

## Crash-Analyse mit SDA

DECUS Symposium Bonn      VMS 2A07

Volker.Halle@compaq.com 17-APR-2002

## Crash-Analyse mit SDA

- Benutzung von SDA
- Crash Context
- INVEXCEPTN Beispiel
- Inline Crashes
- CNXMGRERR Beispiel
- SMP Crashes
- Forced Crashes
- Forced Crash/Hang Beispiel

## Was Sie immer schon über Crashes wissen wollten...

- Einfache und komplizierte Crashes
- Crash-Analyse: automatisch bis unmöglich
  
- Ein Versuch der Darstellung:  
    grundsätzliche Vorgehensweise  
    einige Beispiele

## Benutzung von SDA

- SDA = System Dump Analyzer
- Benutzung hauptsächlich durch Compaq Support und Compaq Engineering
- OpenVMS Internals Kenntnisse sehr hilfreich
- Zugriff auf OpenVMS Source Listings
- viele 'interessante' SHOW Befehle
- Einsatz durch System Manager:  
    Status System Crash (SDA> CLUE CRASH)  
    Troubleshooting im laufenden System

## Benutzung von SDA...

- Analyse von Crashdumps
  - \$ ANALYZE/CRASH [node::]dumpfile
  - SDA>
- SDA im laufenden System (braucht CMKRNL)
  - \$ ANALYZE/SYSTEM
  - SDA>
  - Einige Befehle sind nicht möglich im laufenden System (z.B. CLUE CRASH)
  - geeignet für Prozess-Hangs

## Benutzung von SDA...

- SDA Erweiterungen (SDA Extensions)
  - \$ DIR SYS\$SHARE:\*\$SDA.EXE
  - CLUE\$SDA.EXE - CLUE Befehle SDA>CLUE ...
  - NET\$SDA.EXE - DECnet/Plus SDA>NET ...
  - PTHREAD\$SDA - DECthreads SDA>PTHREAD...
  - Keine Abkürzung der Befehle (z.B: SDA> CLU ...) möglich
- SYS\$EXAMPLES:MBX\$SDA.C Beispiel

UNSUPPORTED !

## Benutzung von SDA...

- *OpenVMS Alpha System Analysis Tool Manual*
- *OpenVMS VAX System Dump Analyzer Utility Manual*
  - keine SDA Erweiterungen (SDA> CLUE ...)
- SDA> HELP
- Im Internet unter: [www.compaq.de/support](http://www.compaq.de/support)
  - > Referenz Bibliothek
  - > OpenVMS OS

## Die ersten SDA Befehle

- Das Wichtigste auf einen Blick:
  - SDA> CLUE CRASH
- 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 -- PID --	Indx	Process name	Username	State	Pri	PCB/KTB	PHD/FRED	Wkset
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

## Crash Context

Verschiedene Arten von Crashes:

- Exception Crashes

INVEXCEPTN, SSRVEXCEPT, UNXSIGNAL,  
FATALEXCEPT, PGFIPLHI

‘unerwartete‘ Exception im Kernel Mode

meist Access Violation (ACCVIO) bei Zugriff auf  
ungültige bzw. nicht erlaubte virtuelle Adressen

## Crash Context...

Verschiedene Arten von Crashes...

- Inline Crashes
  - programmierter Konsistenz-Check
  - INCONSTATE, CLUEXIT, SHADDETINCON,
  - CNXMGRERR, LOCKMGRERR,
  - CPUSPINWAIT, MACHINECHK,...
- RESTART Crashes
  - CPU HALT und >>> AUTO\_ACTION RESTART
  - HALT, KRNLSTAKNV, OPERCRASH,...

## Crash Context...

- Software oder Hardware ?
  - MACHINECHK, IOMACHINECHK, MCHECKPAL**
  - meistens Hardware, aber auch Software-Ursache
  - möglich (Driver)
  - CLUE File enthält KEINE zusätzliche Informationen

Immer ERROR LOG analysieren  
mit DECevent oder Compaq Analyze ...

