



VMware Build Group

Perforce Use Model

Perforce User Conference

04/14/05

- What do we use Perforce for?
 - Development community
 - Branching policies
 - Merging policies
 - Extra wrapper functionality
 - Locking policies
 - Routine maintenance
 - Client cleanup
 - User supervision
 - Backups/Recoveries
 - Areas for improvement

How have we grown?

	09/2002	09/2003	09/2004	04/2005
Licenses	90	180	320	520
Clients	500	1900	2600	3200
CKP.gz	210M	440M	980M	1400M
Depot	8G	14G	22G	43G

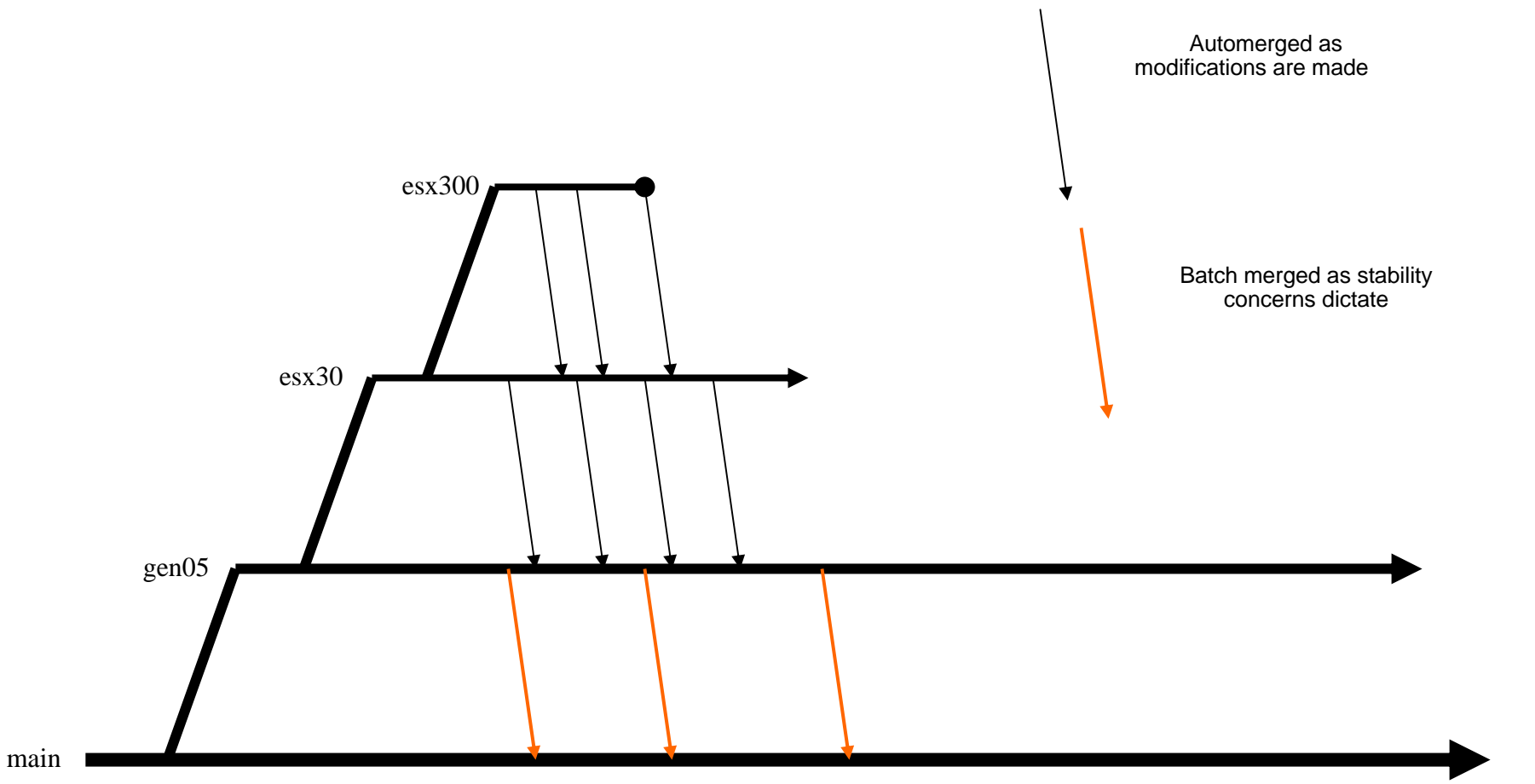
What do we support?

