Consolidation and Resource Management

Adrian.Cockcroft@sun.com

Contents

- The Resource Management BluePrint
- Service Level Management
- Systems Consolidation and Resource Management
- Measurements, Policies and Controls
- Resource Management Products
- Directions

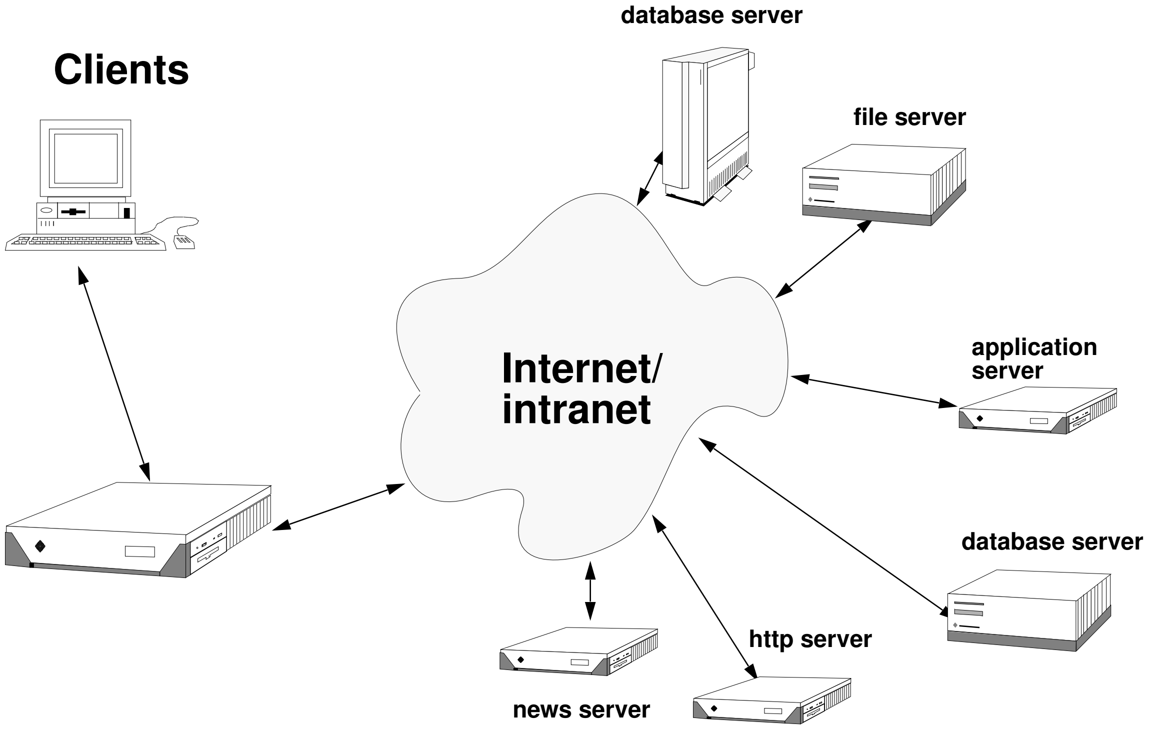
Sun's BluePrint Program

- Similar to the IBM Redbook concept
- Best practices information
- Combinations of products and techniques
- First BluePrint is on Resource Management
- Approximately 500 pages
- Deliver final version to customers next summer
- Please give me feedback on what you want

Service Level Management

- Define the service and the goals
- Measure the service level
- Apply goal-driven policies
- Use controls to change the service level
- Feed into accounting and capacity planning

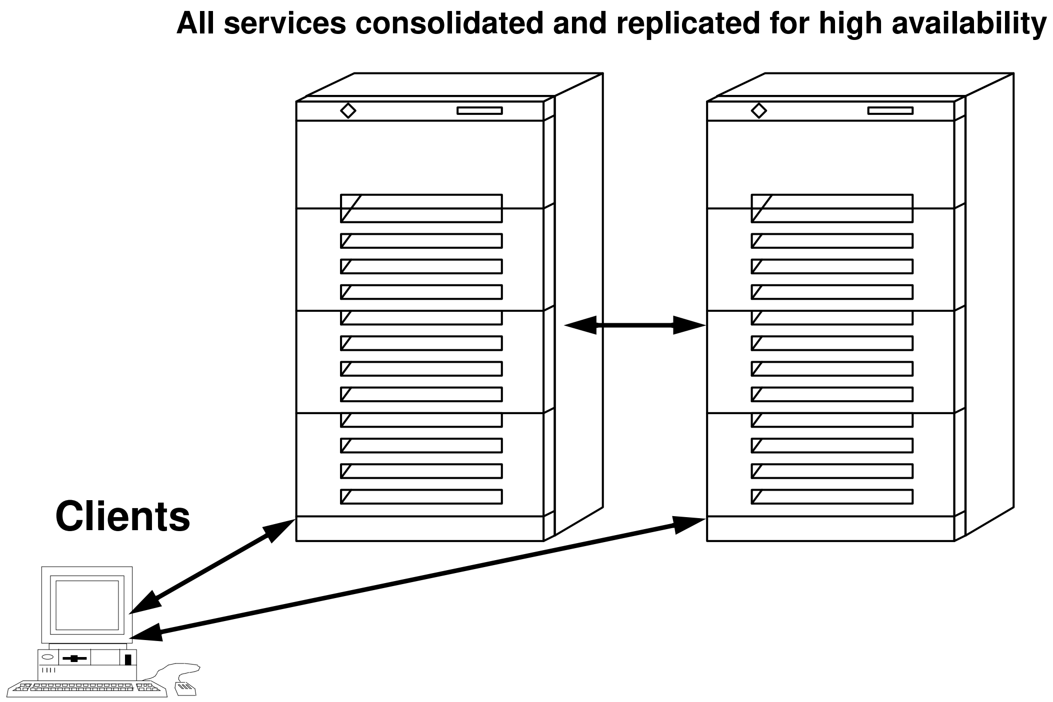
Distributed Systems Service Level Management



