Open HA Cluster

Hochverfügbarkeit mit minimalem Cluster

FrOSCon
23. August 2009

Thorsten Früauf
Availability Engineering
Sun Microsystems GmbH

Agenda

- Motivation
- Open HA Cluster 2009.06
- Minimale HA Konfiguration
  - COMSTAR / iSCSI / ZFS
  - Crossbow
  - Weak Membership
  - IPS
- Live Demo
- Referenzen

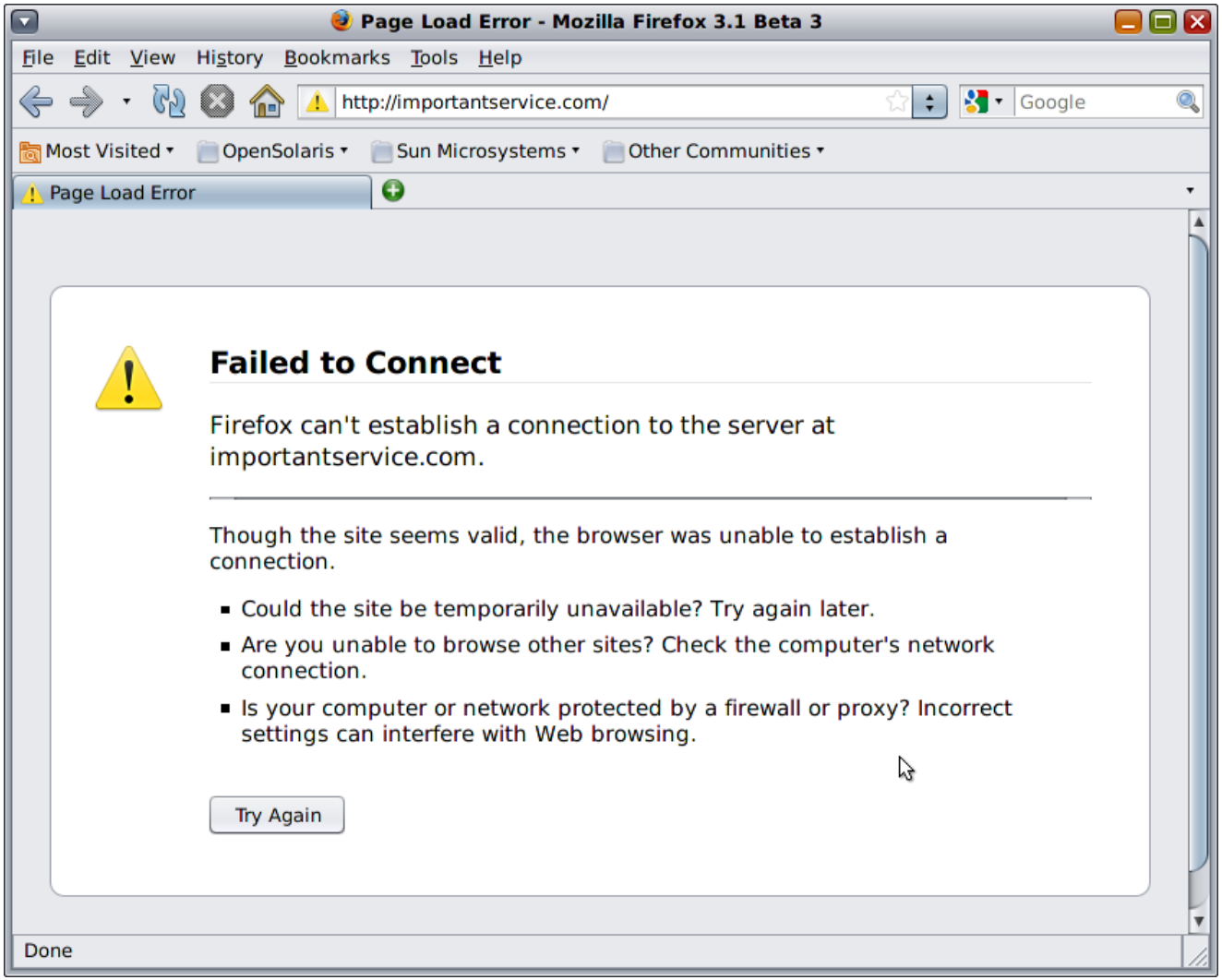
Warum Hochverfügbarkeit?

- Computersysteme bieten Dienste an:
  - Web Services, Datenbanken, Business Logic, File Systems, etc.
- Ausfallzeit ist teuer
  - Dienste sollen möglichst 100% der Zeit verfügbar sein
- Fehler sind unausweichlich:
  - Softwarefehler
  - Ausfall von Hardware-Komponenten
  - Leute und Prozesse
  - Naturkatastrophen
  - Terrorismus

Ziel von Hochverfügbarkeits-Cluster

HA Cluster automatisieren die Reaktion auf unvermeidbare Fehler und ermöglichen durch Wiederherstellung des Dienstes eine Minimierung von Ausfallzeiten und Kosten.

Was man also vermeiden möchte...



Methoden HA zu implementieren

- Redundante Hardware (Physikalische Systeme, Netzwerkkarte, Netzwerkpfade, Speichersysteme, Speicherkarte, Speicherpfade, etc.)
- Software Überwachung (die kompletten Hardwarekomponenten, Anwendungen, etc.)
- Schwenkt im Fehlerfall Dienste zur verbleibenden Hardware

Wahrnehmung von HA Clustern

- kompliziert
- schwerfällig
- schwierig zu installieren
- schwierig zu benutzen
- braucht spezielle Hardware
- teuer

Wahrnehmung ist nicht ganz unbegründet...

