

# Vorgehen Troubleshooting



# Houston we have a problem

- Stress
- Management ist involviert
- Druck irgendetwas zu tun – vorschnelle Aktionen
- Viele Beteiligte mit Halbwissen – viele Vermutungen
- “Ich habe nichts gemacht” - Syndrom

# Am Unfallort

- 1) Wichtigsten Daten sichern! (Notfallplan vorhanden?)
- 2) Funktion wiederherstellen (z.B. Neustart)
- 3) Problemanalyse in die Wege leiten

# Problemanalyse: Wissenschaftlicher Ansatz

- 1) These über das Problem aufstellen
- 2) Test für These definieren
- 3) Test durchführen
- 4) Weiter bei 1) mit verfeinerter Fragestellung

# Gängige Problemtypen

- Speicher geht aus
  - Memory-Leak oder Speicherhunger
- Threads blockieren
- Anwendung benötigt zu viel CPU
- Fachliche Probleme (Programmfehler)

# Grobe Einschränkung Performanceprobleme

CPU - Verbrauch Heap	gering	hoch
konstant	<p>Der Server wartet auf irgendetwas. Auf eingene Threads, Antworten von Fremdsystemen, wie z.B. Datenbank oder Webservices. Vermutung: Viele Threads sind Blockiert → Problem: <b>Blockierte Threads</b></p>	<p>Server ist mit abarbeiten eigenen Codes beschäftigt. Vermutung: Viele aktive Threads bzw. ein Thread der viel CPU benötigt. → Problem: <b>CPU-Fresser</b></p>
ansteigend bzw. am Maximum	<p>Bei Multiprozessor-Rechnern kann sein dass nur ein Prozess mit Garbage Collection (GC) beschäftigt ist. Vermutung: GC-Zeiten sind lange und häufig, viele/nur Major Collections → Problem: <b>Memory Verbrauch</b></p>	<p>CPU-Zeit wird mit GC verbraucht. Vermutung: GC-Zeiten sind lange und häufig, viele/nur Major Collections → Problem: <b>Memory Verbrauch</b></p>

# CPU / Memory-Verbrauch Host

## ■ Allgemein:

- vmstat
- iostat
- top
- sar

```
[esiegl@devjava4 ~]$ vmstat 5
```

```
procs  -----memory-----  --swap--  -----io-----  -system--  -----cpu-----
r  b    swpd    free    buff    cache    si    so    bi    bo    in    cs    us    sy    id    wa    st
0  0        0  3886200  414880  2165180    0    0    3    9    49    65    0    0    99    0    0
0  0        0  3885216  414880  2165180    0    0    0    6   667   413    4    0    96    0    0
0  0        0  3884904  414888  2165216    0    0    0  4750   353   235    3    0    97    0    0
0  0        0  3883632  414896  2165216    0    0    0   120   340   305    4    0    96    0    0
0  0        0  3883508  414896  2165216    0    0    0   170   236   192    2    0    98    0    0
```

```
Tasks: 246 total,   1 running, 245 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2,0 us,  0,2 sy,  0,0 ni, 97,8 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 8167276 total, 3874956 free, 1711800 used, 2580520 buff/cache
KiB Swap: 2097148 total, 2097148 free,    0 used. 6335372 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
23159	esiegl	20	0	5019832	764128	25968	S	7,9	9,4	0:35.92	java
1176	root	20	0	162756	13196	9668	S	0,3	0,2	3:10.65	vmtoolsd
21290	root	20	0	0	0	0	S	0,3	0,0	0:00.01	kworker/3:0
23531	esiegl	20	0	153572	4072	2804	S	0,3	0,0	0:00.01	sshd
23700	esiegl	20	0	60872	4912	4092	R	0,3	0,1	0:00.04	top
1	root	20	0	128792	8908	5744	S	0,0	0,1	0:07.27	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.10	kthreadd

```
[esiegl@devjava4 ~]$ sar -r 5
```

```
Linux 4.3.5-300.fc23.x86_64 (devjava4.gepardec.com)      2018-02-27      _x86_64_      (4 CPU)
```

	kbmemfree	kbmemused	%memused	kbbuffers	kbcached	kbcommit	%commit	kbactive	kbinact	kbdirty
21:49:25										
21:49:30	3142492	5024784	61,52	537052	1929688	5176536	50,43	3077936	1458800	12
21:49:35	3142492	5024784	61,52	537052	1929688	5176536	50,43	3077936	1458800	16

# Java Memory

## ■ jstat -gc

```
jstat -gc -t `eap7 pid` 10s | tee tmpmen.csv
```

