



Parallel Environment Developer Edition

Using PTP as a development environment for a PTP-based product, the IBM Parallel Environment Developer Edition

Brian D. Watt

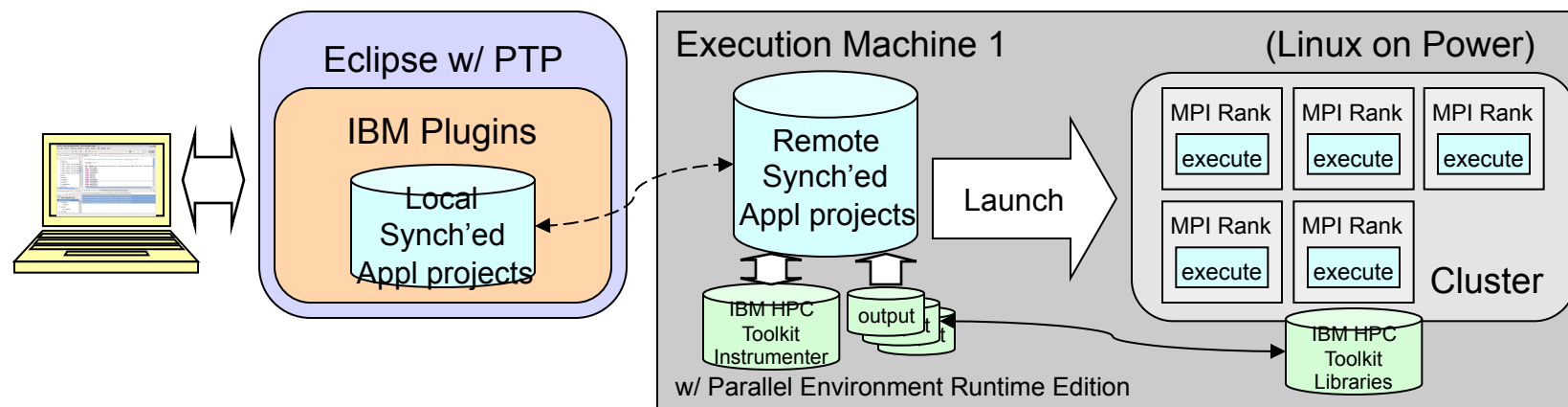
IBM Austin TX

2012 NCSA Eclipse PTP User-Developer Workshop

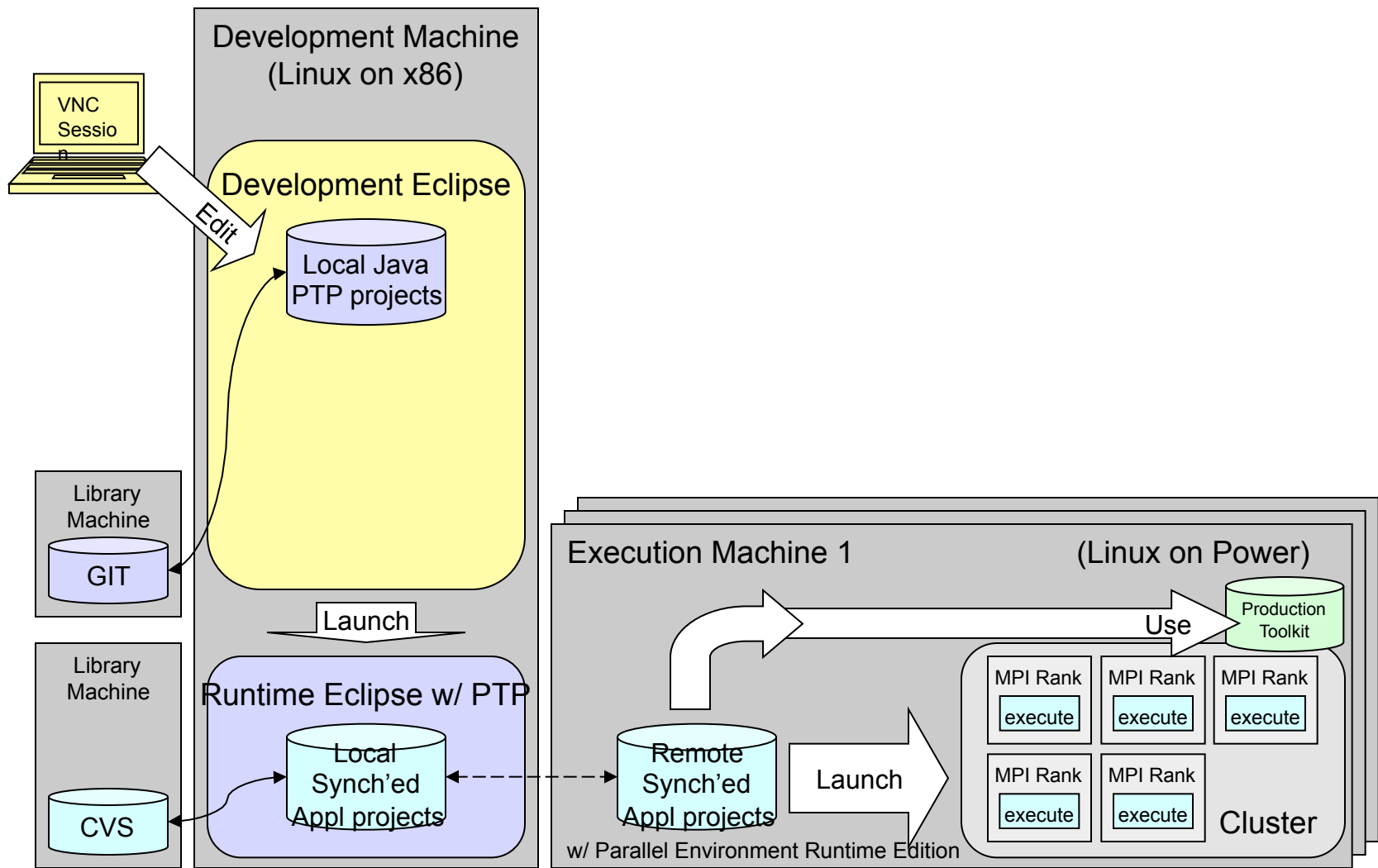
September 18-20, 2012

IBM Parallel Environment Developer Edition Contents

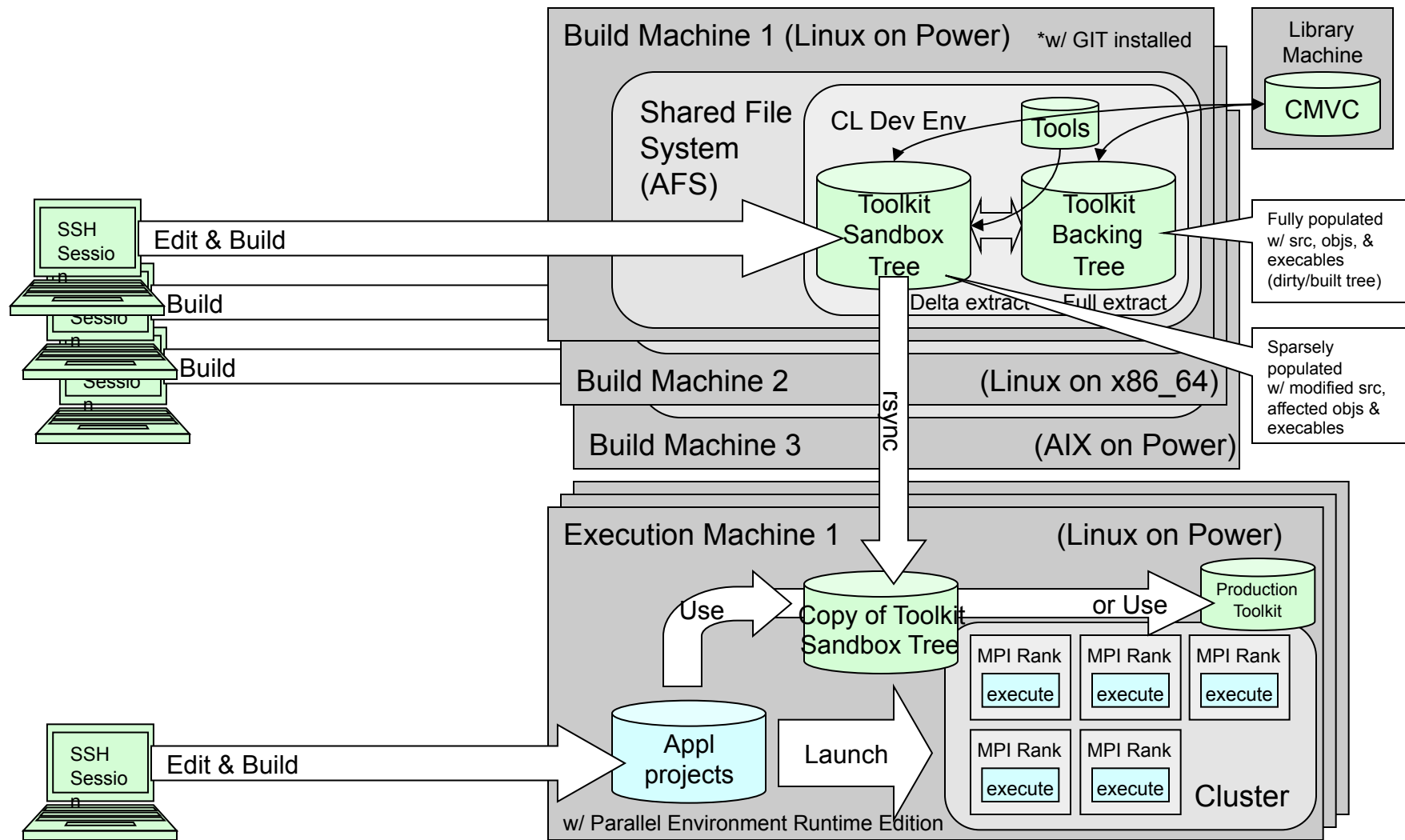
- **Eclipse 4.2 with PTP 6.0**
- **IBM Plugins**
 - PLDT (Programming Language Development Tools) Plugins
 - HPC Toolkit Plugin – HPCT Perspective w/ Profile Launcher
- **IBM HPC Toolkit – instrumenter, supporting libraries and GUI applications written in C/ C++ Language**
 - Hardware performance monitoring using hardware counters
 - MPI performance and tracing
 - OpenMP performance
 - I/O performance and tracing
 - CPU profiling (using -pg/GMON)