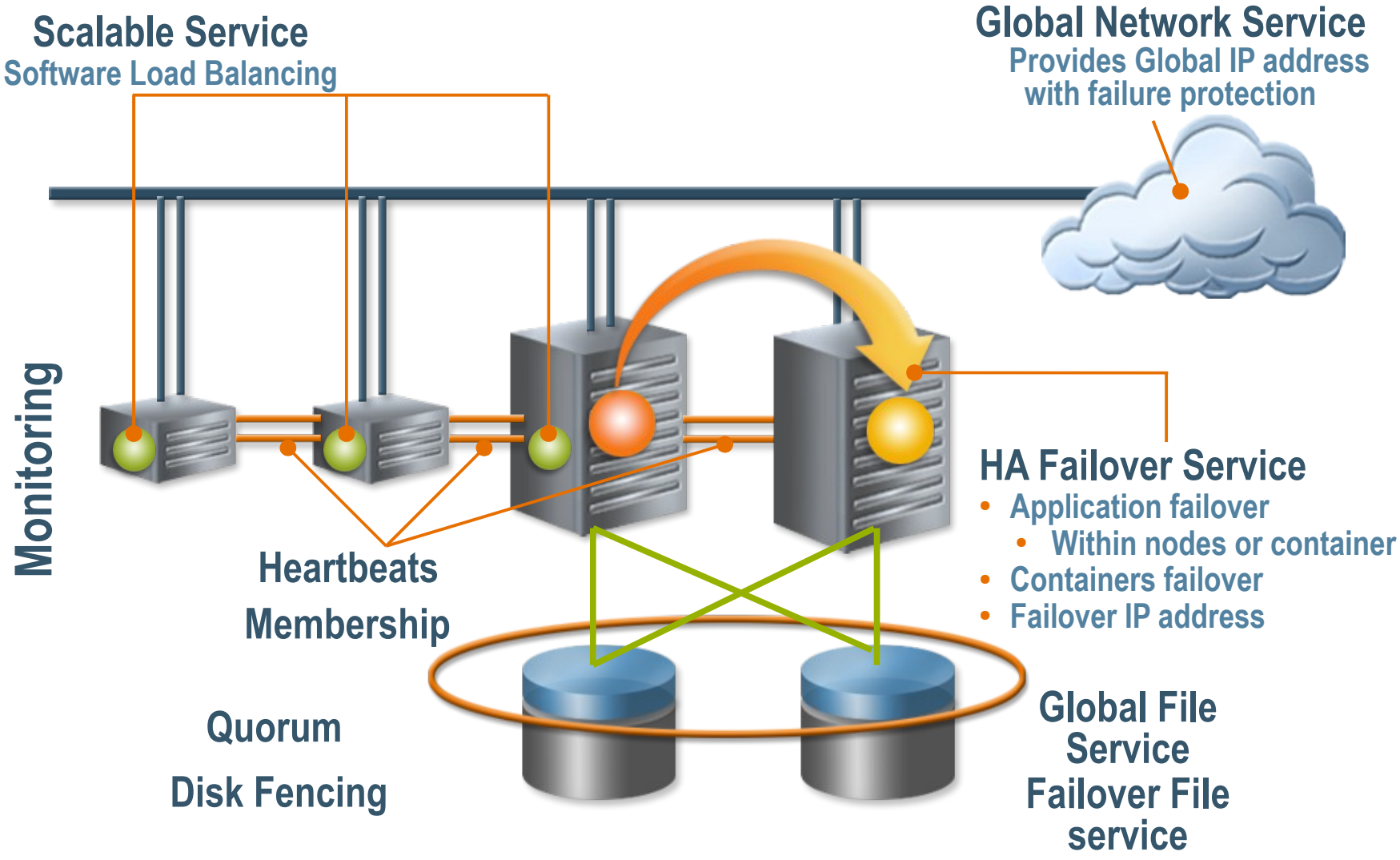
Typische HA Hardware Konfiguration

- Zwei oder mehr physikalische Systeme
- Vier oder mehr Netzwerkkarten pro System
- Dedizierter Interconnect zwischen Knoten
- Gemeinsam benutzbare Speichersysteme (DAS, SAN, NAS)
- Redundante Speicherpfade von jedem Knoten
- Quorum arbitration device
- etc.