- 5 products
 - ESX Server (Linux server product)
 - VirtualCenter (Windows hosted product)
 - GSX Server (Linux & Windows hosted product)
 - Workstation (Linux & Windows hosted product)
 - Ace (Linux & Windows hosted product)
 - P2V Assistant (Windows hosted product)
- 2 or 3 active branches, each releasing monthly
- Averaging 12 releases/month

- Need to Support:
 - Major Release, currently expected at 18 month intervals
 - Minor Release, features and bug fixes, expected at 9 month intervals
 - Quarterly Rollup incorporating cumulative bug fixes (every 3 months)
 - Monthly hotfix patch (as needed). No additional functionality
- Examples:
 - ESX 2.0 Major Release
 - ESX 2.1 Minor Release
 - ESX 2.1.1 Quarterly rollup
 - ESX 2.1.1.1 Hot fix as needed

- Considerations:
 - 40% code shared among all major products
 - Extremely tight integration between certain products
- Source Code Organization
 - Essentially a “Big ball of mud”
 - Currently undergoing an accepted, prioritized, and championed restructuring.
 - Who knows when it will be done
- Private, Project, and Official branches

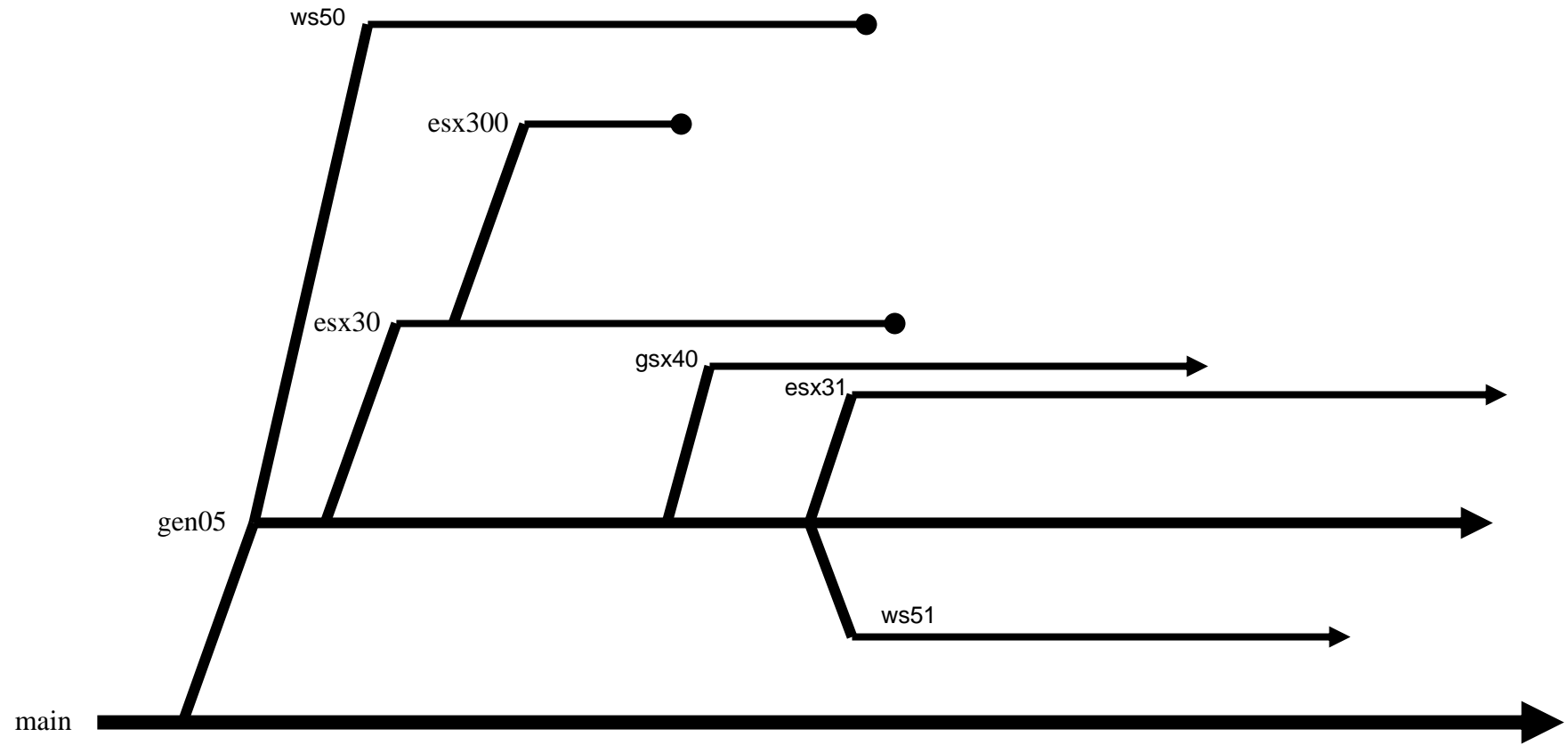
- Reality Support requirements:
 - Hot fix (2.1.1.1) (Insulated)
 - Next Maintenance release (2.1.2) (Insulated)
 - Next Minor Release (2.2.0) (Insulated)
 - Next Major Release (3.0)
 - Advanced Development (Isolated)
- Merge Simplicity
 - Keep products of same generation together and easy to merge between.



