

Plastic SCM 2.0 Features: Distributed System

Plastic SCM distributed system is one of the new features included on the improved Plastic SCM 2.0 version.

This highly useful feature implies once more Plastic SCM freedom of use and customization capabilities, allowing companies to decide among different working methods. As distributed environments are becoming a must for software development, coordinating the efforts of development teams is one of the biggest challenges faced by today's companies. It is hard to imagine any sizable development project managed without a powerful software configuration management tool providing distributed development. Depending on each organization needs and their structure, the most suitable SCM working pattern will be chosen. Plastic SCM supports both centralized and distributed work; let's have a look at the wide range of possibilities provided:

CENTRALIZED WORK

The traditional way of working with a SCM tool, using a single centralized server is called centralized work, which is the method used by companies whose developers working on a project are located in the same building.

Centralized SCM is usually structured as Figure 1 shows: there is a central server which is formed of a repository server; providing access to data as well as creating new repositories and deleting old ones, this is where the databases are stored, and a workspace server; providing different functionalities for managing workspaces: A Plastic SCM workspace is a directory in the user's file system in which repository contents are downloaded, it provides the location used to edit, compile and work with development projects.

Every time that a developer wants to save (check in) his tasks or creates a new working branch, his changes will be kept in the centralized server and when his colleagues update their own workspaces, those changes will be available.

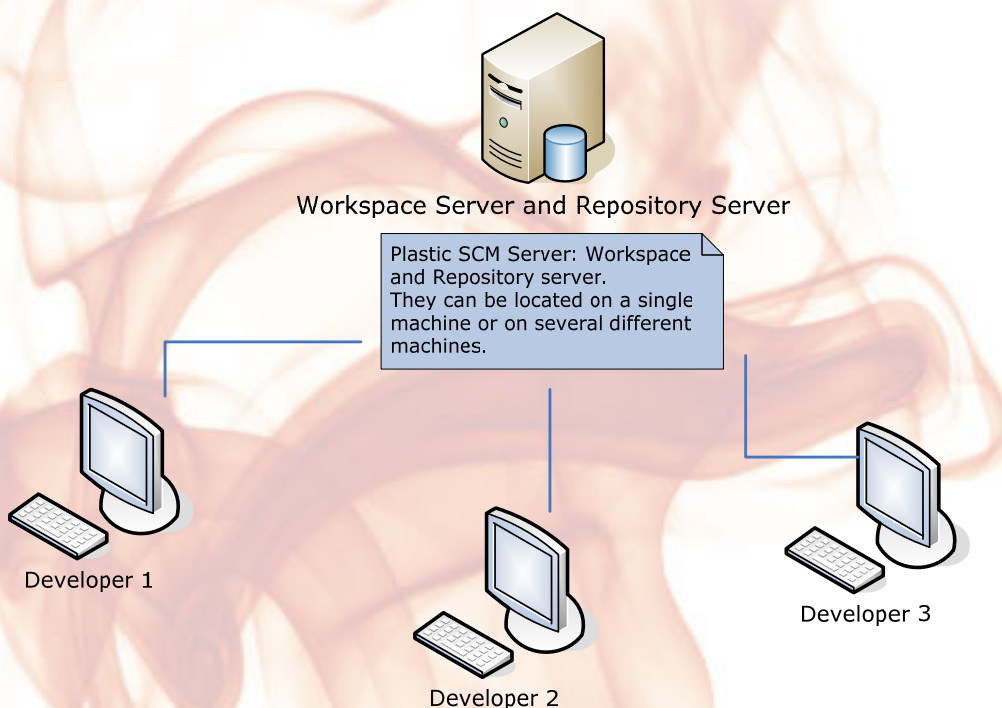


Figure 1: Centralized work

The repository server and the workspace server can be mounted on the same server machine or on different ones, through Plastic SCM multiple server system. Figure 2 shows an example of a company which has one centralized repository server and as many workspace servers as clients, each client machine has its own workspace server installed.

This working method decreases the load on the central server as operations related with the workspace are solved in the local machine, so developers only need to contact with the central server for operations related to the repository. Each client would connect to the repository server in order to carry out the repository related operations, in any other case (checkouts view, workspace management, etc), the user would only need to connect to the workspace server, so the workload on the central server will decrease, and therefore, its performance will increase.

