Sun Microsystems' *AutoClient* and management of computer farms at *BaBar*

- Modern HEP experiments require immense amounts of computing power

BaBar : mostly Solaris SPARC

- **Analysis and Prompt Reconstruction farm: 300+ nodes**
- **Online Data Flow farm: 79 nodes**
- **Control Room Console farm: 16 nodes**
- **User workstations, servers, common use machines, etc.**

(and that's just at SLAC)

- Systems performing tasks critical to experiment operation require minimal downtime in case of a hardware malfunction, power outage, software upgrade, etc.,

BUT...

Unless a special system maintenance scheme is devised, the required administration effort scales *linearly* with the number of machines and becomes unbearable:

Setting up a *Solaris* stand-alone, customizing it to make best use of the SLAC computing environment:

Brute force approach:

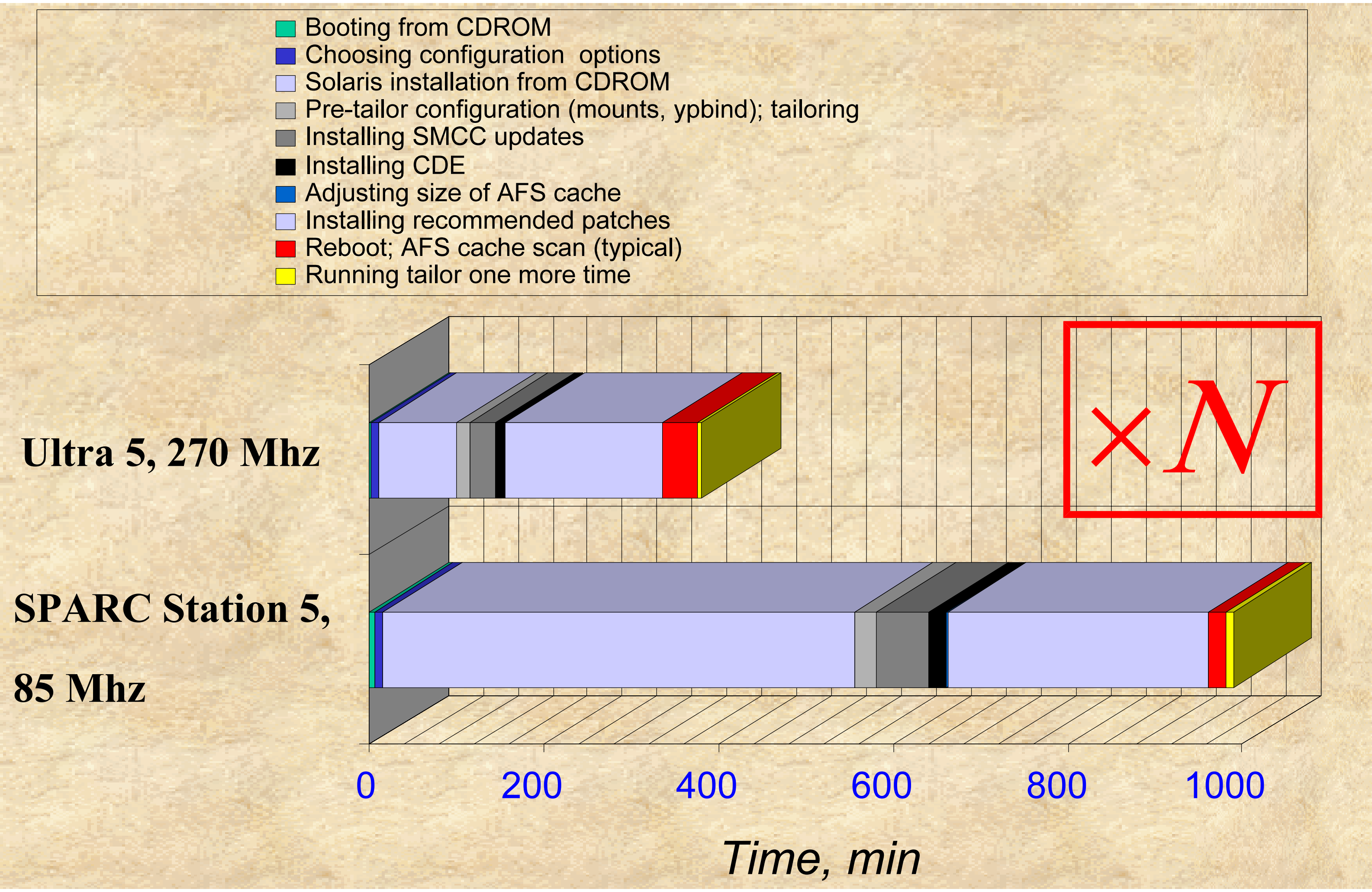
~ 10 hours, ~ 3 hours of active administrator involvement

Same with `tailor`:

3-5 hours, ~ 1 hour of administrator involvement; automation and centralization of many routine system maintenance tasks

It's a lot of time ×N !

