

kernel and device drivers.

DTrace uses a scripting language very similar to C and awk, called *D*, which provides numerous possibilities for filtering and summarizing data in a safe way in the kernel before handing it off for more processing in user space. This minimizes the overhead associated with collecting and presenting data and allows using DTrace safely and with minimal performance impact on production systems.

The DTrace Toolkit provides a large collection of ready-made scripts that cover almost all aspects of performance and runtime system analysis.

More information on DTrace can be found at <http://dtrace.org/>.

Distributions based on illumos

OpenIndiana



OpenIndiana is the successor to Sun's OpenSolaris distribution and can be used on both desktop and server systems. It preserves the idea of having a general purpose based operating system. OpenIndiana is used by many illumos developers as primary development and test platform.

<https://www.openindiana.org>

SmartOS

SmartOS is developed by Edgecast Inc. specifically for cloud computing and virtualization purposes. The base system is optimized for this task

and uses no packaging or installation. Additional software for SmartOS zones is provided through the NetBSD pkgsrc, which is also worked on by Edgecast Inc. developers. SmartOS is part of the TritonDataCenter product produced and supported by Edgecast Inc.

<https://smartos.org>



illumos is an advanced open source operating system core derived from OpenSolaris. The project was founded in 2010 to continue the development of the OpenSolaris code independently of Oracle.

illumos is supported by a few companies who use the illumos code as a platform for their own products. The illumos project functions as a hub for collaboration on development of the core operating system used by the various different illumos distributions.

Some illumos distributions are developed by companies to match their requirements, and they may focus on storage, virtualization, or cloud computing, while others are developed by hobbyists and target a more general server/desktop use case. They all share the common code consisting of the kernel, libraries, daemons, and utilities.

What's in it for me?

Containers - illumos Zones

Yes, we do have containers - one of the earliest implementations. A **zone** in illumos provides a separated userland environment, that feels almost like a virtual system, but without the overhead of a hypervisor. Zones support not only encapsulation of

processes, file systems, users and network connectivity, but also resource management. With the built-in resource management, you can hard limit the zone to an amount of CPU resources, or pin it to specific CPUs, or even set up a Fair-Share-Scheduler (*FSS*) which allows running containers smoothly side-by-side by ensuring a minimum of needed resources, but also using spare resources in case your container needs more for a time.

Virtual Machines

Of course, illumos also supports virtualization. On the amd64 platform, we support the *hyve* Hypervisor ported from FreeBSD, which provides modern virtualization features like PCI-passthrough. The major illumos distributions will run virtual machines in zones, so you only need one interface to manage both virtualization technologies.

ZFS - the Zettabyte Filesystem

illumos is the source of ZFS as a direct descendant of OpenSolaris. The illumos project maintains a stable ZFS version, independent of the OpenZFS development. While this means that illumos might get new developments later, it provides more stability for the user. ZFS manages storage devices such as SSDs in pools and provides two simple commands **zfs(8)** and **zpool(8)** to manage the whole storage stack from RAID to filesystem attributes. ZFS does full checksumming on all data and metadata and works transaction based, so your filesystem is always in a stable state.

Crossbow - Network Virtualization

The network virtualization technology *Crossbow* allows creation of virtual networks, switches, and network interfaces on a single system. Separate instances of the network stack allow bandwidth control and isolation of protocols, services, and virtual machines. Whether you need virtual network interface cards (*VNICs*) on a physical one, virtual switches (*Etherstubs*) to host *VNICs*, you want to setup VLANs or even VXLANs - everything can be done with a single and simple command, **dladm(8)**. This makes it possible to consolidate complex network topologies including servers, routers, and firewalls on a single system.

COMSTAR - Storage Virtualization

illumos features the *COMSTAR* storage virtualization which allows you to use an illumos system as an *iSCSI* or *Fibre-Channel* storage system - parallel to the network filesystem (*NFS*) or *CIFS*. In combination with *ZFS* you can build storage server solutions that exactly fit your needs.

Service Management Facility

The *Service Management Facility* (SMF) manages the startup of various system services. Of course you can also use it to start your applications. SMF also monitors your service and will try to restart it in case of an error and collects the output of the application in a separate log file. SMF maintains dependencies between services and can start independent services in parallel.

Fault Management Architecture

The *Fault Management Architecture* (FMA) collects, analyzes, and reports errors from different hardware components such as CPUs, memory, and disks as well as from OS subsystems such as ZFS and SMF. Analysing soft errors like correctable ECC errors in main memory or unexpectedly long access times on disks allows prediction of failures and timely deactivation of devices before catastrophic failure occurs. Error reports are created in a consistent format and allow the system administrator to react quickly to minimize or even prevent system downtime.

Role-Based Access Control

Role-Based Access Control (RBAC) is a security feature allowing fine-grained privileges and privileged roles to be assigned to users and processes. Roles are a special kind of account which cannot be logged into. Instead they can temporarily be assumed by normal users that have that role assigned to their account. The individual privileges may allow a user or role to perform a certain task, like rebooting the system, or managing a certain SMF service, without the requirement to allow fully privileged access to certain commands.

DTrace - Dynamic Tracing

DTrace is an instrumentation tool mostly used for performance analysis and troubleshooting. To achieve this DTrace dynamically patches live running instructions with instrumentation code. DTrace can instrument the whole software stack on a system, programs in user space as well as the