**FUNCIONAMIENTO DVK DAEMON (DVKD)**

***ALUMNOS*: ALDO JAVIER RAMONDA**

**Objetivo:** Desde Webmin ejecutar comandos o APIs del DVK remotamente en los nodos del cluster.

**FUNCIONAMIENTO PARA LA PRUEBA**

En sistema tiene 3 componentes:

* *MONITOR*: El cual está en C pero luego será realizado en PERL para ejecutarlo desde webmin. Se utilizan dos NAMED PIPES (INCMD, OUTCMD) para comunicarlo con el PROXY.
* *PROXY*: Se encarga de interfacear el MONITOR con los diferentes agentes. Convierte lo que recibe por la PIPE INCMD en un multicast hacia el AGENTE, y convierte el multicast recibido desde el AGENTE a un mensaje en el PIPE OUTCMD
* *AGENTES*: Estan a la espera de recibir un COMANDO o un API CALL desde el PROYX via spread. Responde por SPREAD.

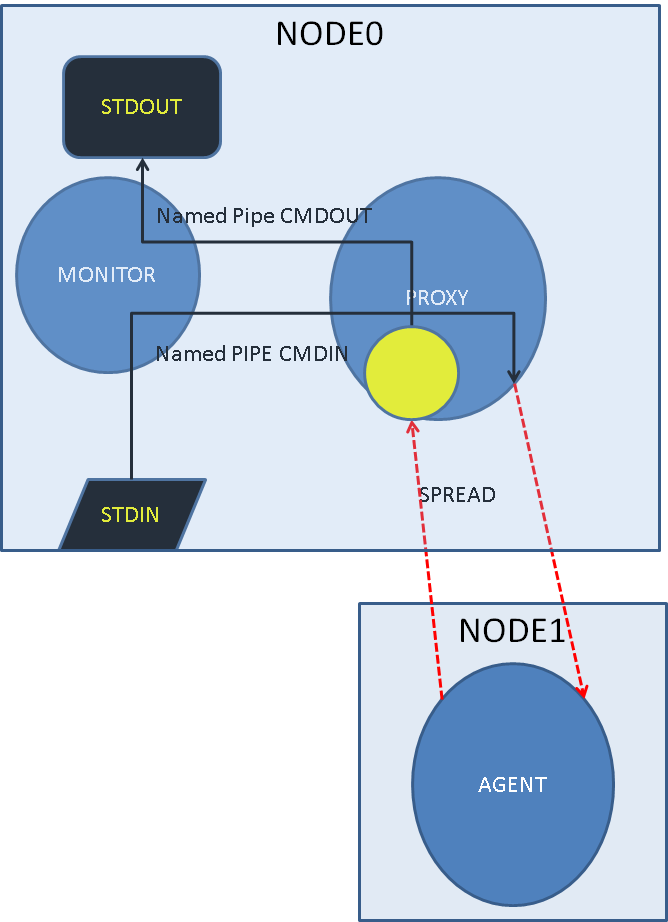
**INFORMACION PARA EL DESARROLLO**

|  |  |
| --- | --- |
|  | En el archivo /usr/src/dvs/include/com/ipc.h Se encuentran los formatos definidos para los mensajes, Si se agregan nuevos formatos de mensaje no deben superar el tamaño sizeof(message)=36. Al ejecutar el Monitor presenta en pantalla el tamaño de mensaje y los tamaños de los formatos que puede contener (max = 28 bytes)  Se dispone ademas de formatos preparados para el printf o USRDEBUG  #define MSGC\_FORMAT "source=%d type=%d mCi1=%d mCi2=%d mCi3=%d mCi4=%d mCca1=[%s]\n"  #define MSGC\_FIELDS(p) p->m\_source,p->m\_type, p->mC\_i1, p->mC\_i2, p->mC\_i3, p->mC\_i4, p->mC\_ca1  Y se usaria:  USRDEBUG(MSGC\_FORMAT, MSGC\_FIELDS(m\_ptr));  Imprimira formatos tipo MSGC. |
|  | En el archivo /usr/src/dvs/include/com/dvk\_calls.h estan las constantes figurativas para todas las DVK CALLS. |
|  | En el archivo /usr/src/dvs/include/com/stub\_dvkcalls.h estan las definiciones de los formatos de todas las DVK CALLS con sus parametros. |
|  | En el archivo /usr/src/dvs/include/com/dvk\_errno.h estan las las constantes figurativas para todos los codigos de error. |
|  | En el archivo /usr/src/dvs/dvs-apps/macros.h hay macros utiles para presentar codigos de error y hacer un seguimiento de los LOCKs y UNLOCKs de semaforos y WAIT y SIGNAL de variables de condicion |