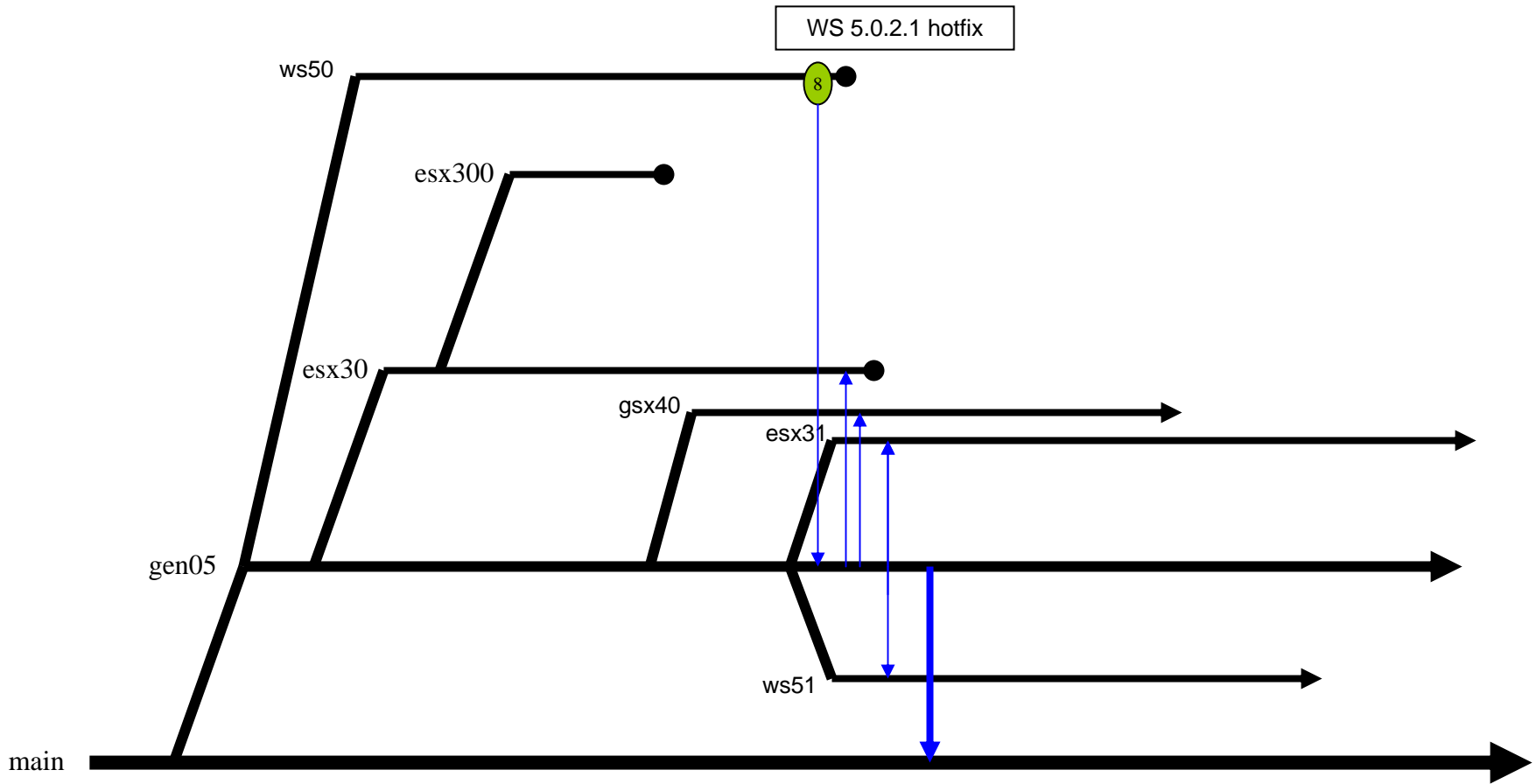
Automerged as modifications are made

Batch merged as stability concerns dictate

Multiple Products



Merging (complex)



- Automerge – What?
 - Transparent merge across multiple branches
- Automerge - Why?
 - Reduce regressions
 - Quickly propagate your code through all associated branches
 - Eliminate manual crossporting
 - Preserve functionality

- Automerge
 - Defined by branch description and management policy
 - Warns if merge requires manual intervention
 - Nags if you ignore the manual intervention mail
 - Controlled through Submit Template
 - Multiple Policies:
 - On Merging will occur if YES is provided
 - Selective Merging will only occur when NOW is provided, YES requests will be deferred until Policy is changed to On
 - Deferred Merging is not occurring but in the future will and YES merges will be processed at that time
 - Off Merging will NOT occur for these branches
- Master control file
 - Merge-<from branch>-<to branch>
 - Policy
 - Default Value

- Automerge.pl
 - Invoked from client aware of target branch
 - Runs a simulated integrate
 - Parses output identifying changesets to be applied
 - For each changeset to be applied, parses description to see if merge is requested.
 - If merge requested, and merges enabled, then applies merge and submits.
 - Guts of changeset identification stolen from public depot (sorry, can't remember who or what it was originally)
