory abstraite: LogFactory
- ❖ Par défaut, utilisation de SimpleLog



La factory de Log



La factory abstraite recherche une factory dans:

- ↪ Les attributs de la JVM
- ↪ La recherche de service du JDK 1.3
- ↪ Le fichier commons-logging.properties

Par défaut:

`org.apache.commons.logging.impl.LogFactoryImpl`





L'interface Log

2 méthodes pour chaque niveau XXXX:

- ❖ *isXXXXEnabled*: teste la prise en compte du niveau
- ❖ *XXXX*: enregistre le message

Exemple:

```
Log myLog = LogFactory.getLogFactory().getInstance();  
if(myLog.isInfoEnabled()){  
    myLog.info(« J'aime travailler»);  
}
```



L'instanciation du Log

Le LogFactoryImpl recherche une implémentation dans:

- ↳ Les propriétés (JVM et fichiers)
- ↳ Le système de journalisation Log4J
- ↳ Le système de journalisation JDK 1.4
- ↳ Le système de journalisation JDK 1.3

Par défaut:

`org.apache.commons.logging.impl.SimpleLog`





Log4J

- ✧ API complexe et répandu
- ✧ Configuration par propriétés ou XML
- ✧ Niveaux définissables (Level)
- ✧ Notion de hiérarchie des journaux (Loggers)
- ✧ Notion de format de message (Layout)
- ✧ Notion d'héritage des récepteurs (Appenders)
- ✧ Système de récepteurs de secours
- ✧ Modification du niveau minimal à la volée (WatchDogs)



Les récepteurs (Appenders)



- ✧ JDBCAppender
- ✧ NTEventLogAppender
- ✧ TelnetAppender
- ✧ NullAppender
- ✧ SMTPAppender
- ✧ SyslogAppender
- ✧ ConsoleAppender
- ✧ FileAppender
- ✧ LF5Appender
- ✧ RollingFileAppender



Format des messages (Layout)



- ✧ **SimpleLayout**
- ✧ **PatternLayout**
- ✧ **XMLLayout**
- ✧ **HTMLLayout**



Marre du Java ?



- ✧ Log4Cxx (c++)
- ✧ Log4Net
- ✧ Log4Perl
- ✧ Log4PHP
- ✧ Log4PLSQL



Sources



- ✧ <http://jakarta.apache.org/commons/logging/>
- ✧ <http://logging.apache.org/> (log4j)
- ✧ <http://beuss.developpez.com/tutoriels/java/jakarta/log4j/>
- ✧ Un livre: Log4J (20\$)

