

RESUME

Dtrace (Dynamic Tracing)

MATA KULIAH SISTEM OPERASI



DISUSUN OLEH:

NAMA: RUSMIATI

NRP: 08.04.111.00039

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO
2010**

Dtrace (Dynamic Tracing)

DTrace, atau Dynamic Tracing, adalah alat diagnostik yang kuat diperkenalkan pada Solaris 10 OS akhir tahun 2003. Sejak diluncurkannya, telah diimplementasikan dalam sistem operasi lain, yang paling penting yang FreeBSD dan Mac OS X. Tutorial ini menggunakan DTrace menganalisis beberapa aplikasi.

Dengan Dtrace programmer dan systems administrator dapat melakukan `'trace'` atau menelusuri kernel dan aplikasi di sistem yang sedang jalan (produktif) dalam upaya menemukan kekeliruan.

DTrace memiliki kemampuan yang memadai dalam untuk tracking daripada solusi yang ada sebelumnya. Setelah Sun Microsystems membebaskan kode sumbernya awal 2005, alat ini kemudian menjadi platform software yang kemudian diadopsi sistem operasi lain termasuk OpenSolaris, Mac OSX dan FreeBSD.

DTrace adalah Infrastruktur atau alat yang dilengkapi dengan AD Trace language dan interpreter (penerjemah). DTrace menjadi infrastruktur yang tangguh dan membolehkan developers untuk diagnosa system and tingkah laku aplikasi.

Ada beberapa istilah yang berkaitan dengan DTrace yakni Probe, Agregasi, Predikat dan Features.

Berikut beberapa penjelasan singkat dari beberapa istilah tersebut.

Probe mengacu dari instrumentation dengan dengan namespace schema – Provider: Module: Function : Name.

- Provider – merepresentasikan sebuah metodologi untuk melakukan instrumentasi terhadap system
- Module – Nama dari DTrace kernel module atau user library.
- Function – Nama dari program fungsi
- Name – Nama dari probe contoh: “fvt:zfs:arc_read:entry”

Aggregasi.

Sintaks Aggregasi Dtrace adalah `@name[keys] = aggfunc (args)`. Sintaks ini dapat didefinisikan sebagai berikut :

- name – Name variabel agregasi yang didahului oleh karakter `@`.
- keys – Kumpulan ekspresi D yang dipisahkan oleh simbol koma
- aggfunc – Salah satu fungsi-fungsi agregasi DTrace
- args – Kumpulan argumen yang sesuai dengan fungsi agregasi dipisahkan oleh simbol koma

Predikat.

Program D terdiri dari sekumpulan klausa-klausa probe. Klausa probe memiliki bentuk umum sebagai berikut:

```
probe descriptions
```

```
/ predicate /
```

```
{
```

```
action statements
```

```
}
```

Features

- Dynamic instrumentation Zero probe effect ketika disable
- Instrumentationnya disatukan atau sekaligus Instrument kernel dan applications yang dapat menelusuri flow-control melintas tanpa batas.
- Kernel instrumentation. Instrumen terdapat di-kernel subsystems seperti virtual memory, synchronization, scheduler
- Data Integrity dalam arti setiap error system atau program selalu dilaporkan

Sebagai fitur yang paling digemborkan dalam Solaris versi 10, DTrace dapat digunakan sebagai:

- **Debugging** Untuk mengetahui rutin-rutin apa yang sedang dipanggil, argumen dan nilai balik dari setiap fungsi-fungsinya
- **Analisa Kinerja** Dapat diamati dimana aplikasi banyak menggunakan waktunya dan berapa lama. Digunakan untuk menggali lebih dalam dan menemukan penyebab utama masalah kinerja
- **Observabilitas** DTrace membantu pemahaman terhadap sistem. Dapat diketahui siapa yang memanggil sebuah fungsi dan fungsi-fungsi lain apa yang dipanggil oleh satu fungsi tertentu yang sedang dipanggil
- **Jangkauan Kode (Code Coverage)** Sebagai piranti pengetes, DTrace digunakan untuk mengetahui apakah sebuah fungsi dipanggil dan dimana

