

iLogger

Novell Sentinel Collector Solution

Kuang-Chun Cheng

kccheng@LinuxDAQ-Labs.com, kcc1967@gmail.com



Sentinel Collectors

- Legacy Scripts

- 不易學習，結構不完整，需配合 State Machine 才構成一完整的程式。
- regular expression 與常見 programming language 所支援的語法比較起來較複雜（要多加很多 escape)
- 不是一個完整的程式語言
- 不支援中文 !!



Sentinel Collectors

- JavaScript Collector
 - 使用 Rhino (JS interpreter impl. by Java)
 - Java + JavaScript 的混合，而非單純的 JavaScript. 改進 Legacy Scripts 的許多缺點，
 - New SDK 把 Java 的部份用 Javascript 包起來 ... with more APIs. 但不論如何，依然需要透過“特殊” APIs (不論是 Java 還是 JS) 與 Sentinel 溝通。
 - JS spec 指定義語法，沒有定義到 system level I/O



Collectors Problems

- Collector 的開發與驗證無法獨立於 Sentinel 之外。維護不易。
 - 必須將 Collector 安裝於 Sentinel 上，才能驗證 Collector 的正確性。
 - 雖然 Sentinel 上有 Debugger, 但並不是一個開發環境。
 - Sentinel footprint 太大
 - 雖可使用外部 JavaScript interpreter 做先前開發，但最後驗證仍需有 Sentinel.



Solution

- 使用已經事先驗正過的 collector, 單一的 collector.
 - generic-collector
- 將語法驗證轉成 *data format* 的驗證
 - 將 collector 的開發轉到 Sentinel 的外部處理.
 - 可使用所有手邊拿的到的工具, 沒有與 Sentinel 綁在一起的工具來處理 collector



iLogger Approach

- generic-collector
- syslog-ng
- shell scripts (plugins) or optional logx
 - shell scripts run in background
 - parse log message
 - pack log message in “generic-collector” format
 - use “logger” command send to syslog-ng



**Novell Sentinel
generic-collector**

message in generic-collector's format

**syslog-ng
filtering, backup**

logger

plugin_script &



Data Format

- NVP (name value pair)
- TAG1="VAL1" \t TAG2="VAL2" \t
 - TAGn is Sentinel variable (in Javascript's term)
 - EVT, SEVERITY, MESSAGE, SOURCE_IP, DEST_IP
 - double quote is optional



Shell Script 常用工具

- tail, tailf, read
- echo, cut
- awk, sed, grep, egrep ...
- **logger**



Demo



Demo Setup

- logx -c sentinel-emu.xml (模擬 Sentinel)
 - udp 1468 ----> STDERR (紅色)
 - tcp 1468 ----> STDOUT (黃色)
- syslog-ng
 - match “^sentinel:” 往 tcp 1468 送
 - match “^apache:” 往 udp 1468 送



Q & A



iLogger 缺點

- performace 較差
 - 呼叫太多外部命令
 - Solution: 使用 perl, python, php, C/C++ ... 取代 shell script.
- 缺乏統一管理的介面
 - Solution: follow Novell iLogger 的 rules, 如檔案命名方式, 檔案放置位置 ...
- 使用者還是需寫程式



iLogger 缺點

- generic-collector 容錯能力較差
 - collector 的撰寫使用到“非正式”的語法 `eval()`，相當於在 `interpreter` 中再呼叫一次 `interpreter`。
Sentinel 沒處理這個部份，所以一旦 `data format` 不正確，整個 `collector` 會停掉。
 - Solution: 利用 `script` 做外部檢查。Format 正確才往 `generic-collector` 送。



Behind iLogger

- iLogger 架構可行的關鍵
 - 將程式語法的問題成功轉換成 data format 的問題
- Behind iLogger
 - 將程式設計轉換成 configuration file 的調整。對制式的 data format 如 NVP, CSV, some DB format ... 將不再需要寫 collector.
 - Solution: logx – a flexible log exchange utility
 - 類似 syslog-ng 的企業版
 - xml based configuration file



iLogger + logx

- 目前 Novell 使用的方式，
 - 以 logx 取代部份的 shell script
 - 保留 syslog-ng 做為與 Sentinel 溝通的橋樑
 - 使用 syslog-ng 對 log message 事先做分類，並側錄 raw message 供 debug 使用。