|  | En el archivo /usr/src/dvs/dvs-apps/debug.h hay macros utiles para debugging, sobretodo USRDEBUG, que funciona igual que un printf, con la unica diferencia de que imprime mas informacion por ejemplo “**1741:dd\_monitor.c:do\_cmd\_input:167:”.** Da el PID=1741, el programa fuente, el nombre de la funcion y el nro de linea.  Anulando la constante figurativa USRDBG no se realiza mas DEBUGGING  // #define USRDBG 1 << **ANULADA COMO COMENTARIO** |

# DIAGRAMAS DE FLUJO



# PRUEBAS



|  |  |  |  |
| --- | --- | --- | --- |
| **SEQ** | **NODE0 (MONITOR)** | **NODE0(PROXY)** | **NODE1(AGENT)** |
| 3 |  | cd /usr/src/dvs/dvk-tests/ | cd /usr/src/dvs/dvk-tests/ |
| 4 |  | ./tests.sh 0 0 | ./tests.sh 1 0 |
| 5 |  | . /dev/shm/DC0.sh | . /dev/shm/DC0.sh |
| 6 |  | cd /usr/src/dvs/dvs-apps/dvkd | cd /usr/src/dvs/dvs-apps/dvkd |
| 7 |  | ./dd\_proxy >proxy.out 2>proxy.err & | ./dd\_agent >agent.out 2>agent.err & |
| 8 | ./dd\_monitor |  |  |

El MONITOR arroja los mensajes por pantalla porque require el ingreso de datos.

Tanto el PROXY como el AGENT dejan sus salidas STDOUT y STDIN para debugging en los archivos redireccionados.

SESION EN NODE0 (MONITOR)

root@node0:/usr/src/dvs/dvs-apps/dvkd# **./dd\_monitor**

1764:dd\_monitor.c:main:34:DVKD MONITOR: Initializing...

1764:dd\_monitor.c:main:35:DVKD MONITOR: sizeof(message)=36

1764:dd\_monitor.c:main:37:DVKD MONITOR: sizeof(mess\_1)=24

1764:dd\_monitor.c:main:38:DVKD MONITOR: sizeof(mess\_2)=24

1764:dd\_monitor.c:main:39:DVKD MONITOR: sizeof(mess\_3)=28

1764:dd\_monitor.c:main:40:DVKD MONITOR: sizeof(mess\_4)=20

1764:dd\_monitor.c:main:41:DVKD MONITOR: sizeof(mess\_5)=24

1764:dd\_monitor.c:main:42:DVKD MONITOR: sizeof(mess\_6)=28

1764:dd\_monitor.c:main:43:DVKD MONITOR: sizeof(mess\_7)=24

1764:dd\_monitor.c:main:44:DVKD MONITOR: sizeof(mess\_8)=24

1764:dd\_monitor.c:main:45:DVKD MONITOR: sizeof(mess\_9)=16

1764:dd\_monitor.c:main:46:DVKD MONITOR: sizeof(mess\_A)=24

1764:dd\_monitor.c:main:47:DVKD MONITOR: sizeof(mess\_B)=20

1764:dd\_monitor.c:main:48:DVKD MONITOR: sizeof(mess\_C)=28

1764:dd\_monitor.c:main:53:gethostname=node0

1764:dd\_monitor.c:main:61:Open MONITOR pipe=/tmp/OUTCMD\_node0 for READING

1764:dd\_monitor.c:main:70:Open MONITOR pipe=/tmp/INCMD\_node0 for WRITING

1764:dd\_monitor.c:do\_cmd\_input:98:

**(node0)Input command> ls**

1764:dd\_monitor.c:do\_cmd\_input:117:cmd\_input[ls] inbytes=2

1764:dd\_monitor.c:do\_cmd\_input:155:INCMD HEADER (36): source=1 type=101 m1i1=2 m1i2=0 m1i3=0 m1p1=(nil) m1p2=(nil) m1p3=(nil)

1764:dd\_monitor.c:do\_cmd\_input:167:INCMD: cmd\_input=ls

1764:dd\_monitor.c:do\_cmd\_input:186:OUTCMD HEADER: source=2 type=8293 m1i1=240 m1i2=2 m1i3=0 m1p1=(nil) m1p2=(nil) m1p3=(nil)

1764:dd\_monitor.c:do\_cmd\_input:191:Reading from 240 PIPE...

**agent.err**

**agent.out**

**dd\_agent**

**dd\_agent.c**

**dd\_agent.h**

**dd\_agent.o**

**dd\_common.h**

**dd\_monitor**

**dd\_monitor.c**

**dd\_monitor.h**

**dd\_monitor.o**

**dd\_proxy**

**dd\_proxy.c**

**dd\_proxy.h**

**dd\_proxy.o**

**dvkd.pptx**

**error.txt**

**glo\_agent.h**

**glo\_monitor.h**

**glo\_proxy.h**

**Makefile**

**user.c**

**(node0)Input command> hostname**

1764:dd\_monitor.c:do\_cmd\_input:117:cmd\_input[hostname] inbytes=8

1764:dd\_monitor.c:do\_cmd\_input:155:INCMD HEADER (36): source=1 type=101 m1i1=8 m1i2=1 m1i3=0 m1p1=(nil) m1p2=(nil) m1p3=(nil)

1764:dd\_monitor.c:do\_cmd\_input:167:INCMD: cmd\_input=hostname

1764:dd\_monitor.c:do\_cmd\_input:186:OUTCMD HEADER: source=2 type=8293 m1i1=6 m1i2=3 m1i3=0 m1p1=(nil) m1p2=(nil) m1p3=(nil)

1764:dd\_monitor.c:do\_cmd\_input:191:Reading from 6 PIPE...

**node1**

(**node0)Input command> API**

1764:dd\_monitor.c:do\_cmd\_input:117:cmd\_input[API] inbytes=3

(node0)API> rmtbind <dcid> <ep> <nodeid> <name>

**(node0)API> rmtbind 0 100 0 proc100**

1764:dd\_monitor.c:do\_getAPI:224: rmtbind

1764:dd\_monitor.c:do\_getAPI:235: 0

1764:dd\_monitor.c:do\_getAPI:235: 100

1764:dd\_monitor.c:do\_getAPI:235: 0

1764:dd\_monitor.c:do\_getAPI:235: proc100

1764:dd\_monitor.c:do\_getAPI:252:API HEADER (36): source=1 type=9 mCi1=2 mCi2=0 mCi3=100 mCi4=0 mCca1=[proc100]

1764:dd\_monitor.c:do\_cmd\_input:205:API REPLY (36): source=3 type=8201 mCi1=100 mCi2=0 mCi3=0 mCi4=0 mCca1=[rmtbind]

**API RESULT rmtbind =100**

**(node0)Input command> exit**

1764:dd\_monitor.c:do\_cmd\_input:117:cmd\_input[exit] inbytes=4

ERROR: dd\_monitor.c:do\_cmd\_input:121: rcode=0

ERROR: dd\_monitor.c:do\_cmd\_input:121: rcode=0

RESULTADO EN NODE1

root@node1:/usr/src/dvs/dvs-apps/dvkd# **cat /proc/dvs/DC0/procs**

DC pnr -endp -lpid/vpid- nd flag misc -getf -sndt -wmig -prxy name

0 100 100 -1/-1 0 1000 0 27342 27342 27342 27342 proc100