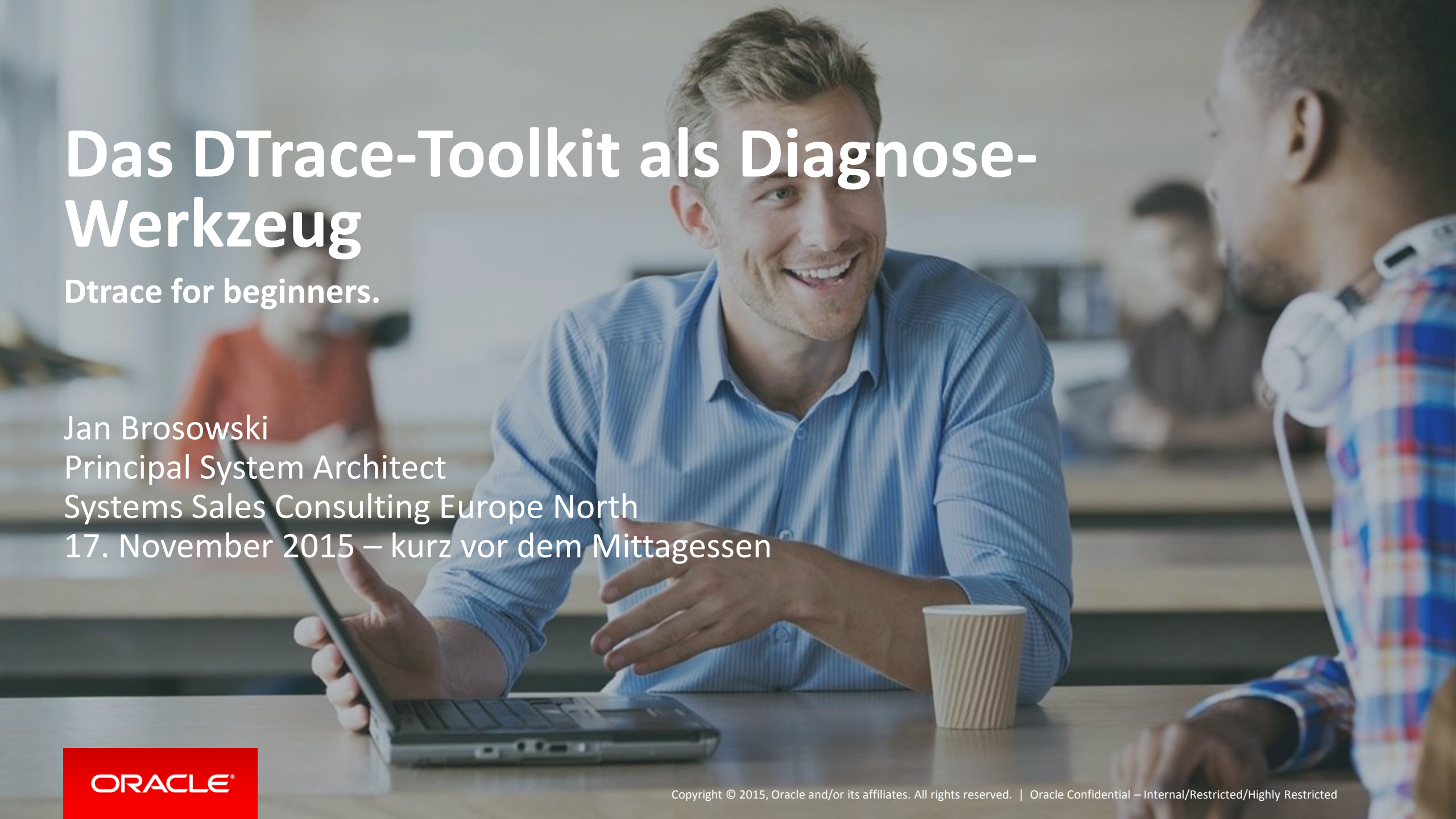

ORACLE®



Das DTrace-Toolkit als Diagnose-Werkzeug

Dtrace for beginners.