## Crash Context...

- Errorlog Daten aus System Dumpfile extrahieren:

- OpenVMS Alpha

```
$ ANA/CRASH SYS$SYSTEM:SYSDUMP.DMP
```

```
SDA> clue errlog
```

```
Sequence    Date        Time
```

```
-----
```

```
33565 10-NOV-2000 10:36:48.14
```

```
33566 10-NOV-2000 11:13:07.52 * Crash Entry
```

```
SDA> EXIT
```

```
$ DIAGNOSE CLUE$ERRLOG
```

## Crash Context...

- Errorlog Daten aus System Dumpfile extrahieren:

- OpenVMS VAX

```
$ CLUE==$CLUE
```

```
$ CLUE/ERROR_LOGS SYS$SYSTEM:SYSDUMP.DMP
```

```
...
```

```
Error Log entries will be written to CLUE_ELOG.SYS -- use ANAL/ERR to display
```

```
Entry Type : 64 Seq # : 139 Time : 7-JAN-1998 03:20:55.05
```

```
Entry Type : 64 Seq # : 140 Time : 7-JAN-1998 03:20:56.12
```

```
...
```

```
$ DIAGNOSE CLUE_ELOG.SYS
```

ACHTUNG: CRD Summary Errlog-Entries beim Shutdown ignorieren !

## Crash Context...

- Software oder Hardware ... ?

### **CPUSANITY, CPUSPINWAIT, CPUCEASED**

sowohl Hardware (CPU) oder Software möglich  
 CLUE CRASH gibt zusätzliche Information bei  
 CPUSPINWAIT Crashes

### **alle anderen Crashes**

meistens Software  
 detaillierte Crash-Analyse notwendig  
 Errorlog-Check schadet nicht (10-15 min. vor Crash)

## Crash Context...

- \$ HELP/MESSAGE bugcheckname  
 liefert bei einigen Crashes Kontext-Informationen  
 und Vorschläge für die weitere Analyse
- SDA> CLUE CRASH  
 Immer der erste Schritt:  
 \$ ANAL/CRASH SYS\$SYSTEM:SYSDUMP.DMP  
 SDA> READ/EXEC [/NOLOG]  
 SDA> CLUE CRASH

## Crash Context...

- Process oder System Context ?

SDA> eva/ps @ps

```

MBZ SPAL  MBZ  IPL VMM MBZ CURMOD INT PRVMOD
0   00 00000000 00 0 0   KERN  0  USER

```

- Interrupt Stack (INT = 1) = System Context  
Current Process und Image meist irrelevant
- Process Stack (INT = 0) = Process Context  
Current Process und Image oft relevant

## Crash Context - Ein Beispiel

- SDA> CLUE CRASH

```

Crash Time:      12-MAR-2001 12:23:53.21
Bugcheck Type:  SSRVEXCEPT, Unexpected system service exception
Node:           HERO (Cluster)
CPU Type:       AlphaServer 8400 5/440
VMS Version:    V7.2-1
Current Process: TCPIP$INET_ACP
Current Image:   $4$DKA0:[SYS0.SYSEXEXE]TCPIP$INETACP.EXE;1
Failing PC:     00000000.00032ED0  TCPIP$INETACP+32ED0
Failing PS:     18000000.00000003
Module:         TCPIP$INETACP
Offset:         00032ED0

```

## Crash Context - Ein Beispiel...

### ▪ CLUE CRASH...

Boot Time: 10-MAR-2001 11:44:10.00  
 System Uptime: 2 00:39:43.21  
 Crash/Primary CPU: 01/00  
 System/CPU Type: 0C05  
 Saved Processes: 97  
 Pagesize: 8 KByte (8192 bytes)  
 Physical Memory: 5120 MByte (655360 PFNs, contiguous memory)  
 Dumpfile Pagelets: 347794 blocks  
 Dump Flags: olddump,writecomp,errlogcomp,dump\_style  
 Dump Type: compressed,selective  
 EXE\$GL\_FLAGS: poolpging,init,bugdump  
 Paging Files: 1 Pagefile and 1 Swapfile installed

## Crash Context - Ein Beispiel...

### ▪ CLUE CRASH...

General Registers:  
 R21 = 00000000.35383950 ...

Signal Array:  
 Arg Count = 00000005  
 Condition = 0000000C <<< ACCVIO  
 Argument #2 = 00000000 <<< Reason mask (Data Read)  
 Argument #3 = 35383950 <<< Failing Virtual Address  
 Argument #4 = 00032ED0 <<< Failing PC  
 Argument #5 = 00000003 <<< Failing PS  
 Failing Instruction:  
 TCPIP\$INETACP+32ED0: LDL R22,(R21)

## Crash Context - Ein Beispiel...

- CLUE REGISTER...

Pointer zu Datenstrukturen werden automatisch analysiert und decodiert:

R0 = 00000000.00130031

...