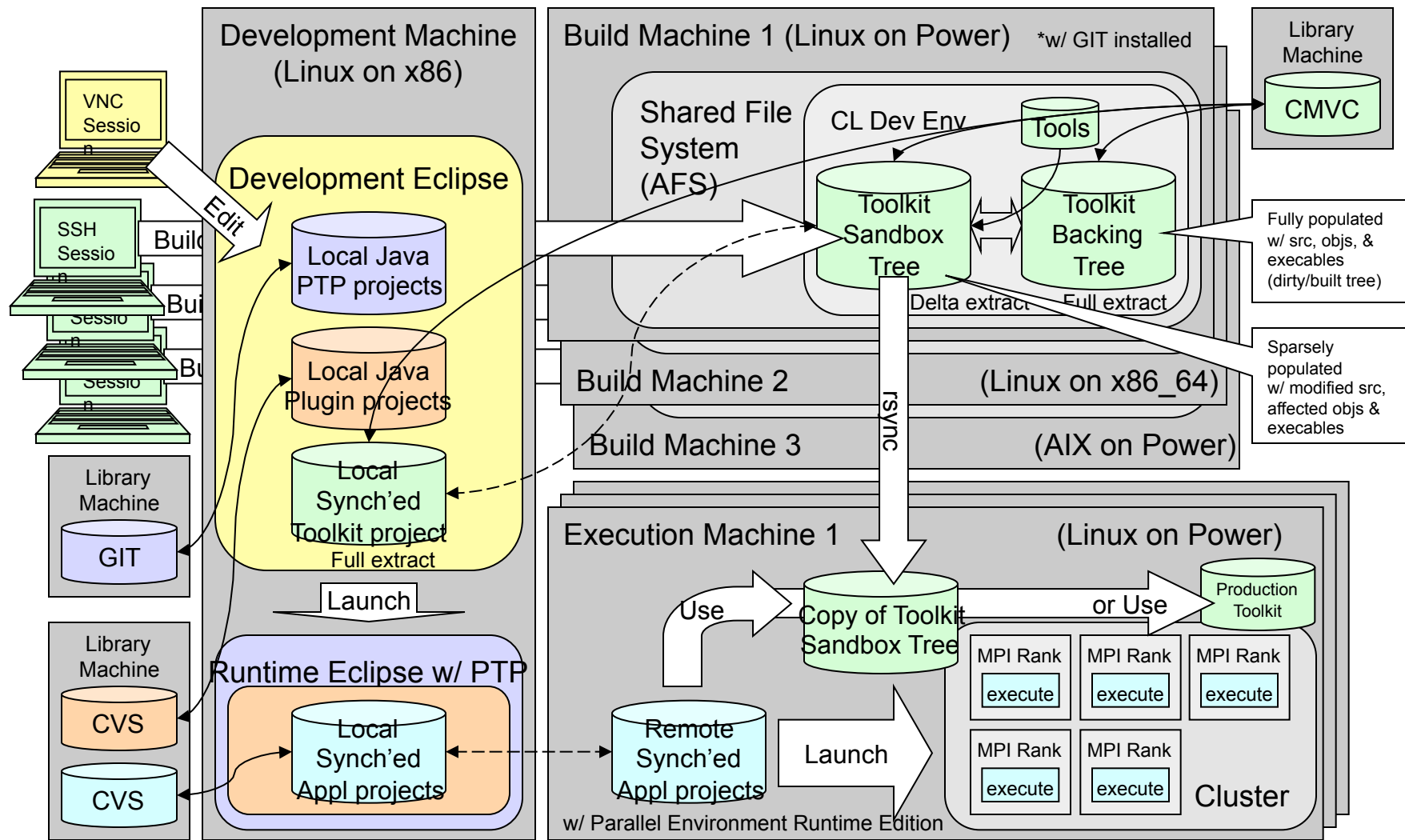
PTP Development Environment



IBM Parallel Environment Developer Edition Development Environment without Eclipse



IBM Parallel Environment Developer Edition Development Environment with Eclipse



Workspace Statistics

- **Number of projects**
 - Eclipse/PTP – too many projects to count, but then you know this already
 - IBM Plugins – approx. 11 projects – IMHO small when compared to PTP
 - IBM HPC Toolkit – one large project with approx. 5700 files
- **Time to synchronize C/C++ project**
 - Approx. 20-25 seconds after every file change
- **Time to build C/C++ project – manually using ssh session w/ commands (not Eclipse build because complex environment variable settings)**
 - Build is done in adjacent build tree separate from source, one per platform
 - Depends upon platform, hardware, etc. – approx from 10-20 minutes - ugh
- **C/C++ project synchronization filters**
 - Only synch files that are in the library (CMVC) – source files
 - 9 filters to eliminate special sandbox files that are not in the library and adjacent build tree - OPTLEVEL_FILES, link, rc_files, src/rules_mk, src/Makeconf, export, obj, ship, inst.images

Strengths

- **A single development environment for all code – not just parallel programs**
- **Multi-language supported: Java, C/C++, and FORTRAN development within one workspace – this product has all three**
- **Multi-project types in a single workspace: Local, Remote (not used), and Synchronized projects**
 - Especially Synchronized projects
- **Multi-libraries supported: CVS, SVN, GIT and CMVC**
- **Multi-level development: The ability for the host Eclipse SDK to launch a target Eclipse and then to use it to work with secondary synchronized Application/Test project which references updated C/C++ code - sweet**