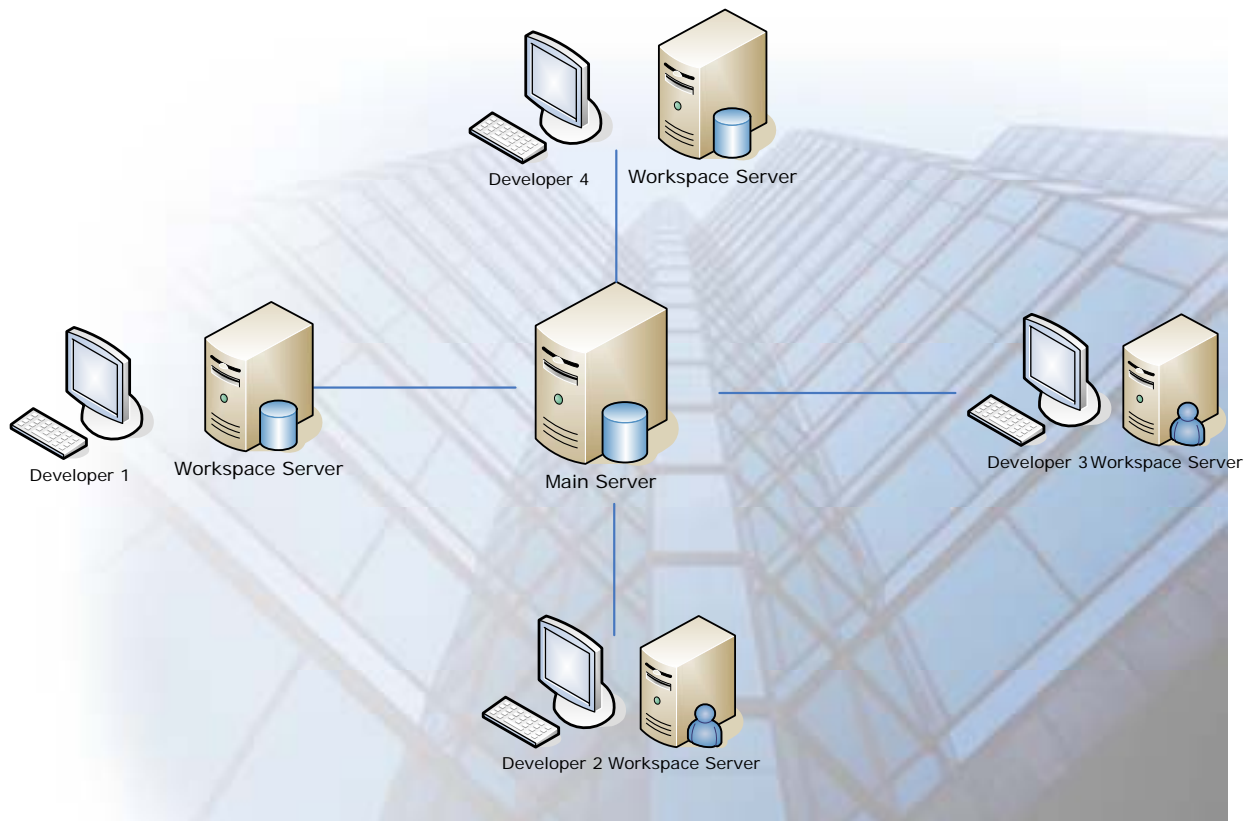


Figure 2: Centralized work with multiple servers

DISTRIBUTED WORK

Nowadays, the constant-changing market is searching for new solutions as global teams of thousands of users, combined with modern methodologies such as agile development, result in processes which are both geographically dispersed and highly interactive, requiring reliable exchange of source code, documents and images between team members, that is why software vendors must align with trends of having globally based developers as well as developers having to cope with multiple releases of software simultaneously.

The basic concept of distributed SCM work focuses on having as many servers as the organization requires depending on its structure, so there would be one source server and several destination servers and through the replicate operation, data will be replicated from one to the others. Which data shall be copied? That would also depend on each company requirements; it could be the whole repository or just one branch (usually br/main).

Even though distributed version control tools have for several years been as robust and usable as their previous-generation counterparts, people using older tools have not yet necessarily woken up to their advantages. There are a number of ways in which distributed tools shine compared to centralized ones.

For an individual developer distributed tools are almost always much faster than centralized ones, which is due to a simple reason: a centralized tool needs to talk over the network for many common operations because most metadata are stored in a single copy on the central

server. On the other hand, distributed tools can store all of their metadata locally (providing a server on each client machine or on each location, depending both on the company's structure and needs).

