


Analyse System/Process Hangs

DECUS Symposium Bonn VMS 1L05

Volker.Halle@hp.com 20-APR-2004

Analyse System/Process Hangs 

- Benutzung von SDA
- Forced Crashes
- Forced Crash/Hang Beispiel
- Process Hangs
- Resource Wait States

Benutzung von SDA



- SDA = System Dump Analyzer
- Benutzung hauptsächlich durch HP Support und HP Engineering
- viele 'interessante' SHOW Befehle
- Einsatz durch System Manager:
Troubleshooting im laufenden System

Benutzung von SDA...



- Analyse von Crashdumps
- SDA im laufenden System (braucht CMKRNL)

```
$ ANALYZE/CRASH [node::]dumpfile  
SDA>
```

```
$ ANALYZE/SYSTEM  
SDA>
```

geeignet zur Analyse von Prozess-Hangs

Benutzung von SDA...



- *OpenVMS Alpha System Analysis Tool Manual*
- *OpenVMS VAX System Dump Analyzer Utility Manual*
keine SDA Erweiterungen (SDA> CLUE ...)
- SDA> HELP
- <http://h71000.www7.hp.com/doc/>


Die ersten SDA Befehle



- Was läuft im System ?

SDA> SHOW SUMMARY
Ausgabe ähnlich \$ SHOW SYSTEM
- Wie sieht das Cluster aus ?
SDA> SHOW CLUSTER
- usw... SDA> HELP SHOW
SDA> CLUE HELP

Die ersten SDA Befehle...



- Process-Kontext


```

SDA> SHOW SUMMARY
Current process summary
Extended  Indx  Process name  Username  State  Pri PCB/KTB  PHD/FRED  Wkset
-- PID --  ---  -
00000041 0001 SWAPPER           HIB      16 80C44D90 80C44800  0
00000044 0004 LANACP           SYSTEM   HIBO    13 80E0DD40 815EC000  27
00000046 0006 IPCACP           SYSTEM   HIB     10 80E1B3C0 815FE000  20
00000047 0007 ERRFMT           SYSTEM   HIB     8 80E1D7C0 81604000  27
00000048 0008 OPCOM           SYSTEM   HIB     8 80E1EAC0 8160A000  44
    
```

```

SDA> SET PROCESS/INDEX=nnnn
SDA> HELP SHOW PROCESS
    
```

Forced Crash



- System oder wichtige Prozesse hängen
- keine Zeit für lange Analyse im laufenden System
- ein paar schnelle Tests:
 - SHOW CLUSTER ?
 - PING ?
 - Console Terminal: <RETURN> - <BEL> ?
 - SET HOST oder TELNET Username: Prompt ?
Password: Prompt ?

Forced Crash...



- Crash forcieren mit:
 - CTRL/P oder HALT-Knopf (ACHTUNG: RESET !)
>>> CRASH
 - DECamds oder AvailabilityManager: Crash Node Fix
 - \$ DEFINE OPC\$REBOOT Y
\$ DEFINE OPC\$NODUMP N
\$ MC OPCCRASH

Forced Crash oder Hang



- Ähnliche Analyse-Techniken für Forced Crash oder im laufenden System (\$ ANA/SYSTEM)
- Wer oder was hängt ? Seit wann ?
- System-Prozesse NICHT in HIB ?
ERRFMT, OPCOM etc. sind normalerweise in HIB.
Verdächtig, falls in LEF !
- Prozesse in Resource Wait Zustand (RW*) ?
- SDA> CLUE XQP/ACTIVE - FileSystem aktiv ?

Forced Crash oder Hang...



- Vorgehensweise:
hängenden Prozess finden: SDA> SHOW SUMM
Prozess-Context setzen: SDA> SET PROC/IND=n

SDA> SHOW PROC/CHAN - busy ?
SDA> SHOW PROC/LOCK - 1. Lock Waiting ?

Forced Crash oder Hang...