Distributed Systems Issues and Solutions

- One system per application or service
- System level measurements can be used
- Good performance and fault isolation
- Hardware changes and upgrades control resources
- Policies implemented by hand in hardware
- Easy to manage each individual system
- Too many systems to manage
- Wasted un-sharable resources

Consolidated Systems Service Level Management



Consolidated Systems Issues and Solutions

- Need to break down process activity to get per-application measurements
- Security and RAS are combined, harder to limit partial access or failures
- Few boxes to manage, centralized management tools
- Good resource usage due to sharing
- Poor isolation against common mode problems
- Few Solaris 2 controls are available
- Need to automate policies in software

Workload Analysis

- All systems run a mixture of workloads
- Pick out relevant processes and aggregate
- The rest is overheads or unwanted/unplanned activity
- Include network workloads as well

Tools

- BMC Best/1, TeamQuest, Metron for processes
- Sun Bandwidth Allocator for network accounting
- Resolute RAPS for integrated protocol analysis

Accounting

Use accounting extended to break down by workload
Monthly trends, normalized to working hours/month
Standard accounting and extensions
Develop extended accounting & ps tools
Integrate ps measurements into acct data
Take top N processes/workloads or top N% of system, accumulate over months
Plot the results

Policies

Business driven

- Prioritize applications or groups of users

Funding driven

- Get what you pay for when funding is shared

Security driven

- Keep denial of service attacks constrained

Discretionary work

- Run zero share batch with low impact on main workloads to soak up idle time

Exceptions and rules

- Need to integrate management automation frameworks

Solaris standard tools

limits

- systemwide default limits per process, not very useful

Maximum number of processes per user - maxuprc

- systemwide limit for all non-root users, set in /etc/system

Disk quotas

- BSD based disk quota system, crude but effective

Processor Sets - psrset

- Solaris 2.6 feature allows large systems to be subdivided
- Used for combined TPC-C and TPC-D DB2 benchmark result
- Processes are locked into a set, no sharing of idle time

E10000 Domains

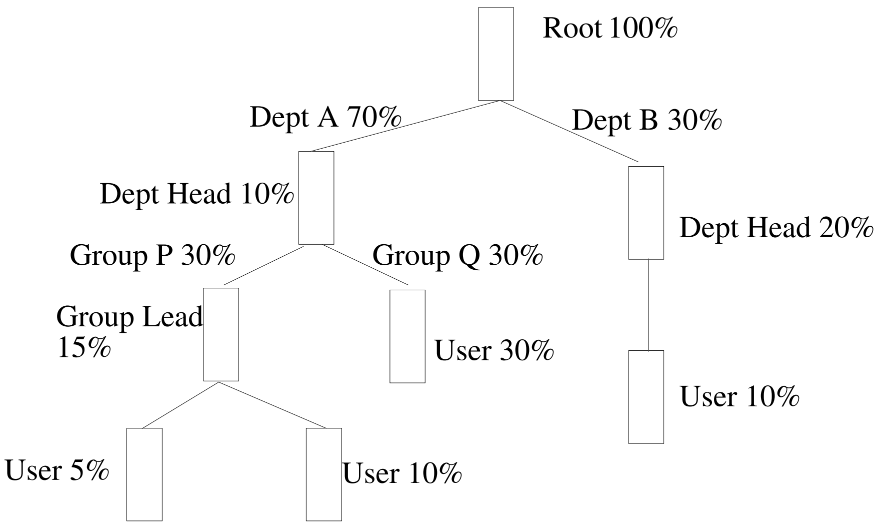
- Dynamically reconfigure hardware over multiple Solaris images

Solaris Resource Manager 1.0

A fully integrated DR-safe update of Softway Share II

- **Hierarchical share based per-user/group controls**
- **CPU Fair Share Scheduler**
- **Virtual memory usage limit (swap space)**
- **Per-user process count**
- **Soft and hard limits**
- **Share II "Disk Domain" quotas are not part of SRM 1.0**
  - **Filesystem quotas are being re-examined in general**
  - **Use the existing BSD quota system for now**