Distributed tools are indifferent to the vagaries of your company's server infrastructure as they replicate metadata to many locations. If you use a centralized system and the server catches fire you'd better hope that your backup media are reliable and that your last backup was recent and actually worked. With a distributed tool you will have many backups available.

Plastic SCM distributed system enables development teams located across different geographic locations and time zones to have immediate and transparent access to all data, it provides various working options for diverse organizations and situations:

- Traditional distributed method: Each developer has a server on his local machine, which is a replica of the central server.
- Multisite distributed method: In this case each site has a replica of the server, and the team will be working against their replica. New changes will be synchronized following regularly-scheduled replication processes or by means of real-time data replication if the servers are connected.
- Off-line work.

Usage Examples

As mentioned above, Plastic SCM distributed system can be used for different purposes and for all of them set up is as easy as installing the server, pointing it to the central server and issuing a single command to replicate the required branch.

The first usage option is the **basic distributed working method**:

This method implies that each user will have a replica of the main server on his local machine; with this kind of system, if your network connection goes down while you are working you may not even notice, the only thing you will not be able to do is connect to repositories on other computers, something that is relatively rare compared with local operations.

Any user may modify any file on any branch at any time, without restriction; in this case replicating back the information into main server in order to produce stable versions of the product requires a little more time, as there are many different servers, but as the servers would be connected synchronization can be done as often as needed, not every contributor's changes are to be replicated at the same time.

As Plastic SCM distributed system allows replication of a single branch there are many variations to this example: many developing companies work on different projects which share a common code base with variations for each of their clients. In this case their work can be separated into branches, one of them for each project, and a very effective working method would be that developers working on a project only replicate the required branch for their work and synchronization will be done among the team devoted to each project.

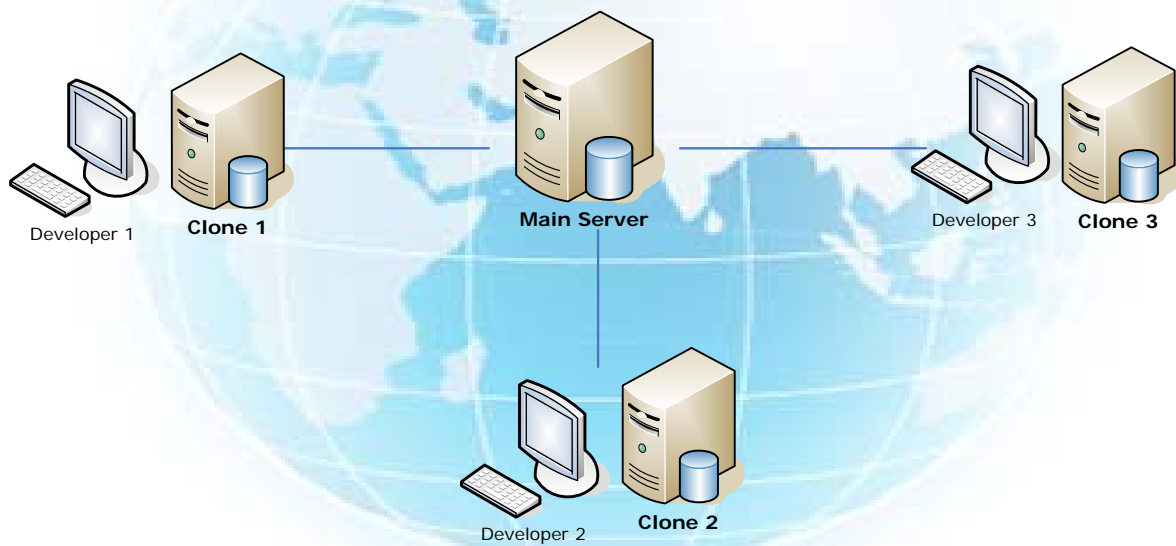


Figure 3: Basic distributed work

The second option, shown on Figure 4 is **multisite distributed work** which does not provide a separate server for each user but for each location: it enables users to selectively synchronize servers at multiple sites, a software project in San Diego, for instance, can easily be replicated in Bangkok. Plastic SCM provides a full and up-to-date copy of the required data at each site for immediate local access, which makes remote teams integration into the development process as transparent as a team which is located in the same building.

Users on each of these sites will be working against their server and every now and then, depending on the company's requirements the replication process will be synchronized taking the newly added revisions from one server into the other one.

With this situation the company can choose to allow the replicated server users different permissions: they may be allowed only to read but not write on /main branch so the synchronization process is simplified and the main server could have a /Bangkok branch on which to include these server revisions, then, whenever a new stable version of the project including changes done by both offices is created, the Bangkok server replicates /main new changes to have that new version. Plastic SCM distributed system imposes no constraints so these policies would just depend on each organization.