R3 = FFFFFFFF.814102E0 NET

R4 = FFFFFFFF.815A5C80 UCB (Device BG1236:)

...

R12 = FFFFFFFF.8161F0C0 UCB (Device TNA172:)

...

PC = 00000000.00032ED0

PS = 18000000.00000003 Kernel Mode, IPL 0

## Crash Context - Ein Beispiel...

- Zusammenfassung:

SSRVEXCEPTN, im Process Context,

Failing PC im P0-Space von TCPIP\$INETACP

- Was ist das Problem ?

- ACHTUNG: es ist selten so einfach !

**INVEXCEPTN Beispiel:**

Bugcheck Type: INVEXCEPTN, Exception while above ASTDEL  
 CPU Type: COMPAQ AlphaServer DS10 466 MHz  
 VMS Version: V7.2-1  
 Current Process: NULL  
 Current Image: <not available>  
 Failing PC: FFFFFFFF.803841A0 SYS\$IIDRIVER+021A0  
 Failing PS: 20000000.00000804  
 Module: SYS\$IIDRIVER (Link Date/Time: 29-DEC-1999 04:07:28.30)  
 Offset: 000021A0

**INVEXCEPTN Beispiel...**

```

SDA> eva/psl 20000000.00000804
MBZ SPAL  MBZ  IPL VMM MBZ CURMOD INT PRVMOD
 0  00 000000000000 8  0  0  KERN  1  KERN
Signal Array:
Arg Count  = 00000005
Condition  = 0000000C <<< Access Violation
Argument #2 = 00000000 <<< Reason mask (Data Read)
Argument #3 = 00000024 <<< Virtual Address
Argument #4 = 803841A0 <<< Failing PC
Argument #5 = 00000804 <<< Failing PS  IPL 8.
Failing Instruction:
SYS$IIDRIVER+021A0:  LDL          R5,#X0024(R16)
R16 = 00000000.00000000
R18 = FFFFFFFF.80CF6880  UCB (Device IIA0:)
  
```

## INVEXCEPTN Beispiel...

- Problembeschreibung aus Patch Kit:

PROBLEMS ADDRESSED IN VMS721\_IIDRIVER-V0100 KIT:

- o The system can crash with an INVEXCEPTN bugcheck at SYS\$IIDRIVER+021A0.

Images Affected: [SYS\$LDR]SYS\$IIDRIVER.EXE

## INVEXCEPTN Beispiel...

- Parameter der CCAT Rule CCAT-V-A-2120

Bugcheck: INVEXCEPTN

Condition Code: 0000000C

Failing Instruction: LDL

Failing Module Name: SYS\$IIDRIVER

Failing Module Offset: 000021A0

Virtual Address: 00000024

OS Version: V7.2-1, V7.2-1H1

Map Module: SYS\$IIDRIVER

## Inline Crashes

- Konsistenz-Prüfung von Daten/Zustand
- Typische Bugchecks:  
INCONSTATE, SHADDETINCON, XQPERR,  
LOCKMGRERR, CNXMGRERR, CLUEXIT, ...
- PC = Code, der die Inkonsistenz bemerkt
- Analyse (fast) unmöglich ohne Source Listings
- Context-Information über Register-Inhalte

## Inline Crashes...

Ein paar 'populäre' Inline-Crashes:

- CLUEXIT  
Connectivity im Cluster (meistens Netzwerk)  
RECNXINTERVAL überprüfen/erhöhen
- XQPERR  
File System (XQP)  
Disk-Caching Tools. Defragmenter. Pool Corruption
- INSF\_NONPAGED  
NPAGEDYN erhöhen. AUTOGEN FEEDBACK.

## Inline Crashes...

Noch mehr 'populäre' Inline-Crashes:

- SHADDETINCON

Shadowing.

Errorlog überprüfen. SDA> CLUE REGISTER oder

SDA> SHOW DEV/ADDR=@R5 zeigt Device

- INCONSTATE

meistens im Driver. Errorlog überprüfen.