- **Shared File System (AFS) access works**

Weaknesses (my perceived view of them)

- **Team synchronization versus project synchronization is confusing – the models are different, but the words are the same**
- **Synchronization errors with insufficient message details to diagnose issue**
 - Manually (under-the-covers outside of Eclipse) deleting files on local or remote sync projects seems to cause problems, but I haven't had time to isolate the specifics
 - Life saver – delete both .ptp-syncs and re-synch – thanks John, but takes a long long time
- **Seems like too many synchronizations, and I feel I must wait for them all to complete before starting a build using ssh session (I watch the Progress view – it's my new friend)**
- **The length of time it takes to synchronize when there is little or no change**
- **Doing many manual synchronizations (seems that I don't trust it, delay starting)**
- **The new synchronization error view is good, but I was confused about what the various choices meant, and after selecting one then the file was not removed from the view to suggest that the problem was resolved**
- **Inability (or my lack of knowledge) to use Eclipse PTP to BUILD in parallel (or serial) on multiple platforms with current (old) complex environment variable settings with a shared file system**
- **Manually daily klog on development machine to refresh AFS token else things get strange, such as, unexplained sync failures**

Summary

- **PTP is quite usable for developing a multi-platform multi-language PTP-base product**
- **There are rough edges you must accept, but nothing is too sharp as to cut you**
- **I feel as though I'm only using 10% of what I could be using if I really were knowledgeable, but regardless I'm satisfied overall with the experience**