Jan Brosowski
Principal System Architect
Systems Sales Consulting Europe North
17. November 2015 – kurz vor dem Mittagessen

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

1. DTrace – was ist das eigentlich
2. Überblick DTrace-Toolkit
3. Ein Hauch Praxis

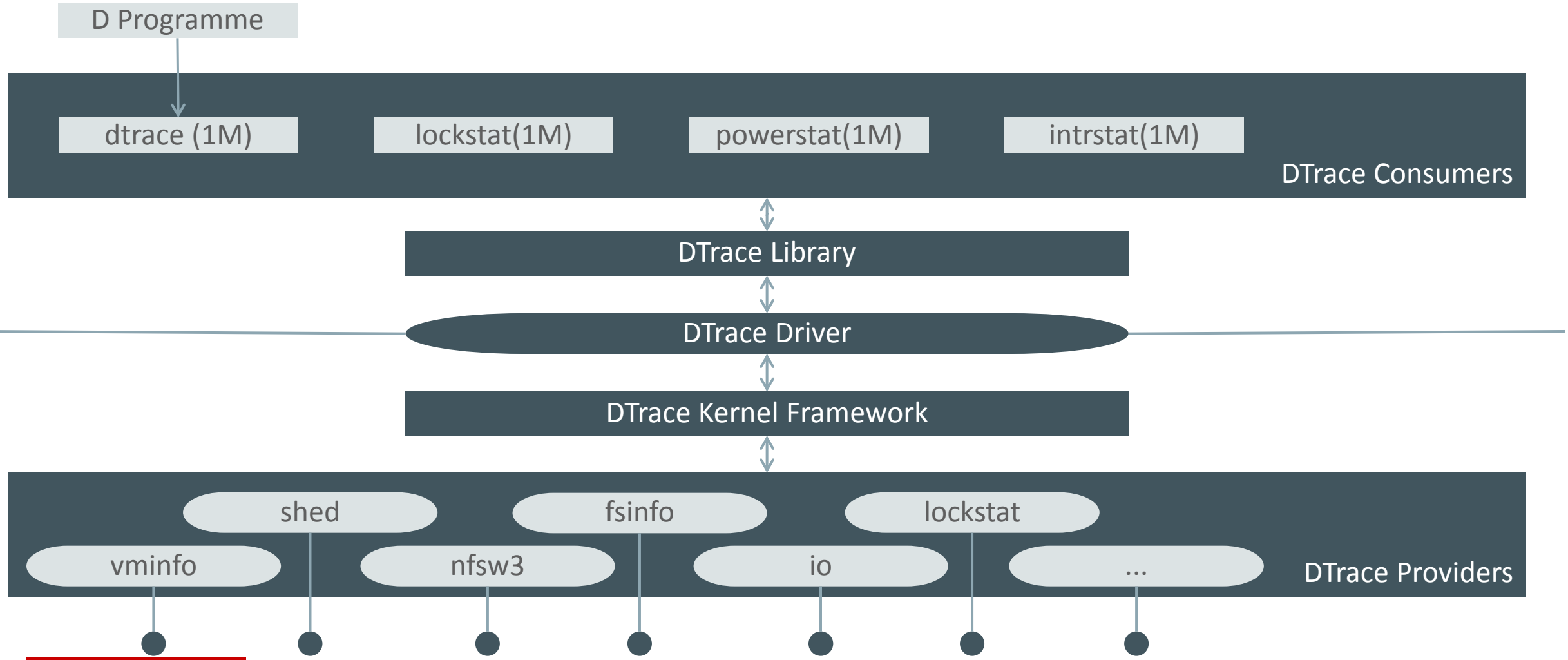
DTrace

Was ist das eigentlich?

Was ist Dtrace?