SDA> CLUE CRASH zeigt Module/Driver

## CNXMGRERR - Beispiel

Bugcheck Type:

CNXMGRERR, Error detected by VMScluster Connection Manager

CPU Type: Compaq AlphaServer ES40

VMS Version: V7.2-1

Current Process: NULL

Current Image:

Failing PC: FFFFFFFF.801DDE48 CNX\$RCV\_MSG\_C+003F8

Failing PS: 04000000.00000804

Module: SYS\$CLUSTER (Link Date/Time: 6-SEP-2000 17:29:17.67)

Offset: 0000DE48

Failing Instruction:

CNX\$RCV\_MSG\_C+003F8: BUGCHK

## CNXMGRERR - Beispiel...

- SDA> CLUE REGISTER

```
R3 = FFFFFFFF.8161D200 CLU_CSB          <<< cluster member
R4 = FFFFFFFF.81543070 SCS_PDT         <<< SCS port
R6 = FFFFFFFF.8569BD58 SYS$PJDRIVER+14080 <<< SCS port driver DSSI
```

- SDA> FORMAT ...

```
SDA> FORMAT 8161D200
```

```
...
```

```
8161D24C: CSB$L_CSID 00010002
```

```
...
```

```
SDA> SHOW CLUSTER/CSID=00010002
```

```
--- HAOVSB Cluster System Block (CSB) 8161D200 ---
```

## CNXMGERR - Beispiel...

- weitere Analyse nur mit OpenVMS Source Listings möglich
- dieses Problem kann Hardware-Ursachen haben (Data-Korruption auf SCS Communication Path)
- Information von SDA> CLUE REGISTER gibt Aufschluss über SCS Communication Path

## SMP Crashes (Multi-Processing)

- CPUs überprüfen sich gegenseitig  
CPUSANITY Crash, wenn CPU hängt
- Synchronization von Daten über SPINLOCKS  
CPUSPINWAIT bei Überschreitung von Timeout  
SMP\_SPINWAIT (default: 1 Sekunde, IPL > 8)  
SMP\_LNGSPINWAIT (default: 30 Sekunden)

## SMP Crashes (Multi-Processing)...

- CPUSANITY  
SDA> SHOW CRASH ->CPU failed to service bugcheck req  
SDA> CLUE CONFIG  
...  
CPU ID 01                    CPU State rc,pa,pp,cv,pv,pmv,pl  
PAL Code 1.20-3            Halt PC    FFFFFFFF.8012A284    <<<  
CPU Revision ...            Halt PS    00000000.00000000  
Serial Number .....        Halt Code "Kernel Stack not Valid" <<<  
Console Vers V5.3-14        Halt Request "Default, No Action" <<<  
  
>>> SET AUTO\_ACTION RESTART  
hätte einen KRNLSTAKNV Restart-Crash erzeugt

## SMP Crashes (Multi-Processing)...

### ▪ CPUSPINWAIT

SDA> CLUE CRASH liefert mehr Informationen

CPUSPINWAIT Bugcheck:

Cause: timeout acquiring spinlock

Spinlock name: IOLOCK8/SCS

Spinlock address: 891AD700 <<< immer in Register R14

Spinlock owner CPU Id: 00

Crash CPU Id: 01

CPU Id	CPUDB	BugCode	State	WorkReq	Int PC
-----	-----	-----	-----	-----	-ab V7.3-
00	816CE000	CPUEXIT	Run	-	xxxxxx+nnnn
01	B8F84000	CPUSPINWAIT	Stopped	-	

## SMP Crashes (Multi-Processing)...

### ▪ CPUSPINWAIT...

SDA> SET CPU 01 <= Spinlock Owner CPU !

SDA> SHOW STACK

...

SP => AD573AF8 00000000 00001F04

...

SP+D8 => AD573BF0 FFFFFFFF AE324B70 UCX\$BGDRIVER+38B70

(V6.2 +F8) AD573BF8 00000000 00000804 <<< PS ^ PC

...

aktiver Instruction-Stream zum Crash-Zeitpunkt !!!

## 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
  - \$ DEFINE OPC\$REBOOT Y  
   \$ 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> 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 SYS$LOADABLE_IMAGES:SYSDEF.STB
SDA> READ SYS$SYSTEM:SYSDEF.STB      (VAX)

$ SEARCH SYS$LIBRARY:LIB.REQ  PCB$l_WAITIME
macro PCB$l_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:       COMPAQ 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 RQSEQNM: 000F

**Waiting for PW** 00000000-FFFFFFFF

Resource: 000B7324 42313146 **F11B\$s**.. 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 580025A9 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.
- 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 !

## Crash-Analyse mit SDA

- Ein kleiner Einblick in die Möglichkeiten mit SDA
- Übersicht über System-Crashes
- Einblick in Crash-Analyse

SDA> EXIT

**Fragen ?**

**COMPAQ**

**Volker.Halle@compaq.com**