- Seit wann hängt der Prozess ?

```
SDA> SHOW PROCESS
SDA> READ SYSDEF
SDA> EXA PCB+pcb$l_waitime
PCB+00098: 000000F7.142C6A5B
SDA> EXA exe$gl_abstim_tics
EXE$GL_ABSTIM_TICS: 00000000.142C8530
SDA> EVALUATE 142C8530 - 142C6A5B
Hex = 00000000.00001AD5  Decimal = 6869  (in Ticks)
        Prozess wartet seit 68,69 ^^^^ Sekunden
```

```
SDA> eva ((@exe$gl_abstim_tics)-@(pcb+pcb$l_waitime))
```

Forced Crash oder Hang...



- Tipp: Datenstruktur-Definitionen

```
SDA> FORMAT xxxxxxxx
```

```
%SDA-E-NOSYMBOLS, no "PCB" symbols found to format this block
```

```
SDA> READ SYSDEF.STB (Alpha)
```

```
SDA> READ SYS$SYSTEM:SYSDEF.STB (VAX)
```

```
$ SEARCH SYS$LIBRARY:LIB.REQ PCB$_WAITIME
```

```
macro PCB$_WAITIME = 152,0,32,0 %; ! Abs time of last process event
```

Forced Crash - Beispiel



```
Crash Time:      3-APR-2001 08:47:33.37
Bugcheck Type:  OPERCRASH, Operator forced system crash
Node:           HANG (Standalone)
CPU Type:       hp AlphaServer DS20E 500 MHz
VMS Version:    V7.2-1
Current Process: NULL
Current Image:  <not available>
Failing PC:     FFFFFFFF.9DB14D48
Failing PS:     00000000.000007D1
Module:         <not available>
Offset:         00000000

Boot Time:      14-MAR-2001 14:04:29.00
System Uptime:  19 18:43:04.37
```

Forced Crash - Ein Beispiel...




```

SDA> SHOW SUMMARY
Current process summary
-----
Extended  Indx Process name Username State  Pri PCB/KTB  PHD/FRED Wkset
-- PID --  ----  -----
00000401 0001 SWAPPER  SYSTEM HIB   16  84A3A3C0 84A39E00  0
0003D6DC 02DC_OPA0:  <login> LEF   4  81869FC0 85B20000  48
          ^^^

SDA> SHOW PROC/CHAN/IND=02DC
          ^^^
Channel Window      Status      Device/file accessed
-----  -----
0020 8111F2C0          DSA0:(20349,4,0) (section file)
0030 00000000      Busy      DSA0:
0040 00000000          OPA0:

SDA> SHOW DEVICE DSA0:      - Pending IOs ?
    
```

Forced Crash - Ein Beispiel...



```

SDA> SHOW PROC/LOCK

Lock id: 75003871          PID: 00F502DC  Flags: VALBLK SYNCSTS SYSTEM
Par. id: 01000019          SUBLCKs: 0      NOQUOTA NODLCKW
LKB: FFFFFFFF.7EBDD750  BLKAST: 00000000
Priority: 0000          RSEQNM: 000F

Waiting for PW 00000000-FFFFFFF

Resource: 000B7324 42313146 F11B$.. Status: ASYNC NOQUOTA
Length 10 00000000 00000000 .....
Kernel mode 00000000 00000000 .....
System 00000000 00000000 .....

WICHTIG: 1. Lock im WAITING State ?
    
```


Forced Crash - Ein Beispiel...



```
SDA> SHOW RESOURCE/LOCK= 75003871
```

```
Resource Database
```

```
-----
```

```
...
```

```
Granted queue (Lock ID / Gr mode / Range):
```

```
3E0013B5 PW 00000000-FFFFFFFF 180009D0 NL 00000000-...
```

```
Conversion queue (Lock ID / Gr mode / Range -> Rq mode / Range):
```

```
*** EMPTY QUEUE ***
```

```
Waiting queue (Lock ID / Rq mode / Range):
```

```
70002FF2 PW 00000000-FFFFFFFF 75003871 PW 00000000-...
```

WICHTIG: Inkompatibler Lock-Request in Granted State

Forced Crash - Ein Beispiel...



```
SDA> SHOW LOCK 3E0013B5
```

```
...
```

```
PID: 00D5013F
```

```
SDA> SET PROC/IND=00D5013F
```

```
SDA> SHOW PROCESS
```

```
...
```

Damit ist der Prozess gefunden, dem der PW-Lock gehört, der den anderen PW-Lock blockiert

Wie lange hängt dieser Prozess ?

```
SDA> eva ((@exe$gl_abstim_tics)-@(pcb+pcb$I_waitime))
```

```
Hex = 00000000.00325999 Decimal = 3299737 < über 9 Std.
```

Forced Crash - Ein Beispiel...



