Dokumentation

Domino Server Hook für Agent Execution Prevention

Version 1.0
Maßweiler, 14.11.2006
Joachim Mutter

Dipl. Ing. Inf. Joachim Mutter
Waldstraße 4
66506 Maßweiler

Mobil: +49 (0) 172 / 7 38 35 25
Festnetz: +49 (0) 6334 / 983403
web: www.sysarc.de/IT
eMail: joachim.mutter@sysarc.de

1 Einleitung

In einer verteilten Domino Umgebung wird aufgrund der Anforderungen der Replikation standardmäßig ein klassisches Hub- / Spoke Konzept angestrebt. Dabei sind dedizierte Server rein für die Replikation verantwortlich, damit die Wege und Richtungen der Replikation klar darstellbar und damit administrierbar sind. Dabei stößt man aber auf ein nicht triviales Problem hinsichtlich Domino Datenbanken, deren Backend-Datenverarbeitung über Agenten durchgeführt wird, die vom Domino Agentmanager als Zeit oder Event getriggerte Aktionen ausgeführt werden.

Das Prinzip der Ausführung dieser Agenten lässt sich hinsichtlich der Steuerung recht einfach eingrenzen. Wenn der Agent aktiviert ist, läuft er entweder auf einem dedizierten Server, dessen Name eine Eigenschaft des Agenten ist, oder diese Eigenschaft steht auf „Any Server“ oder auch „*“, womit der Agent auf jedem Server ausgeführt wird, auf dem die Datenbank existiert und der AMGR geladen ist. Es ist also nicht möglich, die Abarbeitung der Agenten auf mehrere dedizierte Server zu beschränken.
Diese Anforderung „Any Server“ ist aber oft bei verteilten Workflow Applikationen nötig, da jeder einzelne Server und damit Agent für eine definierte Datenmenge zuständig ist und somit auf unterschiedlichen Server laufen **muss.**

Der Betreiber muss aber aufgrund seiner Service-Level-Agreements (SLA) einen reibungslosen Betrieb dieser Server 24h * 7 Tage die Woche sicherstellen, was er aber im Prinzip nicht kann, da Agenten, für die er keine Verantwortung hat, unter Umständen frei auf diesen Servern laufen und dabei allerlei Schaden bis hin zu Serverabstürzen erzeugen können.
Nun ist es aber oft so, das verschiedene Agenten wegen Abrechungen, Management oder Monitoring Aufgaben trotzdem laufen müssen, also kann der AMGR nicht einfach abgeschaltet werden.

Es besteht also eine Anforderung, die Ausführung von Agenten auf diesen Server so einzugrenzen, dass nur bestimmte Agenten ausgeführt werden dürfen, aber alle anderen geblockt werden müssen.
Der hier beschriebene Domino Server Hook ist dazu in der Lage. Er steuert über Include oder Exclude Terme (Wildcards „*“ und „?“ sind hier möglich) die Ausführung von Agenten in den darüber definierten Datenbanken.

Dipl. Ing. Inf. Joachim Mutter
Waldstraße 4
66506 Maßweiler

Mobil: +49 (0) 172 / 7 38 35 25
Festnetz: +49 (0) 6334 / 983403
web: www.sysarc.de/IT
eMail: joachim.mutter@sysarc.de

2 Lösung

Es wurde ein Domino Server Hook implementiert, der sich in den Agentmanager (AMGR), der als Serverprozess ausgeführt wird, einklinkt und alle Requests zum Ausführen eines Agents prüft.

Diese Prüfung wird auf Datenbank Ebene durchgeführt. Dabei kann entweder über eine Komma separierte *Includeliste* Datenbanken oder Wildcards angegeben werden, deren Agenten laufen dürfen oder sie definieren eine Komma separierte *Excludeliste*, die alle Datenbanken oder Wildcards angeben, deren Agenten **nicht** ausgeführt werden dürfen. Somit wird die „wilde“ Ausführung von Agenten mit der Eigenschaft „Any Server“ durch den Domino Agentmanager auf diesen dedizierten Servern verhindert.

Die Applikation wird dazu über die Notes.ini als Extension-Manager Plugin (Server Hook) zur Ausführung gebracht. Sie stellt lediglich für den Domino Agentmanger die betreffenden Callback Routinen zur Verfügung, alle anderen Tasks des Servers und der Server selbst sind davon nicht betroffen. Konfiguriert werden nur eine Handvoll INI Variablen (Logging, einzuschließende oder auszuschließende Datenbanken). Dynamische Änderungen sind für einige Settings über Serverkonsolen Kommandos „Config Set var=value„ durchführbar.