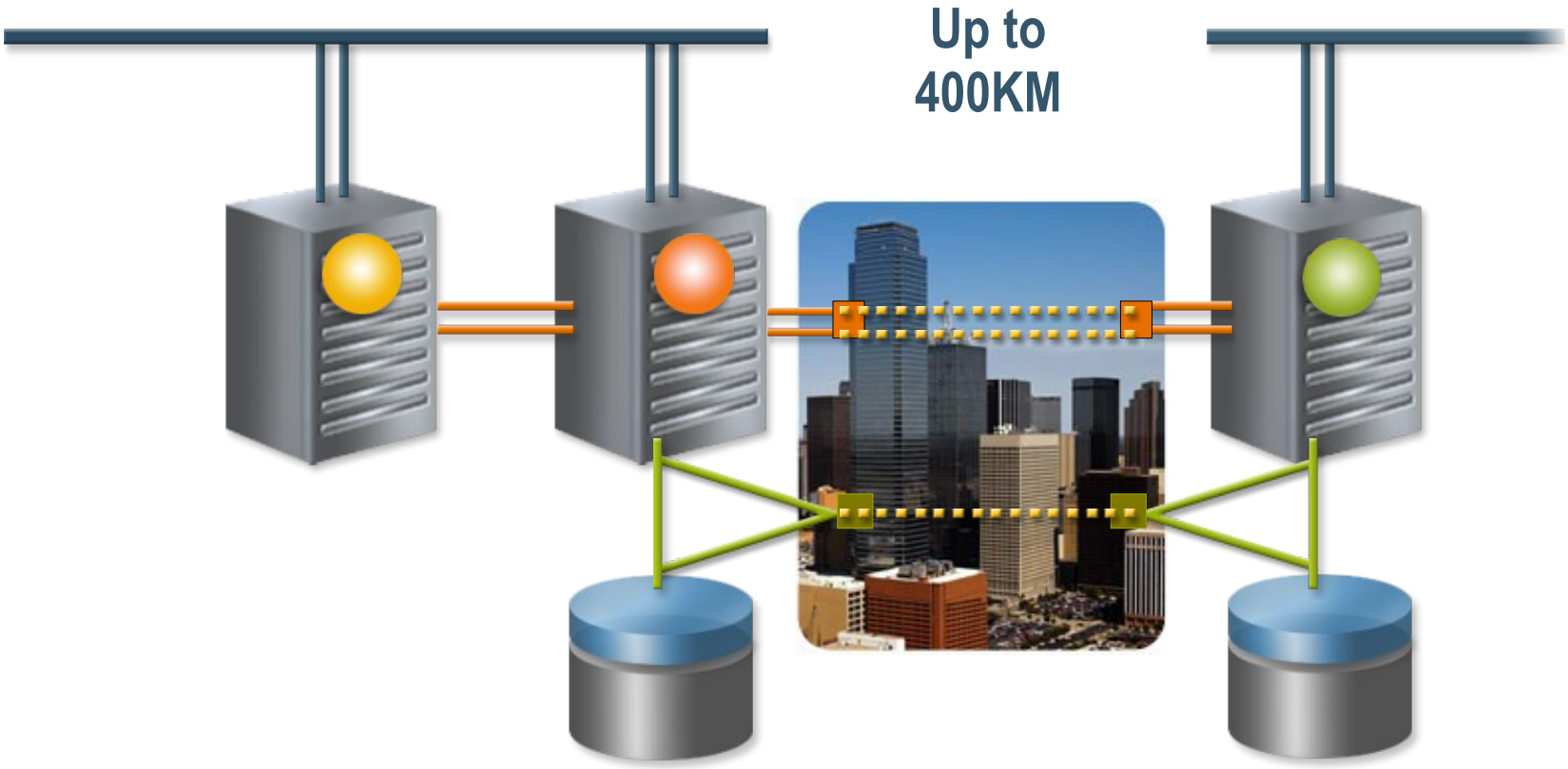
Typische HA Software Komponenten

- Heartbeats
- Membership
- Distributed configuration repository
- Service management
- Cluster-private networking layer
- Global file system
- Network load-balancing
- etc.