Dengan Tools yang ada DTrace dapat membolehkan user untuk :

1. Enable probes dengan cara : `#dtrace -n BEGIN`
2. Run D-language scripts dengan cara : `# dtrace -s scriptQu.d`
3. Mengamati properties dari DTrace framework di system yang bersangkutan dengan cara : `#dtrace -l`

DTrace dapat digunakan pada saat kita ingin melakukan monitoring system, Misalnya ketika hendak menyelidiki kernel subsystem. Di sini kita dapat memantau

file systems, networks, zones, dan sebagainya. Selain itu dapat Mengumpulkan information dari seluruh system melalui DExplorer tool serta melakukan

Tuning system dan aplikasi

DTrace juga dapat digunakan untuk Web-development. Baik dengan Server-side (Ruby, Java, PHP, Groovy, Perl, Python) maupun dengan Client-side (Java-script); Tools (Apache, Glassfish, MySQL, PostgreSQL). Di samping itu, untuk kebutuhan software

development, DTrace dapat digunakan untuk Debugging, Profiling, Memory leaks, Lost CPU cycles, Function calls, dan sebagainya.

script DTrace sederhana akan terlihat seperti ini:

```
syscall:::entry // probe description syscall:::
entri // probe deskripsi
/pid == $1/ // predicate / Pid == $ 1 / / /
predikat
{ (
    @[probecount] = count(); // action @ [Probecount] = count
); // aksi
} )
```

Contoh-contoh di `/usr/demo/dtrace`

In the Solaris OS, the examples from the DTrace user guide are found in `/usr/demo/dtrace`. Di OS Solaris, contoh-contoh dari buku petunjuk DTrace ditemukan di `/usr/demo/dtrace`.

```
$ cd /usr/demo/dtrace $ Cd / usr / demo / dtrace
$ ls | head $ Ls | kepalaku
applicat.d applicat.d
badopen.d badopen.d
begin.d begin.d
callout.d callout.d
clause.d clause.d
clear.d clear.d
countdown.d countdown.d
... ..
```

Each script contains a reference to the chapter from whence it came. Setiap skrip berisi referensi ke bab dari mana ia berasal.

For example, from `applicat.d`: Misalnya, dari `applicat.d`:

```
... ..
* The full text of the this chapter may be found here: * Teks lengkap
dari bab ini dapat ditemukan di sini:
* http://wikis.sun.com/display/DTrace/io+Provider *
Http://wikis.sun.com/display/DTrace/io+Provider
... ..
```

Or you can use a browser to examine `index.html` : Atau Anda bisa menggunakan browser untuk memeriksa `index.html` :

Chapter	Script Bab Script
Introduction	<code>countdown.d</code> Pendahuluan <code>countdown.d</code> <code>counter.d</code> <code>counter.d</code> <code>hello.d</code> <code>hello.d</code>
Variables	<code>clause.d</code> Variabel <code>clause.d</code> <code>rtime.d</code> <code>rtime.d</code>
Structs and Unions	<code>kstat.d</code> Struct dan Serikat <code>kstat.d</code> <code>ksyms.d</code> <code>ksyms.d</code> <code>rwinfod.d</code> <code>rwinfod.d</code>
Aggregations	<code>clear.d</code> Agregasi <code>clear.d</code> <code>dateprof.d</code> <code>dateprof.d</code>

- [DTrace Dokumentasi](#)
- [DTrace Toolkit DTrace Toolkit](#)
- [Open Solaris Open Solaris](#)

Contoh 1: Manajemen Memori

Contoh 2: Variabel Kondisi

Contoh 3: memset

Contoh lain:

- `dtrace -s namafile.d 1234` (Dimana `namafile.d` adalah file yang berisi program D)
- `dtrace -s trace_read.d` (menulis program kita dalam file `trace_read.d`)
- `ppriv -s A+dtrace_kernel,dtrace_proc,dtrace_user [`