Nach der Installtion und Inbetriebname des Hooks brauchen Sie sich dann keine Gedanken mehr zu machen, was auf Ihren Hub bzw. Spoke Servern, die unter anderem nur für Replizieraufgaben vorhanden sind, von wild laufenden Agenten angerichtet werden kann. Sie haben es nun in der Hand, welche Agenten noch laufen dürfen (Maintenance oder Report Aufgaben). Dazu bedarf es keinerlei Änderungen in dem Code der Datenbanken, wie zum Beispiel das Agent-Feld RunOnServer zu beschränken, oder anderer Maßnahmen, die eine Ausführung auf Code-Ebene verhindern, wobei das immer in der Hand des Entwicklers liegt und nicht in der des Providers

3 Verfügbare Systeme und Domino Versionen

Der Domino Server Hook wurde für die Betriebssysteme

- Windows W2K, XP
- Solaris 9 und Solaris 10
- Linux (Suse), aber auch alle anderen Distributionen dürften funktionieren

entwickelt und kompiliert. Es stehen also 3 Binär-Versionen zur Verfügung, die je nach System installiert werden müssen.

Der Hook wurde unter Domino 5,x und 6.x auf den jeweiligen Systemen einem Dauertest von 1 Monat unterzogen. Der Einsatz unter anderen Domino Versionen sollte jederzeit möglich sein, da keine spezifischen API-Calls verwendet wurden, es wird jedoch empfohlen, dem Einsatz eine ausführliche Testzeit unter der anderen Domino Version vorausgehen zu lassen.

Die Bereitstellung anderer Betriebssystem Versionen ist möglich, sofern eine Entwicklungsmaschine unter diesem System verfügbar ist. Dies bedarf aber gesondertem Aufwand.

Dipl. Ing. Inf. Joachim Mutter
Waldstraße 4
66506 Maßweiler

Mobil: +49 (0) 172 / 7 38 35 25
Festnetz: +49 (0) 6334 / 983403
web: www.sysarc.de/IT
eMail: joachim.mutter@sysarc.de

4 Parametrisierung und Installation

4.1 Installation

Das ExtensionManager Plugin muß im Notes Binary Verzeichnis, üblicherweise /opt/lotus/notes/latest/sunspa oder linux unter dem Namen libsarchook.so verfügbar sein.
Die Gruppenzugehörigkeit und der Besitzer müssen die gleichen sein, wie bei allen anderen Notes Libraries oder Tasks auch, also üblicherweise folgende:

-r-xr-xr-x 1 root bin libsarchook.so

4.2 Parameter

Der Hook wird über Einträge in der „notes.ini“ gesteuert. Hier muss der korrekte Registrierungscode angegeben werden und es können Debug Levels und Laufzeitparameter eingetragen werden, die das Verhalten des Hooks steuern.

Variablenname notes.ini	**Muss/Kann Feld, Default**	**Werte**
EXTMGR_ADDINS	Muss für ExtensionManager Betrieb	sarchook
SysArcHookLogOptions	Optional Default = 0	Debug: 25 Standardvorgabe 5 Mögliche Werte: 0x001 [1] Log Errors into LogFile and log.nsf if enabled 0x002 [2] Log Debug Log-Entries into log-gile 0x004 [4] Log defined Log-Entries into Log.nsf 0x010 [16] Log all Everything related to Agent Hooks 0x100 [256] Log all Databas Close events into log-file Der Debuglevel stellt die Addition der einzelnen Levels dar, also „Error Logging“ und „log.nsf Logging“ => 1 + 5 = 5
SysArcHookActions	Optional Default = 0	0 Die Ausführung des Hooks ist deaktiviert. Eine Änderung dieses Wertes in der Notes.Ini wirkt sofort 1 Der Hook ist aktiv und führt seine Aufgaben aus
SysArcExcludeDB	Optional Default = <Leer-String>	Komma separierte Liste von einzelnen Datenbank Pfaden oder Patterns (Wildcards * und ?), die eine bestimmte Menge an Datenbanken definieren. Agenten diese Datenbanken werden **nicht** ausgeführt, aber alle anderen.
SysArcIncludeDB	Optional Default = <Leer-String>	Komma separierte Liste von einzelnen Datenbank Pfaden oder Patterns (Wildcards * und ?), die eine bestimmte Menge an Datenbanken definieren. Agenten diese Datenbanken werden ausgeführt und alle anderen **nicht**.
SysArcRegister	Muss 16 Charakter lang	Der Registrierungscode wird aufgrund des (abbreviated) Servernamens berechnet oder er definiert einen Evaluierungszeitraum.

Zum Beispiel

```
EXTMGR_ADDINS=lsarchook
SysArcHookLogOptions=23                                  // DebugMode 23 , NormalMode 5
SysArcHookActions=1
SysArcExcludeDB=
SysArcIncludeDB=mail/*
SysArcRegister=2be9962e9d456bbc587b925105e3419b // Evaluierungsende : 01.02.2007
```

4.2.1 SysArcHookActions

0x0000 = ACTION_NO

Darüber kann der Hook disabled werden, ohne das der eigentliche ExtensionManager Eintrag EXTMGR_ADDINS entfernt wird. Somit ist es auch im laufenden System möglich, den Hook zu deaktivieren.

0x0001 = ACTION_DO

Dies ist die Default Einstellung. Der Hook ist für den Agentmanager aktiviert und verhindert das Ausführen von Agenten in Datenbanken durch den AMGR, die per Include oder Exclude Liste definiert sind.