 - Available in my public depot (daf)



```

# A Perforce Change Specification.
#
# Change:      The change number. 'new' on a new changelist.  Read-only.
# Date:       The date this specification was last modified.  Read-only.
# Client:     The client on which the changelist was created.  Read-only.
# User:      The user who created the changelist.  Read-only.
# Status:    Either 'pending' or 'submitted'.  Read-only.
# Description: Comments about the changelist.  Required.
# Jobs:      What opened jobs are to be closed by this changelist.
#            You may delete jobs from this list.  (New changelists only.)
# Files:     What opened files from the default changelist are to be added
#            to this changelist.  You may delete files from this list.
#            (New changelists only.)

Change: new

Client: daf-esx25

User:   daf

Status: new

Description:
#      Instructions:
#      1) Use the first line wisely; p4 changes will use it as an overview.
#      2) Prefix each description line with whitespace (space or tab)
#      <put a good 1-line overview here!>

#      <put your Advil-compliant review request here!>

QA Notes:
Bug Number:
Reviewed by:
Testing done:
Breaks vmcore compatibility:
Mailto:
Merge to: main: YES # Merges (from esx20) enabled
Merge to: esx-escher: YES # Merges enabled
Merge to: esx20: YES # Merges (from esx-escher) enabled

Files:
//depot/bora/esx25/Makefile      # edit
    
```