- Oracle Solaris Dtrace ist ein umfassendes Tracing-Werkzeug, mit dem systemische Probleme in Echtzeit analysiert werden können.
- DTrace kann dynamisch aktiviert werden und erlaubt das Beobachten/Vermessen von laufenden Produktionssystemen. Der Einfluss der Messung ist dabei vernachlässigbar gering.
- DTrace erlaubt es somit, die Funktionsabfolgen eines Systems zu verstehen, Probleme in den verschiedenen Layern der Software zu erkennen und zu analysieren
- Dtrace erlaubt es, Daten sowohl global als auch granular zu erfassen und zu analysieren.

Was ist DTrace?



DTrace in Zahlen

80k+ mehr als **80.000 Messpunkte**

500+ **D Scripting Language** mit mehr als 500 Seiten Dokumentation

10+ verfügbar in **Oracle Solaris 10, Oracle Solaris 11, Oracle Linux und weiteren Betriebssystemen**

2.5k **Umfangreichstes Cheat Sheet** aller Oracle-Systems-Produkte

75%

der befragten DTrace-Gelegenheitsanwender waren von der Mächtigkeit des Werkzeuges überfordert.

Überblick über das DTrace-Toolkit

Was ist das eigentlich

Die Ideen hinter dem DTrace-Toolkit

Performance-Analyse

- Vorgefertigte D-Skripte zur Performance-Analyse
- D-Skripte für gängige Programmiersprachen
- Nutzung der Integration in Applikationen (bspw. Apache Webserver)

Lern-Unterstützung

- Umfangreiche Dokumentation, um D-Skripte nachvollziehen zu können
- Teilweise Empfehungen zur Erweiterung bereits eingefügt
- Austausch von Skripten unter Nutzern (Community-Gedanke)

Das Problem des DTrace-Toolkits

I didn't know in 2005 that DTrace would appear on other operating systems, and that each kernel would change as much as it did. In particular there were numerous changes to the DTrace syscall provider, which caused the scripts to be more tied to kernel versions than expected.

Brendan Gregg

Foto von Trav Williams, Broken Banjo Photography

Riding a dead horse?

15 May, 2015

The DTraceToolkit Project Has Ended

Ten years ago on this day I created the DTraceToolkit, and it's time to call the project ended. Its scripts live on in different operating systems: OS X, FreeBSD, Oracle Solaris 11, and other Solaris derivatives; as packages for OmniOS and SmartOS; and integrated into other tools. Thanks to everyone who helped make it a success.

Today, in 2015, the 230-script DTraceToolkit is more like a large collection of ancient kernel patches that, when they do work, often do so by sheer luck. I didn't know in 2005 that DTrace would appear on other operating systems, and that each kernel would change as much as it did. In particular there were numerous changes to the DTrace syscall provider, which caused the scripts to be more tied to kernel versions than expected.

To put this all in today's terms: the DTraceToolkit became like a set of 230 *amazing* Linux 2.6.11 patches that people want working on Linux 3.2, 3.13, and 4.0, *and* FreeBSD 10.0, 11.0, *and* Oracle Solaris 11! Such a feat isn't impossible, in the strictest sense of the word, but it is impractical.

A number of the DTraceToolkit scripts live on in different OSes, and I'm glad for that to happen: they have a life of their own. But I can't recommend anyone continue the DTraceToolkit project. If you want to understand more background as to why, then my mistakes blog post should be a good start.

Dtrace-Toolkit in Solaris 11

- Offizielles Paket in Solaris 11
  - supported, gewartet, gepflegt von Oracle
  - Auswahl von angepassten Skripten aus dem OpenSource-Toolkit

- Installation:
  - bitte aus offiziellen Quellen, und an das verwendete OS angepasst.

```
pkg install dtrace-toolkit
```

  - Ältere Solaris-Versionen: MOS 1428139.1
  - Die Skripte von brendangregg.org werden nicht alle funktionieren, und sie werden nicht supportet. Guaranteed. Period.

