Fishworks

Brendan Gregg
Cindi McGuire

Sun Microsystems
Fishworks is the name of an engineering team at Sun Microsystems

- FISH: “Fully Integrated Software and Hardware” - a suitable acronym to describe our strategy
- Our goal – to provide a unified management framework for appliances built on Solaris
Fishworks Overview

- Fully Integrated Software and Hardware
- Unified User Interface
- Turning Solaris into an appliance
- Example: NAS appliance
What Does it Take to Build an Appliance?

- Solid OS foundation
  - Key Solaris 10 building blocks:
    - SMF (Service Management Facility)
    - FMA (Fault Management Architecture)
    - DTrace (Dynamic Tracing)
    - Networking
    - Security
- Common user interface
- Integrated higher-level management and configuration tasks with OS
Unified User Interface

- One User Interface to rule them all
  - BUI: Browser User Interface
  - CLI: Command Line Interface
- This is possible in the confines of an appliance
  - A special-purpose server confined to a limited set of configuration and management tasks
BUI: Browser User Interface

- Consistent look and feel
- As fast as possible
- Usability – no special OS knowledge required
- Value add – a real BUI (not a CLI wrapper)
  - Pie charts, traffic lights, plots, dialogs, navigation, ...
- Status updated live – no need to refresh
- Not a(nother) skin – speaks to akd, which speaks to OS
- Communication secured over HTTPS
- Extensive test framework
  - Required writing a JavaScript CLI
BUI Examples

Masthead:

Lists:

<table>
<thead>
<tr>
<th>NAME</th>
<th>SIZE</th>
<th>MOUNTPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin / accounts</td>
<td>18K</td>
<td>/export/accounts</td>
</tr>
<tr>
<td>admin / exports</td>
<td>18K</td>
<td>/export/exports</td>
</tr>
<tr>
<td>admin / primary</td>
<td>18K</td>
<td>/export/primary</td>
</tr>
<tr>
<td>admin / traffic</td>
<td>18K</td>
<td>/export/traffic</td>
</tr>
<tr>
<td>admin / workflow</td>
<td>18K</td>
<td>/export/workflow</td>
</tr>
</tbody>
</table>
Dashboard:

**Usage**
- **Storage**
  - 5.62G Used
  - 251G Avail
  - Compression 1x
- **Memory**
  - 941M Cache
  - 12.2G Unused
  - 1G8M Mgmt
  - 598M Other
  - 2.11G Kernel

**Services**
- NFS
- iSCSI
- CIFS
- FTP
- HTTP
- NDMP
- Antivirus
- NIS
- LDAP
- AD
- ID Map
- DNS
- IPMP
- NTP
- Routing
- Phone
- SNMP
- Tags
- Sys ID
- SSH

**Hardware**
- CPU
- Memory
- Disks
- Cards
- Fans
- PSU

**Recent Alerts**
- 2008-10-8 12:05:56: Finished replicating project 'test_project2' from appliance 'loader.sf.fishpong.com'.
- 2008-10-8 12:05:52: Began replicating project 'test_project2' from appliance 'loader.sf.fishpong.com'.
CLI: Command Line Interface

- Mirror BUI functionality as much as possible
- Standard framework – a tree of contexts
- Usability
  - Help for every context
  - Tab-completion ++
- Rich scripting environment
  - Stripped-down JavaScript
  - SSH keys can be added for automated scripts from a different host
CLI Example

vimba:> `tree`

```
---+ configuration
   +--- net
       +--- datalinks
       +--- devices
               +--- interfaces
               +--- services
```

...`

vimba:> `configuration net interfaces select e1000gtab`
e1000g0  e1000g1
vimba:> `configuration net interfaces select e1000g1`
vimba:configuration net interfaces e1000g1> `set v4dhcp=tab`
false  true
CLI Scripting Example

% ssh root@vimba << EOF
configuration net interfaces
select e1000g1
show
EOF

Properties:

<state> = up
class = ip
label = Untitled Interface
admin = true
links = nge0
dhcp_clientid =
dhcp_hostname =
dhcp_primary = false
  v4addrs = 192.168.2.124/22
  v4dhcp = true
  v6addrs =
  v6dhcp = false
Solaris Server Configuration

For example...

- **NFS**
  
  `/etc/default/nfs`
  
  `/var/svc/log/network-nfs-server:default.log`

- **DNS**
  
  `/etc/resolv.conf, /etc/nsswitch.conf`
  
  `/var/svc/log/network-dns-client:default.log`

- **Networking**
  
  `ifconfig, dladm, netstat, route, routeadm`
  
  `/etc/inet/hosts, /etc/inet/ipnodes, /etc/hostname.*`
  
  `/var/adm/messages, /var/svc/log/*`

- **Consider NIS, LDAP, FTP, Apache, iSCSI, etc...**
vimba:> configuration services nfs
vimba:configuration services nfs> show
Properties:

<status> = online
version_min = 3
version_max = 4
nfsd_servers = 500
grace_period = 90
  mapid_dns = true
  mapid_domain = domain

vimba:configuration services nfs> set grace_period=30
  grace_period = 30 (uncommitted)
vimba:configuration services nfs> commit
vimba:configuration services nfs> get grace_period
  grace_period = 30
Solaris Server Status

For example...