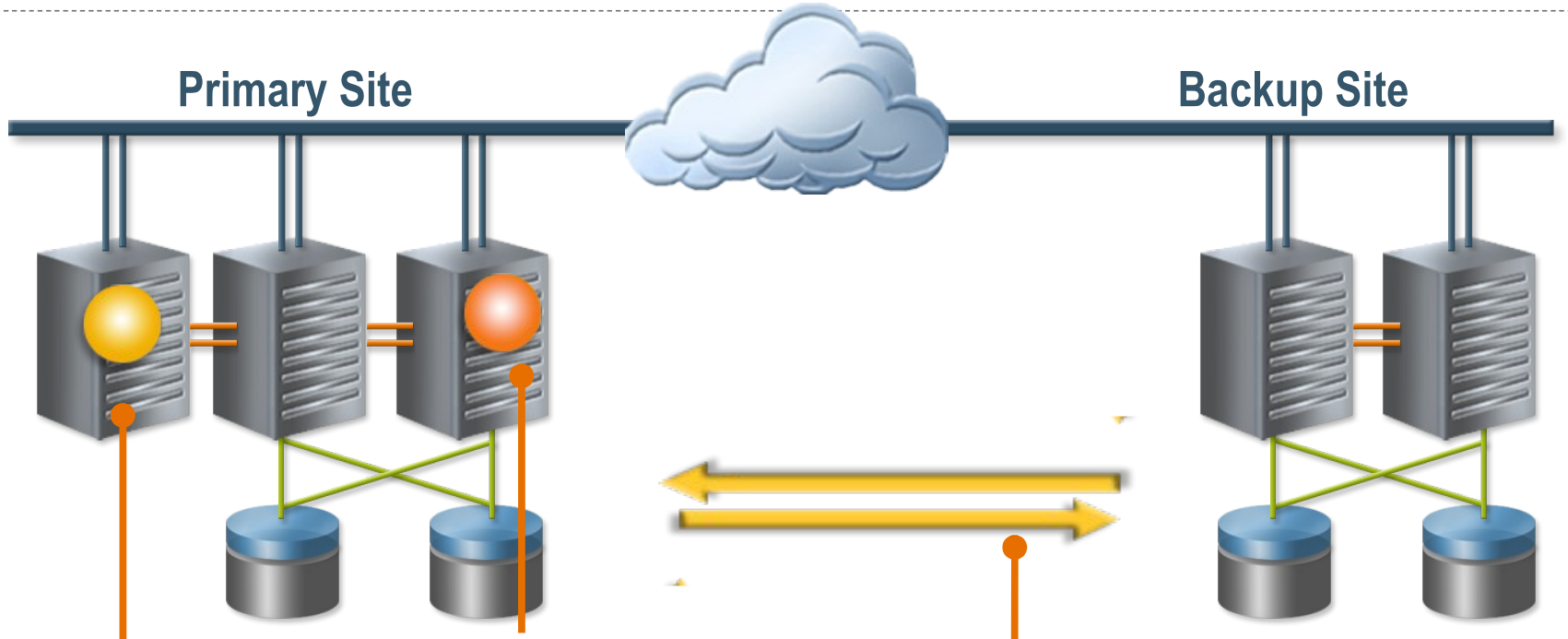
Solaris Cluster Architektur



Campus / Metro Cluster



Solaris Cluster Geographic Edition



solaris 9 & 10

opensolaris

intel

ULTRASPARC

Oracle RAC support

Replication

Sun StorEdge Availability Suite 4.0

EMC SRDF

HDS Truecopy

Neubewertung HA Cluster Komplexität

- Viele Anwendungsfälle (incl. SLA) brauchen alle Hardware und Software Komponenten traditioneller HA Cluster
- … aber eben nicht alle... Ansatz: „gut genug“ reicht auch!
- Installation und Konfiguration nur von den Komponenten, die wirklich gebraucht werden

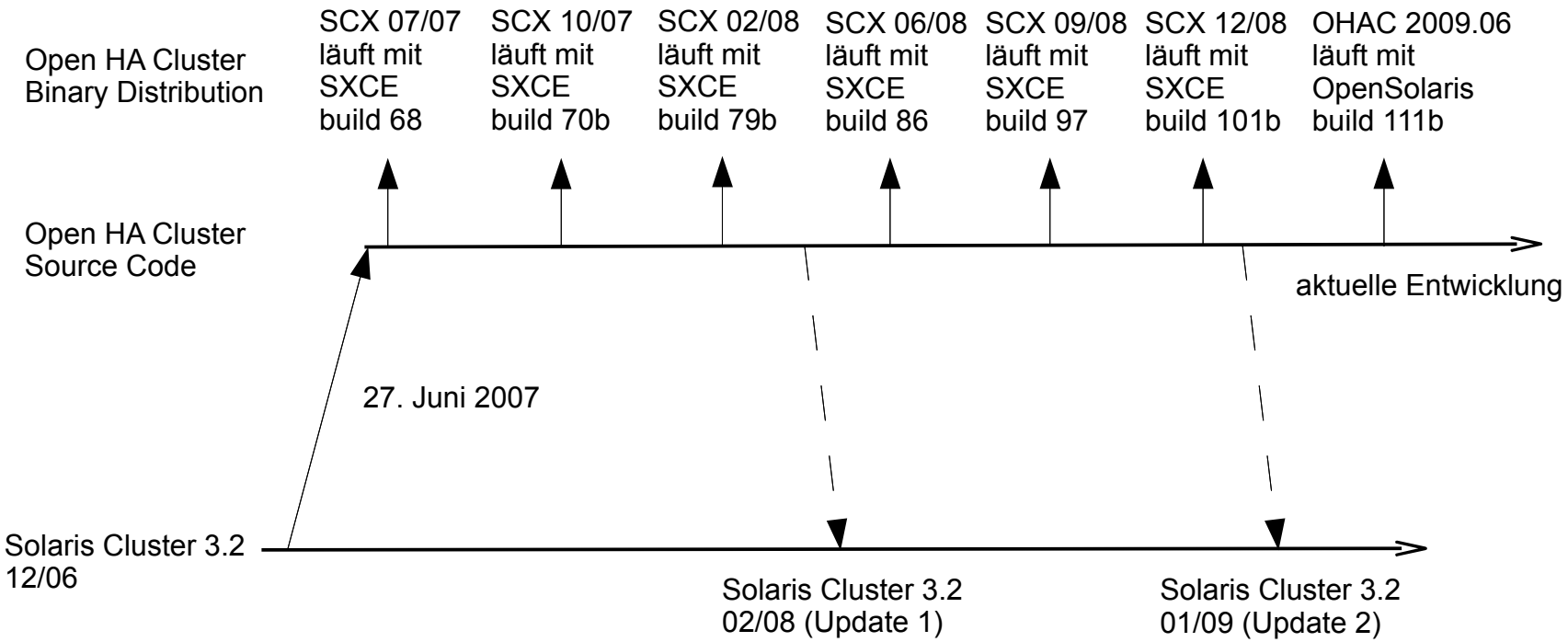
Ziel von Projekt Colorado

- Bereitstellung eines schlanken und modularen Cluster Framework, welches mit minimaler Hardware Konfiguration läuft
- Was bisher möglich war, soll möglich bleiben!