Share Hierarchy



Fairness

Time sharing

```
USER        PID %CPU %MEM   SZ  RSS TT       S    START  TIME COMMAND
user1      1042 84.4  1.7 1280  960 pts/0    R 16:42:57  8:22 job8
user2      1043 10.6  0.7  856  360 pts/1    R 16:42:58  1:00 job1
```

Fair share scheduling

```
USER        PID %CPU %MEM   SZ  RSS TT       S    START  TIME COMMAND
user2       253 51.3  0.7  856  360 pts/0    R 17:44:34  3:06 job1
user1       254 44.3  1.7 1280  960 pts/1    R 17:44:35  2:42 job8
```

Job1 has 1 thread and Job8 has 8 threads. With FSS, each user gets about half of the CPU time

Denial of Service

The problem with timesharing

User1 overloads the system with jobs, User2 gets only 1% of the CPU, root takes 7.9s to run the ps command

```
USER        PID %CPU %MEM   SZ  RSS TT       S    START  TIME COMMAND
user1      1153  9.9  1.7 1280  960 pts/0    R 17:01:04  0:46 job8
user1      1154  9.7  1.7 1280  960 pts/0    R 17:01:06  0:45 job8
user1      1151  9.6  1.7 1280  960 pts/0    R 17:01:00  0:46 job8
user1      1148  9.5  1.7 1280  960 pts/0    R 17:00:58  0:45 job8
user1      1152  9.4  1.7 1280  960 pts/0    R 17:01:02  0:44 job8
user1      1146  9.4  1.7 1280  960 pts/0    R 17:00:56  0:45 job8
user1      1149  9.4  1.7 1280  960 pts/0    R 17:00:58  0:44 job8
user1      1150  9.4  1.7 1280  960 pts/0    R 17:00:59  0:44 job8
user1      1145  9.3  1.7 1280  960 pts/0    R 17:00:56  0:46 job8
user1      1144  9.2  1.7 1280  960 pts/0    R 17:00:56  0:46 job8
user2      1147  1.2  0.7  856  360 pts/1    R 17:00:56  0:05 job1

real        7.9
```

Service Maintained With Identical Load

With SRM 1.0, processes share resources the way you tell them to. Here each user gets half the system, root always gets top priority so ps is instant.

USER	PID	%CPU	%MEM	SZ	RSS	TT	S	START	TIME	COMMAND
user2	428	44.3	0.7	856	360	pts/0	R	17:51:48	1:58	job1
user1	427	5.2	1.7	1280	960	pts/1	R	17:51:47	0:12	job8
user1	426	4.9	1.7	1280	960	pts/1	R	17:51:47	0:12	job8
user1	422	4.8	1.7	1280	960	pts/1	R	17:51:47	0:12	job8
user1	425	4.8	1.7	1280	960	pts/1	R	17:51:47	0:12	job8
user1	423	4.8	1.7	1280	960	pts/1	R	17:51:47	0:12	job8
user1	429	4.7	1.7	1280	960	pts/1	R	17:51:48	0:12	job8
user1	424	4.6	1.7	1280	960	pts/1	R	17:51:47	0:12	job8
user1	431	4.5	1.7	1280	960	pts/1	R	17:51:51	0:12	job8
user1	421	4.4	1.7	1280	960	pts/1	R	17:51:47	0:12	job8
user1	430	4.4	1.7	1280	960	pts/1	R	17:51:49	0:12	job8

```
real        0.0
```


Proportional Sharing

user2 is five times more important than user1, so gets the CPU power even when competing with 8 threads

USER	PID	%CPU	%MEM	SZ	RSS	TT	S	START	TIME	COMMAND
user2	472	79.6	0.7	856	360	pts/0	R	17:58:30	5:00	job1
user1	473	16.5	0.7	856	360	pts/1	R	17:58:31	0:59	job1

USER	PID	%CPU	%MEM	SZ	RSS	TT	S	START	TIME	COMMAND
user2	323	74.4	0.7	856	360	pts/0	R	18:40:39	4:35	job1
user1	324	21.2	1.7	1280	960	pts/1	R	18:40:40	1:29	job8

Sun Bandwidth Manager

Allows server network bandwidth to be controlled

- by application
- by traffic type
- by customer

Accounts for bandwidth usage

- Web flow accounting capability
- Charge for network usage by customer
- Accounting-only mode - secure and low overhead monitoring

Java based agent and GUI

- Based on the Java Device Management Kit (JDMK)
- Graphical monitoring displays
- Remote configuration, portable GUI

Future Development Issues

Please give me feedback on priorities and ideas

Distributed policy management

- Central management of SRM nodes
- Integration of Bandwidth Manager, SRM and other policies

Problems with pools of memory

- Real physical memory is shared at present
- Current page scanner doesn't like huge memory systems
- We are working on re-architecting the whole approach
- priority paging fix in 2.7 and 2.6 with 105181-09 helps somewhat

Do we also need to control disk I/O throughput?

- How would you use an I/O share or I/O accounting logs?

What else needs to be controlled?

Summary

- Tool set and issues change when moving from distributed to consolidated workloads
- Don’t go too far, balance admin centralization against the extra management problems it causes
- Solaris will have more resource controls, improved policy management and better measurements as time goes on