- **Hardware**
  
  `fmadm faulty`

- **Services**
  
  `svcs` (if the service is in SMF, otherwise application specific commands and log files must be used to determine service status)

- **Consider older Solaris (and other OSes):**
  
  `ps -ef, iostat -En, netstat -i
  /var/adm/messages, /var/log/*`
Fishworks Server Status

![Fishworks Server Status Image]

- Sun Microsystems
- 0810QAS004
- Disks, Slots, CPU, Memory, Fans, PSUs, SP
- HDD 0, HDD 1, HDD 2, HDD 3, HDD 4, HDD 5, HDD 6, HDD 7, HDD 8, HDD 9, HDD 10, HDD 11, HDD 12, HDD 13, HDD 14, HDD 15
- Seagate, ST914602SSUN 146G
- 137G, System, Data, Spare

Learn it. Live it. Make IT real. CEC 2008
Fishworks Server Status

```
tarpon:> maintenance hardware show
Chassis:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>MANUFACTURER</th>
<th>MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>chassis-000</td>
<td>0839QCJ01A</td>
<td>ok</td>
<td>Sun Microsystems, Inc...</td>
</tr>
<tr>
<td>cpu-000</td>
<td>CPU 0</td>
<td>ok</td>
<td>AMD Quad-Core AMD Op</td>
</tr>
<tr>
<td>cpu-001</td>
<td>CPU 1</td>
<td>ok</td>
<td>AMD Quad-Core AMD Op</td>
</tr>
<tr>
<td>cpu-002</td>
<td>CPU 2</td>
<td>ok</td>
<td>AMD Quad-Core AMD Op</td>
</tr>
<tr>
<td>cpu-003</td>
<td>CPU 3</td>
<td>ok</td>
<td>AMD Quad-Core AMD Op</td>
</tr>
<tr>
<td>disk-000</td>
<td>HDD 0</td>
<td>ok</td>
<td>STEC MACH8 IOPS</td>
</tr>
<tr>
<td>disk-001</td>
<td>HDD 1</td>
<td>ok</td>
<td>STEC MACH8 IOPS</td>
</tr>
<tr>
<td>disk-002</td>
<td>HDD 2</td>
<td>absent - -</td>
<td></td>
</tr>
<tr>
<td>disk-003</td>
<td>HDD 3</td>
<td>absent - -</td>
<td></td>
</tr>
<tr>
<td>disk-004</td>
<td>HDD 4</td>
<td>absent - -</td>
<td></td>
</tr>
<tr>
<td>disk-005</td>
<td>HDD 5</td>
<td>absent - -</td>
<td></td>
</tr>
<tr>
<td>disk-006</td>
<td>HDD 6</td>
<td>ok</td>
<td>HITACHI HTE5450SASUN500G</td>
</tr>
<tr>
<td>disk-007</td>
<td>HDD 7</td>
<td>ok</td>
<td>HITACHI HTE5450SASUN500G</td>
</tr>
<tr>
<td>fan-000</td>
<td>FT 0</td>
<td>ok</td>
<td>unknown ASY,FAN,BOARD,H2</td>
</tr>
</tbody>
</table>
...```
Fishworks Server Status
vimba:> configuration services show
Services:

   ad => disabled
cifs => disabled
dns => online
ftp => disabled
http => disabled
identity => online
idmap => online
ipmp => online
iscsi => online
ldap => disabled
ndmp => online
nfs => online
nis => disabled
ntp => disabled

&
Solaris Server Performance Observability

For example...

- **CPU**
  - `vmstat`, `mpstat`, `prstat`, `dtrace`
- **Memory**
  - `vmstat`, `prstat`
- **Disk I/O**
  - `iostat`, `dtrace`
- **Network I/O**
  - `netstat`, `dladm`, `nicstat`, `nx.se`, `dtrace`
- **NFS**
  - `nfsstat`, `dtrace`
Fishworks Server Performance Observability
Fishworks Server Performance Observability

Ok, that's a bit hard to do in the CLI. This is one of the few differences between BUI and CLI functionality.

But while the graphs aren't available, the data is:

```
vimba:> status activity show
Activity:
   CPU          10 %util              Sunny
   Disk         2 ops/sec             Sunny
   iSCSI        0 ops/sec             Sunny
   NDMP         0 bytes/sec           Sunny
   NFSv3        0 ops/sec             Sunny
   NFSv4        0 ops/sec             Sunny
   Network      3K bytes/sec          Sunny
   CIFS         0 ops/sec             Sunny
```

And individual statistics (datasets) ...
Fishworks Server Performance Observability

vimba:-> analytics datasets
vimba:analytics datasets> show
Datasets:

<table>
<thead>
<tr>
<th>DATASET</th>
<th>STATE</th>
<th>INCORE</th>
<th>ONDISK</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataset-000</td>
<td>active</td>
<td>893K</td>
<td>342K</td>
<td>arc.accesses[hit/miss]</td>
</tr>
<tr>
<td>dataset-001</td>
<td>active</td>
<td>270K</td>
<td>83.1K</td>
<td>cpu.utilization</td>
</tr>
<tr>
<td>dataset-002</td>
<td>active</td>
<td>748K</td>
<td>280K</td>
<td>cpu.utilization[mode]</td>
</tr>
</tbody>
</table>

&

vimba:analytics datasets> select dataset-006 read 5

<table>
<thead>
<tr>
<th>DATE/TIME</th>
<th>%UTIL</th>
<th>%UTIL BREAKDOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006-2-15 15:56:55</td>
<td>7</td>
<td>6 kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 user</td>
</tr>
<tr>
<td>2006-2-15 15:56:56</td>
<td>7</td>
<td>6 kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 user</td>
</tr>
<tr>
<td>2006-2-15 15:56:57</td>
<td>29</td>
<td>17 user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12 kernel</td>
</tr>
</tbody>
</table>

&
Missing Piece

That looks great but how do we link our new Unified User Interfaces with the core OS services in Solaris?
Fishworks Unified Management

- Appliance Kit Daemon (akd)
  - Not a(nother) wrapper around the Solaris CLIs
  - Tightly integrated with the Solaris OS libraries to provide appliance abstractions for:
    - Storage: ZFS, NDMP
    - Protocols: iSCSI, NFS, CIFS, HTTP, FTP, WebDAV
    - Networking: ifconfig, routing, IPMP
    - Security: OpenSSL, ssh
    - RAS: fmd, libtopo, IPMI, SMBIOS, SNMP
    - Service management: SMF
    - Observation: DTrace, kstats
Fishworks Unified Management

- Additional features added to support appliance-specific tasks
- Clustering
- Software upgrade/rollback
- Integrated phone home, service tag, and audit capabilities
- Roles and authorizations
- Secure communication channel for BUI and CLI

- Customers interact with the BUI or CLI, akd interacts with Solaris
Putting it All Together

- Common BUI, CLI, and test framework to drive management software: JavaScript
- Standard protocol for communication: XML-RPC
- Common control point (akd) to OS libraries
- Enhance OS to leverage appliance hardware: clustering and ZFS L2ARC
- Hardware supported by FMA
SMF: Service Management Facility

- Service abstraction for a running application, device state or set of other services
- SMF(5) provides a common infrastructure for service:
  - Configuration
  - Fault monitoring
  - Restart
  - Observability
- All appliance applications and facilities run under the SMF
FMA: Fault Management Architecture

- Appliance software and hardware errors reported to fmd(1M)
  - CPU/Memory, PCI-Express, HBA controllers, fans, power supplies, and disks
  - Appliance kit software instrumented for FMA
- Faults and defects reported using the Sun Fault Messaging Standard with problem resolution at http://www.sun.com/msg
- Guided FRU replacement made possible by FMA topology libraries
- IPMI, SMART, and other sensor data collected and reported to fmd(1M)
- Configurable SNMP traps and alerts
DTrace

- Analytics uses DTrace (and Kstat) to visualize statistics in real-time
  - Not just bolting on a GUI, but rethinking how to visualize performance – and investigating what new features GUIs make possible
- Statistics can be archived and saved forever
  - Investigate performance issues after the event
- Analytics can answer high level questions:
  “What clients are making NFS requests?”
  “What CIFS files are being accessed?”
  “How long are disk operations taking?”
DTrace: Analytics

Demonstrating how GUIs can *add value*
A Word about the Solaris Shell

• The appliance is entirely manageable from the BUI and CLI: no Solaris shell access required. For example:

  \texttt{ifconfig} → \texttt{buri:> configuration net}

  \texttt{route} → \texttt{buri:> configuration services routing}

  ping/nslookup (builtins)

    \texttt{buri:> ping kipper}
    \texttt{buri:> nslookup 192.168.2.104}

• akd manages resources such as ZFS, use of the original zpool/zfs commands can easily create issues that are extremely difficult to troubleshoot

• The Solaris shell is available for trained Sun Service staff to use only if absolutely necessary.
Example: NAS appliance

- Features from Solaris 10:
  - Enterprise-class scalability, RAS, and performance
  - IPv4 and IPv6 networking, LACP, IPMP, VLANs, ...
  - NFSv3, v4, FTP, HTTP, WebDAV, iSCSI, and now CIFS
  - Scalability of all key subsystems to 64 cores and beyond
  - Unique innovations: ZFS, DTrace, FMA, SMF, …

- Features added/enhanced for this appliance:
  - ZFS: L2ARC, log devices, RAID-Z DP
  - Integration with Solaris CIFS and Windows Identities
  - Clustering

...
Example: NAS appliance
A tour of the interface and features
CEC 2008

Thank you

Learn it. Live it.
Make IT real.

November 9-14, 2008 Las Vegas, NV