- p5 - Perl Wrapper around p4 that does what 'we' think p4 should.
- p5 is a command line tool
 - Windows
 - Linux
- Started as CVS conversion helper

- p5 functionality
 - p5 backout <CLN>
 - Backs out a single committed change
 - Only if nothing has been checked in on top of these files
 - p5 crossport <CLN> <srcBranch> <dstBranch>
 - Integrates CLN submitted to <srcBranch> into <dstBranch>
 - You still need to run “p5 resolve” and “p5 submit” to complete the change
 - Possible only between child/parent branches
 - p5 addcheck
 - Displays files not maintained by Perforce
 - p5 churn
 - Shows the number of changed lines of code over a given time period

- Branch obsoletes/Inactives
 - Protection table
- Release locks
 - Triggers
 - DB supported

- General Cleanup
 - Bi-monthly
 - Database rebuilt
 - Monthly
 - Any client not accessed within 6 months, owner gets notified and asked whether client can be removed
 - Generally 30% of clients are obsolete*
 - Generally 10% of all clients get removed as a result.*
 - Considering harsher notices*
 - Pending changes older than 3 months are identified and owners notified
 - CleanClients.pl in public depot

- General Cleanup
 - Daily User validation
 - Users who have been dormant for more than one month
 - Users without NIS accounts
 - Users with read-only access
 - “validate_p4users” in the public depot.

- Database backups
 - Full Checkpoint done twice Sunday and Thursday at 12:10 AM.
 - After rebuild takes 70 minutes. When it gets to two hours, we rebuild
 - Journal truncation/archival done all other nights at 12:10 AM.
 - Checkpoint/Journal files stored on two different servers, physically separate.
 - “backup_p4” available in public depot
- Depot backups
 - Full backup done Sunday mornings at 12:30 AM
 - Incremental dumps done other mornings at 12:30 AM
 - Handled by IT group ... Yeah, it worries me.

- Hot backups
 - Duplicate Perforce server with duplicate license file
 - Built from checkpoint file originally
 - Primary server has journal files truncated/preserved hourly
 - Journal files copied to duplicate server
 - Depot files rsynced hourly
 - Disabled currently (performance reasons)
 - “hot_backup_p4” in public depot

- Common Client
 - Size was causing p4d to lock since memory footprint was huge
- Protection table
 - Too many groups
 - Hard to compartmentalize access areas
 - Noticeable time delay in parsing
- Branching model
 - Reliance on main for integration
 - >300 checkins a week
 - Broken 25-50% of the time
- DB Backup
 - Nightly downtime affecting 5-20 users a week
 - Intermediate checkpoint file >20G

Areas for Improvement

- Protection table
 - Need better control with audit trail and history.
- Rename functionality
 - Need to track files that have been moved or reorganized for easier merging.
- Upgrade
 - We're using 2003.2
 - REALLY want to use those new triggers
 - Expect to do so in June.

Public Depot Files

- automerge.pl Handles automatic merges
- validate_p4users Does user monitoring
- CleanClients.pl Identifies obsolete clients
- backup_p4 Backup script
- hot_backup_p4 Provides hot backup

Questions?