- Es wurde der Prozess gefunden, der das Login auf OPA0: blockiert durch inkompatiblen Lock.
- Jetzt geht es wieder von vorne los: warum hängt dieser Prozess (seit über 9 Stunden) ?
SDA> SHOW PROC/CHAN - busy ?
SDA> SHOW PROC/LOCK - 1. Lock Waiting ?
- Systematische Vorgehensweise und Geduld sind notwendig !

Process Hangs



- SHOW SYSTEM/STATE=...
- Resource Wait States RWxxx
- MUTEX Wait State MUTEX
- Outswapped State xxxO
- Free Page Wait FPG
- Local Event Flag LEF
- CLUE MEM/STAT - Pool alloc failures ?
- CLUE MEM/FILES - Page/Swap Files voll ?

Resource Wait States



- SDA> SHOW PROC/INDEX=xxx
- Event Flag Wait Mask (PCB\$L_EFWM , KTB\$L_EFWM)
 - Resource RSN\$_xxx - Integer < 21
Werte siehe \$ SEARCH SYS\$LIBRARY:LIB.REQ RSN\$_
 - JIB Adresse (Job Quota Wait - SDA> FORMAT JIB)
 - MUTEX Adresse
Holder hat Mutex Count PCB\$L_MTXCNT > 0 und Prio 16

Resource Wait States ...



- RWAST AST wait
- RWMBX Mailbox full
- RWNPG nonpaged pool
- RWPFF pagefile full
- RWPAG paged pool
- RWMPB modified page writer busy
- RWSCS distributed lock manager wait
- RWCLU cluster state transition / lock remastering
- RWCAP CPU capability
- RWCSV VMScluster Server Process
- RWINS inner mode semaphore (kernel threads)

Resource Wait Ressourcen



- Natural Language Search Assistant (ehemals Ask Compaq)
– <http://askq.compaq.com/>
- OpenVMS FAQ
– http://h71000.www7.hp.com/wizard/openvms_faq.html
- OpenVMS Ask The Wizard
– <http://h71000.www7.hp.com/wizard/>

RWAST - Ein Beispiel




- Problem: Beim Shutdown des WWW_server_80 Prozesses blieb der Prozess mit RWAST haengen.
- \$ STOP/ID=pid hilft nicht
- Alternative: Reboot des Systems oder Analyse des RWAST Status und Abhilfe mit DELTA Debugger.

```
$ SHOW SYSTEM/STATE=R*
```

```
OpenVMS V7.2-2 on node SMART1 9-JUL-2003 15:47:14.77 Uptime 2 02:15:58
  Pid   Process Name   State  Pri  I/O    CPU    Page flts  Pages
2040085D WWW_server_80  RWAST  6   47928  0 00:01:54.61  17757   36 N
```

```
RWAST - Ein Beispiel   
SDA> SHOW PROC/IND=2040085D  
  
Process index: 005D Name: WWW_server_80 Extended PID: 2040085D  
-----  
-----  
Process status:      00240003 RES,DELPEN,PHDRES,NETWRK  
...  
Master internal PID  0001005D Subprocess count      0  
# open files allowed left  492 Direct I/O count/limit  15000/15000  
UIC      [00100,000162] Buffered I/O count/limit  14999/15000 <<  
Abs time of last event  01149548 BUFIO byte count/limit  297248/297440 <<  
...  
  
SDA> SHOW PROC/CHAN  
                Process active channels  
Channel Window      Status      Device/file accessed  
...  
-----  
0280-00000000      Busy      NET4411:
```

```
RWAST - Ein Beispiel   
SDA> show call  
  
Call Frame Information  
-----  
Stack Frame Procedure Descriptor  
Flags: Base Register = FP, No Jacket, Native  
Procedure Entry: FFFFFFFF.800DE2A0          SYS$QIOW_C  
Return address on stack = FFFFFFFF.800D5EE8  
EXE$DASSGN_C+00188  
  
Registers saved on stack  
-----  
7FFA1890 00000000.00000280 Saved R2 <<< busy CHANNEL NET4411:  
7FFA1898 00000000.7FFA1900 Saved R29
```