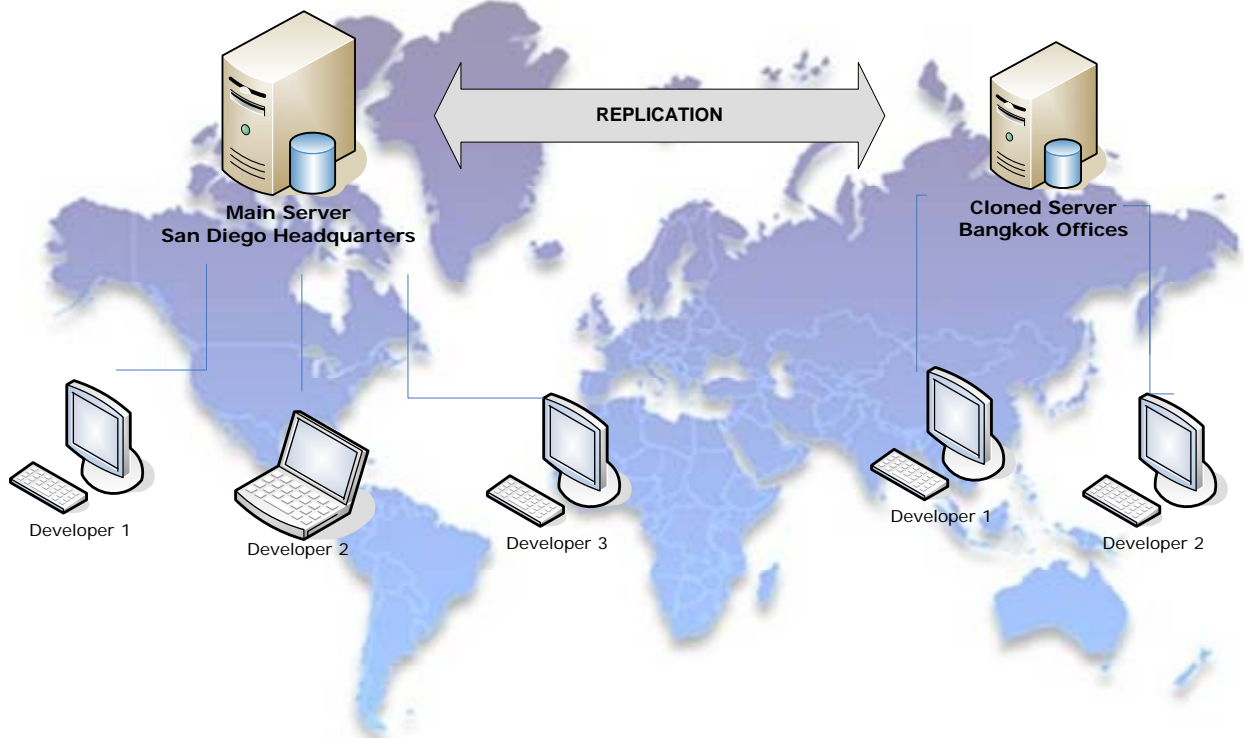


Figure 4: Multisite distributed work

The third option is **working off-line**: have to go to a conference and need to take your job with you? Just replicate on your computer the branches you need and keep on working as if you were in the office, check in your changes, create new branches...anything you need. Whenever you return to your office you only have to replicate your work into the central server.

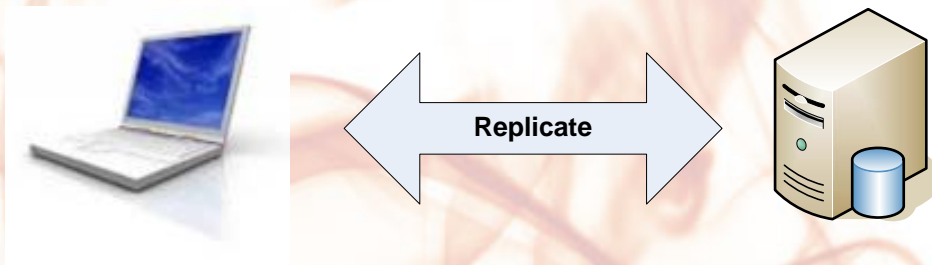


Figure 5: Off-line work

Plastic SCM distributed system offers a unique solution answering to the challenges of globally distributed development, providing different possibilities to answer to your specific needs!