Der Weg zum Ziel

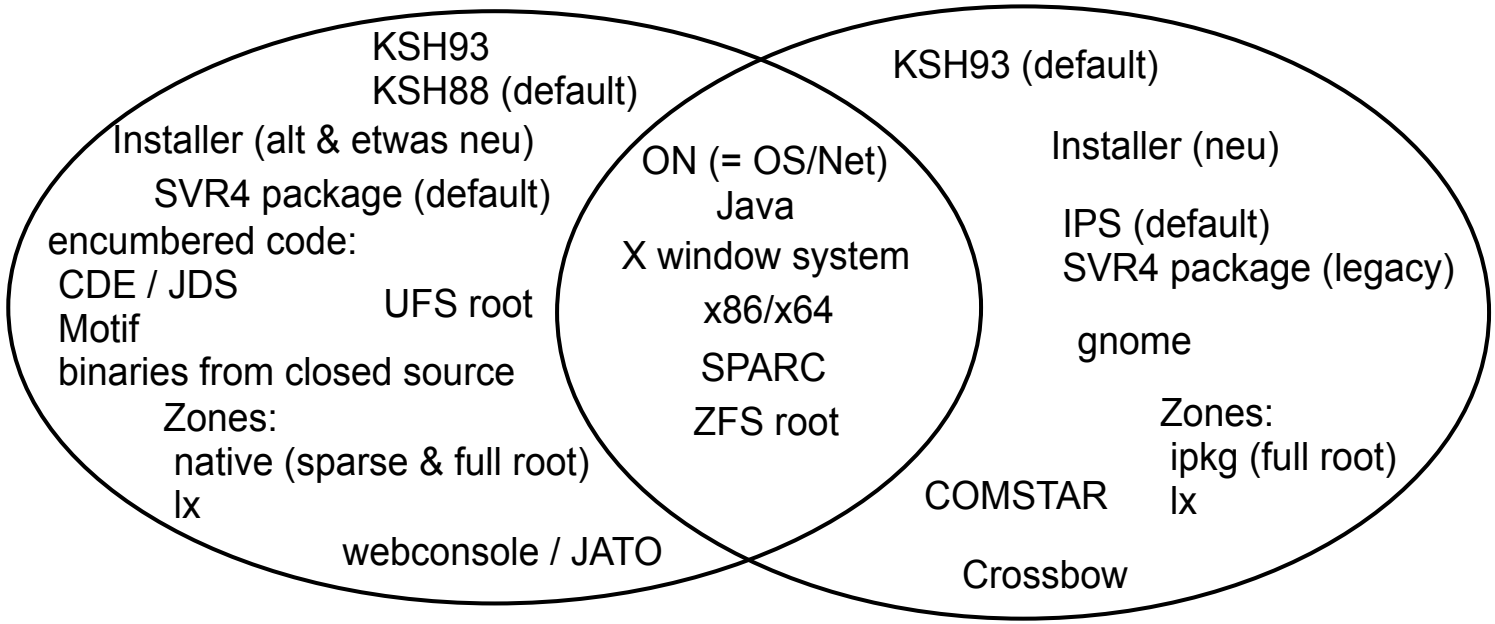
- Portierung des Open HA Cluster Quelltextes nach OpenSolaris
- Hinzufügen von Eigenschaften zur Minimierung der Hardware Anforderungen
- Nutzung des Image Packaging System (IPS) zur Implementierung von Software Modularität und Erweiterbarkeit
  - Analyse aller Paketabhängigkeiten

Entwicklungszusammenhänge



SCX = Solaris Cluster Express
OHAC = Open HA Cluster
SXCE = Solaris Express Community Edition

Solaris Express vs. OpenSolaris



Solaris Express (Nevada)

Binärdistribution aus
usr/src und usr/closed
nicht frei verteilbar

OpenSolaris 200X.Y

Binärdistribution auf LiveCD
frei verteilbare Packete
(pkg.opensolaris.org Repo)

nicht frei verteilbare Packete
(pkg.sun.com Repo)

Open HA Cluster 2009.06 (Colorado-I)

- Läuft mit OpenSolaris 2009.06 (SPARC & x86/x64)
- Viele Eigenschaften von Solaris Cluster 3.2 sind verfügbar
- Freie Nutzung (ohne Support)
  - Support subscriptions verfügbar
- Installation vom IPS package repository
  https://pkg.sun.com/opensolaris/ha-cluster
- Source ist offen und frei verfügbar unter
  http://www.opensolaris.org/os/community/ha-clusters/