RWAST - Ein Beispiel

```

SDA> SHOW DEV NET4411:

NET4411                Unknown          UCB: 837B1C80
Device status: 00010010 online,deleteucb
Characteristics: 0C1C2000 net,avl,mnt,mbx,idv,odv
Owner UIC [000100,000162] Operation count    2 ORB address 8392E340
                        PID 0001005D Error count    0 DDB address 817B2080
Class/Type              00/00 Reference count  1 DDT address C4ECA1E0
...
SDA> sea FFFFFFFF.81068000 : FFFFFFFF.92E82000 837B1C80 <<< UCB Adresse
Searching from FFFFFFFF.81068000 to FFFFFFFF.92E82000 in LONGWORD steps for
837...
Match at FFFFFFFF.831A2064 IRP$_UCB in IRP at 831A2040 <<< nur 1
Referenz
Match at FFFFFFFF.83791440 UCB$_LINK in UCB at 83791400
Match at FFFFFFFF.837B1C80 UCB$_FQFL in UCB at 837B1C80
Match at FFFFFFFF.837B1C84 UCB$_FQBL in UCB at 837B1C80
Match at FFFFFFFF.83972E80 SUD$PS_UCBLINK in SUD at 83972E80
5 matches found
    
```

RWAST - Ein Beispiel

```

SDA> format 7FF1A4E0/type=ccb ! Busy channel to NET4411:
00000000.7FF1A4E0 CCB$_UCB 837B1C80 UCB
00000000.7FF1A4E4 CCB$_STS 00000000
00000000.7FF1A4E8 CCB$_IOC 00000001 <<<
00000000.7FF1A4EC CCB$_DIRP 00000000
00000000.7FF1A4F0 CCB$_AMOD 04
00000000.7FF1A4F1 00000000
00000000.7FF1A4F4 CCB$_WIND 00000001 <<<
00000000.7FF1A4F8 CCB$_CHAN 00000280 <<<
                        CCB$_W_CHAN
    
```

RWAST - Ein Beispiel



```

SDA> SHOW POOL/NONPAGED/HEADER ...

[List]          FFFFFFFF.831A1E00    576    ...
IRP             FFFFFFFF.831A2040    576    ...
[List]          FFFFFFFF.831A2280    896    002004AF ...
    
```

Ausgangssituation:

- Process WWW_server_80 in RWAST und DELPEN
- 1 Busy channel zu NET4411:
- CCB zeigt 1 Outstanding IO
- UCB zeigt 1 Reference Count
- 1 BUFIO outstanding
- IRP ist noch im Pool (nicht freigegeben)

RWAST - Ein Beispiel



- Massnahmen zum Beenden des Prozesses:

```

$ run sys$share:delta
OpenVMS Alpha DELTA Debugger

Exit 00000001

8008E8B4!   LDQ           R28,#X0008(SP)   0001005D:1;m
00000001

7FF1A4E8/00000001   0           ! Loeschen von CCB$_IOC
7FF1A4F4/00000001   0           ! Loeschen von CCB$_WIND

81CA23E0/00003A97  3A98       ! Korrigieren von PCB$_BIOCNT
exit
    
```

RWAST - Ein Beispiel



```
$ SHOW SYS/STATE=RW*  
$
```

Es hat funktioniert...

Voraussetzungen:

- gruendliche Analyse und Risikoabwaegung
- vorheriges Testen der DELTA-Befehle

Use at your own risk ;-)

Analyse System/Process-Hangs



- Ein kleiner Einblick in die Möglichkeiten mit SDA
- Einblick in Hang-Analyse
- Ein RWAST Beispiel

```
SDA> EXIT
```