Setting up a standalone system with help of `tailor`:



[timing is for *Solaris* 2.5.1. HW 11/97 (MU5); *Solaris* 2.6 installs faster]

Alternatives to "brut force" deployment of Solaris systems:

Cloning (`dd, cpio,ufsdump/ufsrestore,` or `tar`; can be made "non-invasive" by cloning while booted as a Diskless Client):
~20 minutes

JumpStart (with custom install/finish scripts):
~1 hour

Diskless Client : ~5 minutes, but very poor performance

AutoClient : ~5 minutes

* AFS cache initialization time is an extra 30 to 60 minutes

System types supported in *Solaris*

System Type	**Local File Systems**	**Local Swap?**	**Remote File Systems**	**Network Use**	**Relative Performance**
Server	root (/) /usr /home /opt /export/home /export/root	Yes	*-none-*	Medium	High
Standalone System	root (/) /usr /opt /export/home	Yes	*-none-*	Low	High
Dataless Client (removed in Solaris 2.6)	root (/)	Yes	/usr /home /opt	Medium	Medium
Diskless Client	*-none-*	No	root (/) swap /usr /opt /home	High	Low
AutoClient System	cached root (/) cached /usr cached /opt	Yes	root (/) /usr /opt /home	Low	High

AutoClient basics

- No persistent data stored locally : root (`/`), shared read-only `/usr` and `/opt`, etc. reside on the server, so the machine is a *"field-replaceable unit"* (FRU)
- All data is on the server and can be manipulated from the server; only the server has to be backed up
- root (`/`), `/usr`, `/opt` are locally cached using *CacheFS*, `swap` is local, and AutoClients reboot from the cache
- There is no noticeable performance deterioration or increase of network load compared with a Solaris stand-alone
- Very modest server hardware requirements
- AutoClients can be halted and rebooted remotely

AutoClient basics (cont'd):

- *CacheFS* consistency check every 24 hours, on reboot, or at request
- All writes immediately update the back file system, but the *'disconnectable'* option allows AutoClients to function if the server is temporarily unavailable
- Specific files or directories can be *'packed'* into the cache to guarantee their presence there
- Replacing a failed unit or deploying a new AutoClient takes just a few minutes
- Many management tasks normally associated with stand-alone *Solaris* systems are thus almost completely eliminated

The quintessence of the centralized administration model, of which AutoClient *is a key component, is a significant reduction of system management efforts and costs*

... that is, if everything works as advertised!

BaBar's experience with *AutoClient*

- Began experimenting with *AutoClient* in June 1998
- Developed a version of `tailor` that works with AutoClients and a set of scripts that 'clone' AutoClient root file systems
- Winter 1998/99 cosmic ray run:

  1 server, ~25 console and Online Data Flow nodes
- Since May 1999:

  Analysis & OPR: 309 AutoClients and 6 AutoClient servers (with `tailor`)

  Consoles, ODF: 100 AutoClients, 1 AutoClient server (w/o `tailor`)

Our impressions after operating over 400 AutoClients for 8 months under real life conditions:

Overall, the farms performed their duties very well

However, the required management effort turned out to be much greater than expected

Prime reason: a bug in *CacheFS* (fixed by Sun for *Solaris 7*, but not *Solaris 2.6*) causing cache corruption in 15-20% of AutoClients during power or network outages, or AutoClient server crashes

> Such outages have been happening about once a week; each time status of each AutoClient had to be checked manually to make sure they were OK; affected AutoClients had to have their caches reconstructed; sometimes, they had to be re-cloned.

Our actions

- connected servers and networking equipment to uninterruptible power
- changed network topology
- working with Sun on getting the bug fixed
- moving AutoClient services from the main online file server to a dedicated machine

Another problem:

Patches and software updates cannot be safely applied to active AutoClients without a reboot; a global farm outage must be scheduled to patch `/usr`

Conclusion

We see a great potential in *AutoClient*, but unless the *CacheFS* bug is fixed in the nearest future, we will be forced to fall back to a more classical approach: *JumpStart* + **`tailor`**, and see how the required system management effort compares to using *AutoClient*.