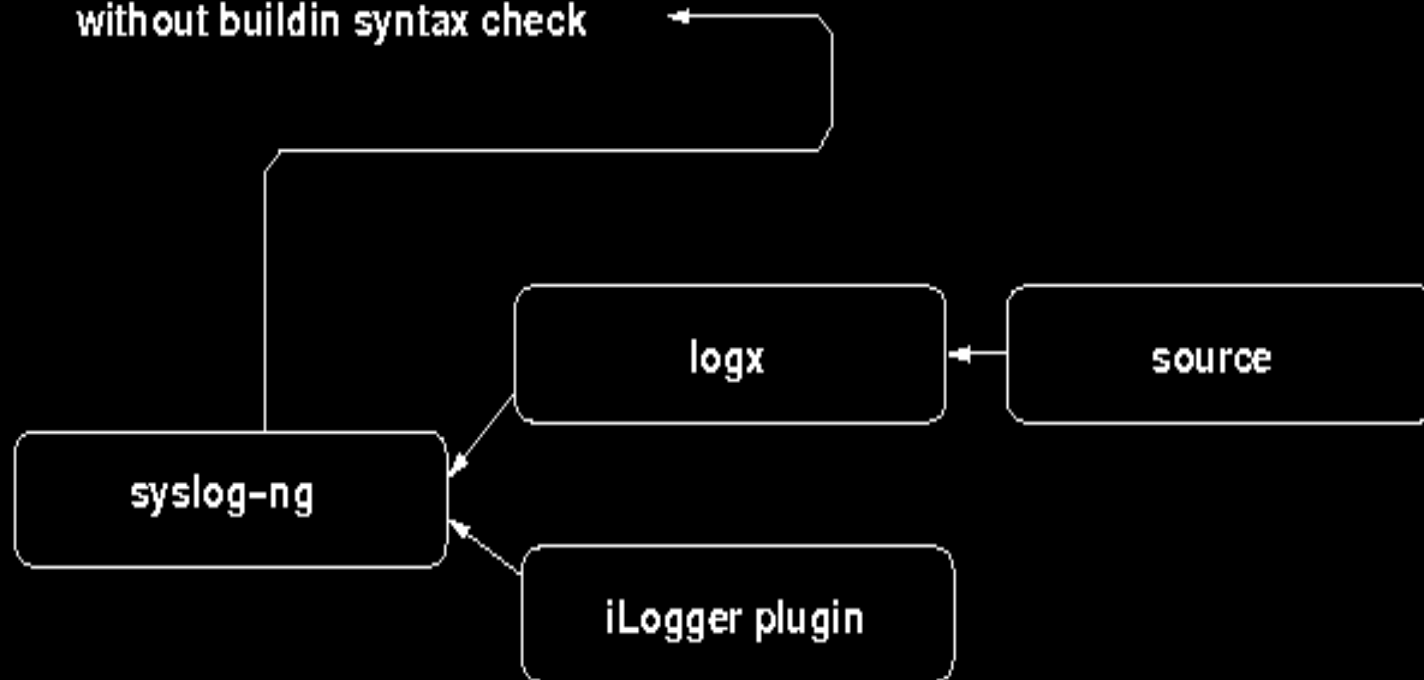


Logx Approach for Sentinel Collectors

Current Approach (iLogger, iKit)

Novell Sentinel

generic-collector (TCP/UDP, 1468)
without buildin syntax check



Logx Approach

- Sentinel Logx-Collector
 - 內建 data format 檢查，容錯能力較嘉
 - 支援更多標準 collector 的可調整 parameters
- optional syslog-ng
- logx



Logx Approach for Sentinel Collectors

Novell Sentinel

Logx-collector (TCP/UDP, 1468)
buildin syntax check