Open HA Cluster 2009.06 Agenten

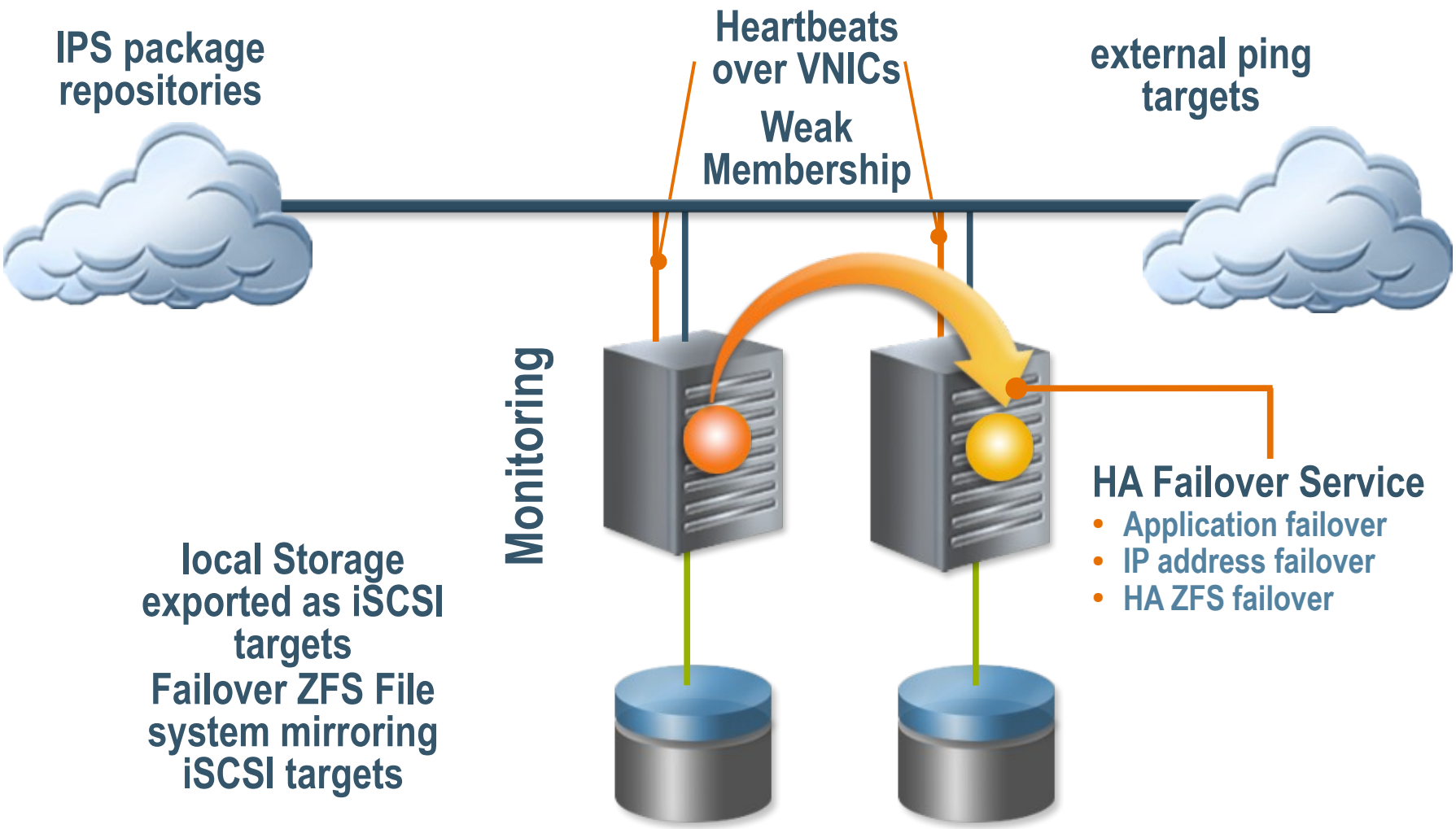
- Apache Webserver
- Apache Tomcat
- MySQL
- GlassFish
- NFS
- DHCP
- DNS
- Kerberos
- Samba
- HA Containers (ipkg Zones)
- Generic Data Service (GDS)

Hardware Minimierung

- Nutzung von lokalen Festplatten als “Poor man's shared storage” mittels COMSTAR iSCSI und ZFS
- Nutzung von Crossbow VNICs um den privaten Cluster Interconnect über das externe Netzwerk zu legen
- “Weak membership” (preview-only feature)

Zusammengefasst: zwei beliebige Knoten im gleichen IP Subnetz können einen funktionsfähigen Cluster bilden.

Minimale HA Konfiguration



Technologien zur Minimierung

- Weak Membership
- Software Quorum
- Quorum Server
- Optional Fencing
- HA ZFS
- COMSTAR / iSCSI
- IPsec
- Crossbow
- IPS
- VirtualBox (für Entwicklung)