Default ist 0, also deaktivierter Hook Betrieb.

4.2.2 SysArcExcludeDB

Agenten dieser hier definierten Datenbanken werden **nicht** ausgeführt, aber alle anderen.

Als Wert wird eine Komma separierte Liste von einzelnen Datenbank Pfaden oder Patterns (Wildcards * und ?) angegeben, die eine bestimmte Menge an Datenbanken definieren. Falls nur ein Wert vorhandne ist, dann entfällt das Komma.

Sie können hier also z.B.

SysArcExcludeDB=mail/*, names.nsf

angeben, was dazu führt, das kein Agent einer Datenbank, die im Verzeichnis „mail“ liegt, ausgeführt wird und zusätzlich kein Agent, der im Namens- und Adressbuch aktiviert ist.

Restriktion: Die Länge der Liste darf 255 Zeichen nicht überschreiten

Default ist der Leerstring, womit diese Liste deaktiviert wird

4.2.3 SysArcIncludeDB

Agenten dieser hier definierten Datenbanken werden ausgeführt, aber alle anderen **nicht**.

Als Wert wird eine Komma separierte Liste von einzelnen Datenbank Pfaden oder Patterns (Wildcards * und ?) angegeben, die eine bestimmte Menge an Datenbanken definieren. Falls nur ein Wert vorhandne ist, dann entfällt das Komma.

Sie können hier also z.B.

SysArcExcludeDB=monitoring/*, billing.nsf

angeben, was dazu führt, das die aktivierten Agenten aller Datenbanken, die im Verzeichnis „monitoring“ liegen, ausgeführt werden und zusätzlich die aktivierten Agenten der Datenbank „billing.nsf“.

Restriktion: Die Länge der Liste darf 255 Zeichen nicht überschreiten

Default ist der Leerstring, womit diese Liste deaktiviert bleibt.

Dipl. Ing. Inf. Joachim Mutter
Waldstraße 4
66506 Maßweiler

Mobil: +49 (0) 172 / 7 38 35 25
Festnetz: +49 (0) 6334 / 983403
web: www.sysarc.de/IT
eMail: joachim.mutter@sysarc.de

4.2.4 SysArcHookLogOptions

Alle Level sind in hexadezimaler und dezimaler Schreibweise aufgelistet!
Die Logfile ist im Notes Datenverzeichnis unter den Namen „sysarc_hook.log“ zu finden.

0x0001 [1 dezimal] = LOG_ERROR
Fehler und SARC Log Meldungen werden in die Logdatei geschrieben.

0x0002 [2 dezimal] = LOG_DEBUG
Sonstige Debug Meldungen wie das Registrieren und Deregistrieren der Hook Callbacks, Task Excluding etc. werden in die Logdatei geschrieben

0x0004 [4 dezimal] = LOG_AGENT
Alle Meldungen, die das Sperren bzw. Zulassen von Agenten betreffen.

0x0010 [16 dezimal] = LOG_NSF
Spezielle Logmeldungen werden in das Systemlog Log.nsf des Domino Servers geschrieben:
Registrieren und Deregistrieren der Hook Callbacks (auch Fehler)
Alle Meldungen, die Fehler darstellen, und zum Abschalten des Hooks führen.

Wenn dieser Level nicht gesetzt ist, wird keinerlei Meldung in das Systemlog log.nsf geschrieben.

0x0100 [256 dezimal] = LOG_CLOSE
Requests für das Schließen von Datenbanken (NSFDBClose) werden in die Logdatei geschrieben.

Die Log-Optionen sind additiv zu betrachten, das bedeutet, das jede Log-Kategorie zu dem resultierenden Log-Level addiert werden muss, damit sie aktiv ist.

Der normale Betriebswert wäre dann :
LogLevel = LOG_ERROR + LOG_DEBUG + LOG_AGENT + LOG_NSF
= 0x0001 + 0x0004 + 0x0100
= 1 + 2 + 4 + 16
= 23
=> folgender Eintrag müsste in der Notes.ini enthalten sein:
SysarcHookLogOptions=5

Der normale Betriebswert wäre dann :
LogLevel = LOG_ERROR + LOG_NSF
= 0x0001 + 0x0004
= 1 + 4
= 5
=> folgender Eintrag müsste in der Notes.ini enthalten sein:
SysarcHookLogOptions=5

Default ist 0, also kein Logging

4.2.5 SysArcRegister

Hier müssen Sie einen gültigen Registrierungsschlüsssel angebene, den Sie von uns erhalten. Dieser wird anhand des Servernamens generiert und ist nur für diesen definierten Server gültig. Ein von uns generierter Evaluierungsschlüssel ist immer zeitlich befristet, funktioniert in dieser Zeit aber genauso wie ein Serverschlüssel. Ist die Evaluierung abgelaufen, oder der Schlüssel ungültig, wird der Hook zwar geladen, bleibtr aber deaktiviert.