Inhalt des DTrace-Toolkits

- Cpu/ CPU analysis
- Disk/ disk I/O analysis
- Kernel/ kernel analysis
- Locks/ lock analysis
- Mem/ memory analysis
- Net/ network analysis
- Proc/ process analysis
- System/ system analysis
- User/ user based activity analysis
- Zones/ analysis by zone

- Apps/ Application specific scripts
- Java/ tracing Java
- JavaScript/ tracing JavaScript
- Perl/ tracing Perl
- Php/ tracing Php
- Python/ tracing Python
- Ruby/ tracing Ruby
- Shell/ tracing Shell languages

Anwendung der Skripte

Systemanalyse

- Meist direkt ausführbar, parameterisierbar
- Teilweise absichtlich nur über `dtrace -s` startbar

Applikationen & Programmiersprachen

- Setzen bestimmte Versionen der Applikation, des Interpreters oder Compilers voraus (README)
- man-Page konsultieren (wann starten, Parameter...)
- Integration in Solaris Studio (und Apple Xcode / Instruments)

Ausgangssituation

Können Sie sich bestimmt gut vorstellen...

- Sie sind SAP-Basis-Admin bei einem relativ großem, mittelständischen Betrieb, verantwortlich für die Server-Hardware, Betriebssysteme, Datebanken und SAP Basis mehrere SAP-Systeme.
- Die Systeme laufen unter Solaris 10 und 11.
- Die IT ist arbeitsteilig organisiert, ihre SAP-Systeme gelten als Exoten, weil sie als einziger eine relativ große Anzahl an Komponenten unter Kontrolle haben.
- Storage beziehen sie als SAN-LUNs von einer anderen Abteilung, deren Mitarbeiter sie nicht mal in der Kantine identifizieren könnten.

Ausgangssituation

... so langsam wird es unangenehm ...

- Es ist Mittwoch morgen, 8:00 Uhr
- Die Kaffeemaschine streikt.
- Es ist kein Gebäck mehr da.
- Das Telefon klingelt.

Die Symptome werden sichbar...

- Server saturiert seine LUN mit sehr vielen Requests
- Auch im Netzwerk ist viel los.
- Server kaum ausgelastet auf CPU (vmstat und prstat), man sieht hin und wieder kurze Lastspitzen durch den SAP-Applikationsserver
- RAM belegt durch OracleDB und SAP-Applikationsserver
- SAP-System zeigt sehr geringe Anzahl an Usern
- Oracle-Datenbank zeigt kaum Last, ist aber deutlich langsamer als sonst.

Analyse

- Frage 0: Wann geht die Kaffeemaschine wieder? Wer besorgt Kekse?
- Frage 1: Welcher Prozess / Welche Prozesse greifen eigentlich ständig auf das SAN zu? Welche Dateien werden da ständig angefasst?
- Frage 2: Wieso greift ZFS Caching hier nicht?

Analyse (II)

- Frage 2 ist einfach zu beantworten gewesen:

Ein Server mit 32GB RAM, bei dem 8GB durch die SGA und 23,8GB für SAP Applikationsserver genutzt sind, hat praktisch kein RAM mehr für ZFS Caches.

Analyse (III)

- Griff in das DTRace-Toolkit
  - iotop – liefert ähnlich top eine Liste, welche Prozesse gerade wieviel IO machen
  - openstat – liefert eine Liste, welche Dateien durch welchen Prozess geöffnet werden.

- Resultat
  - Es wird immer wieder auf die gleichen, relativ großen .msi-Dateien zugegriffen.
  - Der Prozess ist ein smb-Dämon.

- Kurzes Nachdenken:
  - Per CIFS-Share wird das SAP-GUI in unterschiedlichen Versionen an etwa 1000 Client-PCs ausgeliefert. Daher ist ein smb-Dämon aktiv.
  - Das SAP-GUI wird allerdings nie von 1000 PCs simultan heruntergeladen, insbesondere nicht, weil es nicht geupdated werden muss.

- Auflösung durch Anruf beim Windows-Admin
  - Auf allen Clients ist der Virenscanner erneuert worden. Dabei wurde eine Richtlinie nicht sauber übergeben: Alle Windows-Clients prüfen die Installationsdateien des SAP-GUIs auf Viren ....

Hardware and Software
Engineered to Work Together


ORACLE®