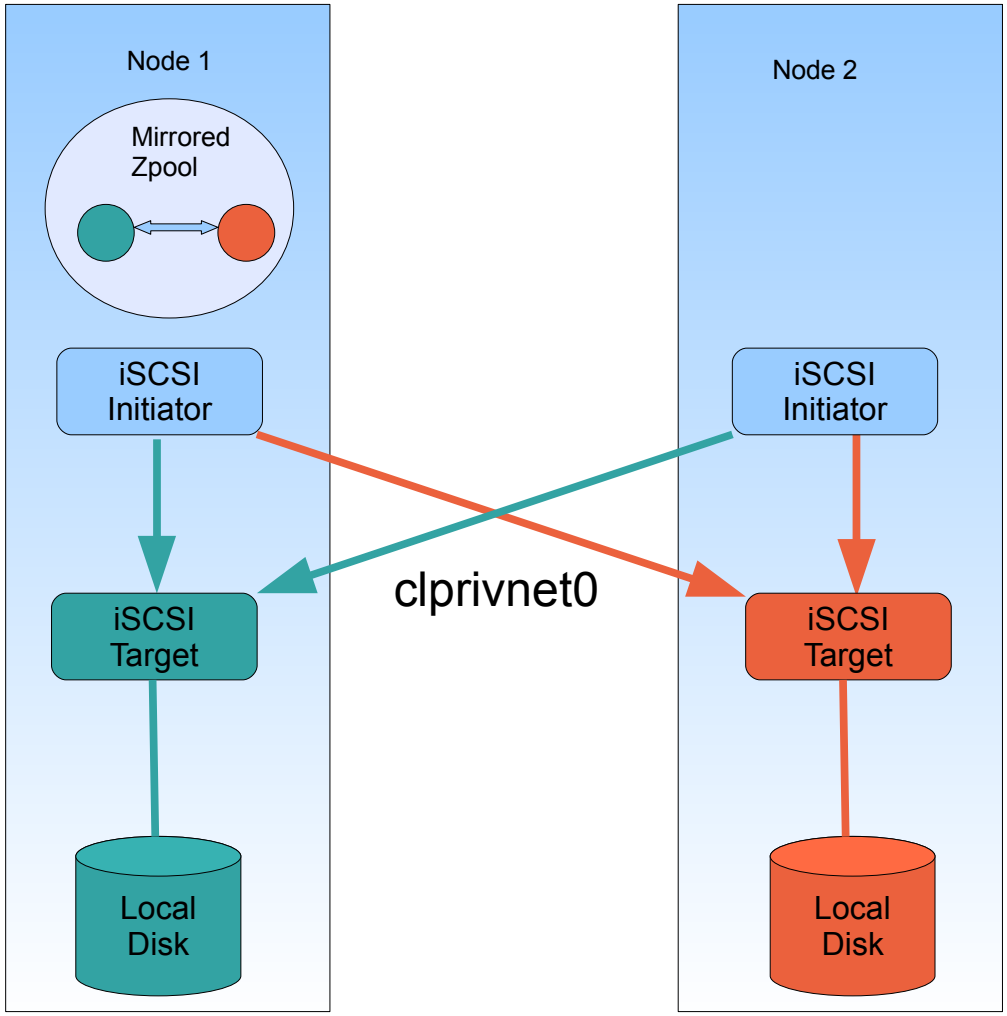
iSCSI Storage

- IP-based storage networking standard
- iSCSI initiators (clients) senden SCSI Befehle zum iSCSI target (storage devices) über reguläres IP Netzwerk
- Alternative zu NAS, SAN und DAS
- OpenSolaris Common Multiprotocol iSCSI Target (COMSTAR) implementiert u.a. das iSCSI Protokoll

COMSTAR iSCSI für OHAC 2009.06

- Jeder Knoten exportiert die lokale Festplatte als iSCSI target
- Jeder Knoten greift auf beide Festplatten als iSCSI initiator zu
- zpool Spiegelung der iSCSI targets
- HAStoragePlus importiert zpool auf dem Knoten der den HA Dienst anbieten soll
- Wenn ein Knoten ausfällt, bleibt die lokale Hälfte des anderen Knoten verfügbar und benutzbar

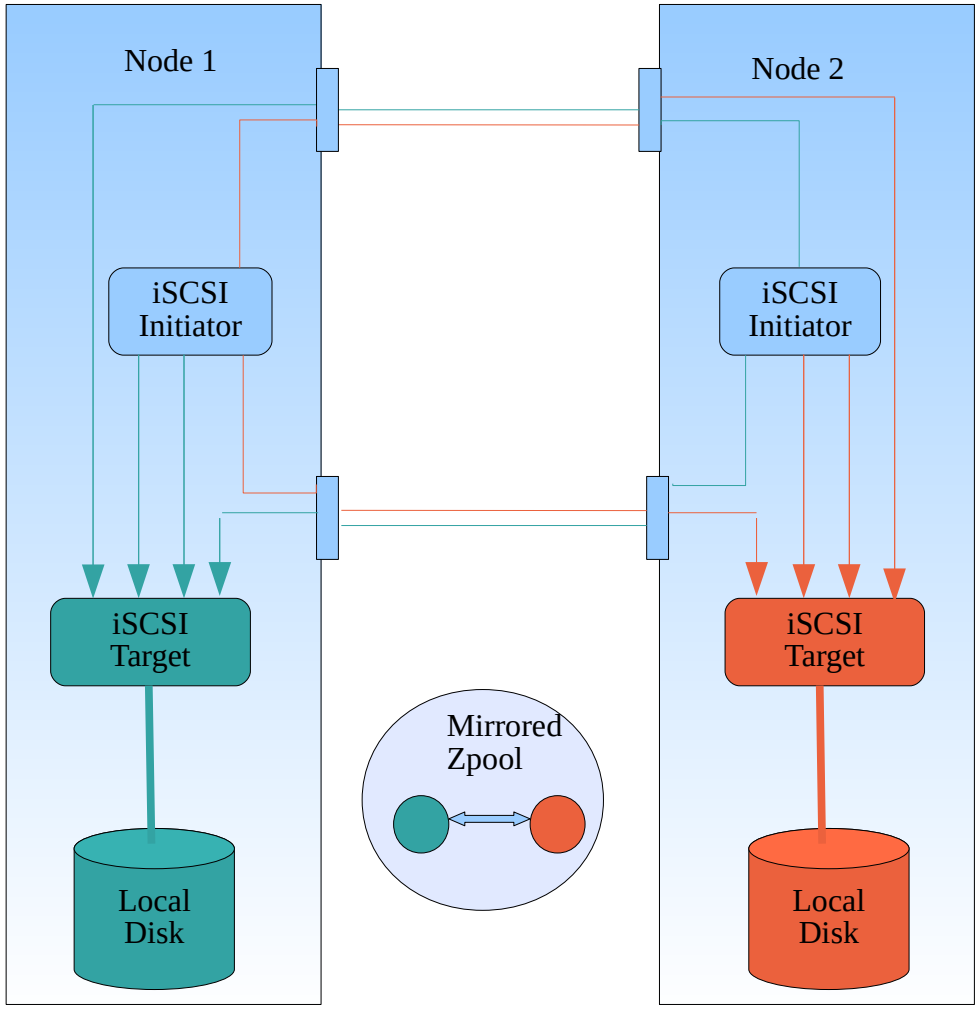
COMSTAR iSCSI Konfiguration



iSCSI mit MPxIO und OHAC 2009.06

- OpenSolaris Storage Multipathing
  - Mehrere redundante Pfade zum Plattenspeicher
  - Arbeitet über der Transportschicht
- Konfiguration von MPxIO über die redundanten Pfade des privaten Interconnects (braucht strong membership)
- Vorteile von MPxIO mit iSCSI in OHAC
  - unterstützt RDMA (Infiniband)
  - Round-robin load balancing zur Erhöhung des IO Durchsatz

iSCSI mit MPxIO Konfiguration



Crossbow VNICs

- Virtual Network Interface Card (VNIC)
- Pseudo-network interface
- VNICs sind über physikalischen Netzwerkadaptern konfiguriert

```
# dladm create-vnic -l e1000g0 vnic1
```

Crossbow VNICs with OHAC 2009.06

- Cluster private interconnect kann VNICs als Endpunkte benutzen, statt expliziter physikalischer Netzwerkadapter
- Funktioniert auch über die physikalischen Adapter des externen Netzwerks
- Nutzung von IPsec um den privaten Cluster Datenverkehr zu schützen
  - schützt allerdings keine DLPI Heartbeats, da unterhalb der IP Schicht

HA Cluster „Strong Membership“

- Konzept: Mehrheitsentscheid um Cluster Konsistenz auch bei Partitionierung in Raum und Zeit sicherzustellen
  - Partition in Raum (Netzwerk Partition) kann zu „split-brain“ führen
  - Partition in Zeit kann zu Amnesie führen
- Zwei-Knoten Cluster benötigt ein drittes Gerät bei Partitionierung
  - Typischerweise SCSI Gerät oder Quorum Server

Weak Membership (preview feature)

- Zwei-Knoten Cluster ohne Quorum Device
- Externe “ping targets” werden kontaktiert als „health check“ im Fall von „split-brain“
  - Worst-case: beide Knoten laufen weiter und bieten Dienste an
  - OpenSolaris Duplicate Address Detection (DAD) kann Wahrscheinlichkeit verringern
- Verfügbarkeit wird höher gewichtet als Datenintegrität
  - kann zu Datenverlust führen

Warum Weak Membership benutzen?

- Read-only oder read-mostly Anwendungen
- Verfügbarkeit ist wichtiger als Datenintegrität
  - das SLA passt (Lösung ist gut genug)
- Test Cluster mit limitierten Mitteln
- Demos
- Entwicklung
- Training

OHAC 2009.06 Software Modularisierung

- ha-cluster-full Gruppenpaket
  - Inhalt: Core Framework, Wizards, Agenten, man pages, I10n, … (alles)
- ha-cluster-minimal Gruppenpaket
  - lediglich das Core Framework
  - Agenten, Wizards, I10n, man pages, telemetry, etc. bei Bedarf hinzufügen
- Installation von Quorum Server and Agentbuilder ohne Core Framework

Warum minimale Installation nützlich ist

- Minimierung benötigter Betriebsmittel (man bezahlt nicht was man nicht braucht)
  - Plattenplatz
  - Netzwerk Bandbereite beim runterladen
  - etc.
- Security minimization
- Minimierung des administrativen Overhead
  - beides - initial und fortlaufend

Installation Open HA Cluster 2009.06

- Nutzerbedingung auf pkg.sun.com akzeptieren und runterladen von Schlüssel und Zertifikat nach /var/pkg/ssl
- ha-cluster publisher (auf allen Knoten) konfigurieren:

```
# pkg set-publisher  \
  -k /var/pkg/ssl/Open_HA_Cluster_2009.06.key.pem \
  -c /var/pkg/ssl/Open_HA_Cluster_2009.06.certificate.pem \
  -O https://pkg.sun.com/opensolaris/ha-cluster/ ha-cluster
```

Installation OHAC 2009.06 (cont)

- Installation der Cluster Software (alle Knoten)

  # pkg install ha-cluster-full

- Konfiguration des Clusters (auf einem Knoten)

  # /usr/cluster/bin/scinstall

Live Demo

- Toshiba M10
  - 4 GB main memory
  - 160 GB hard disk
  - OpenSolaris 2009.06
  - Open HA Cluster 2009.06
  - VirtualBox 2.2.4

Referenzen (1)

- Open HA Cluster 2009.06 Dokumentation
  - http://www.opensolaris.com/learn/features/availability/
  - http://docs.sun.com/app/docs/prod/open.ha.cluster~2509.1#hic
- Solaris Cluster Blog
  - http://blogs.sun.com/SC
- White Paper: Running Open HA Cluster on OpenSolaris with VirtualBox
  - http://opensolaris.org/os/project/colorado/files/Whitepaper-OpenHAClusterOnOpenSolaris-external.pdf

Referenzen (2)

- HA Clusters Community Group
  - http://opensolaris.org/os/community/ha-clusters/
- Projekt Colorado
  - http://opensolaris.org/os/project/colorado/
- Projekt Image Packaging System (IPS)
  - http://opensolaris.org/os/project/pkg/
- Projekt Crossbow (VNICs)
  - http://opensolaris.org/os/project/crossbow/
- Projekt COMSTAR (iSCSI)
  - http://opensolaris.org/os/project/comstar/

Open HA Cluster

Vielen Dank!
Fragen?

thorsten.frueauf@sun.com
http://blogs.sun.com/tf

Thorsten Früauf
Availability Engineering
Sun Microsystems GmbH