GREENTEA TECHNOLOGIES

The Next Generation P2P Distributed Network Computing Platform

GreenTea Platform Whitepaper

GreenTea Platform Whitepaper

1260 Vicente Dr. • Suite G
Phone 408.307.8802 • Fax 650.961.5664

Sunnyvale, CA 94086 USA

Table of Contents

1 GreenTea's Vision and Objective

The GreenTea Vision is that each individual machine on the network should be able to help one another, and maximally share each other's resources without compromising securities. Through a software platform like GT, the whole is greater than the sum. The network and the computing resources of billions of networked devices are harnessed and maximally utilized. GT platform enables each individual PC to become a super-computer in a networked environment. This is achieved by running the GT system software on the machine (PC, MAC, Unix, etc) as a daemon service. Each GT machine on the network will offer as well as utilize the computing and networking resources. Collectively, GT realizes the network, or the Internet, as a computing platform.

Built on this vision and platform, GT opens the door to an unprecedented explosion of applications and services. The following is just a list of applications where GT can benefit:

- CPU resource aggregation
- Distributed storage
- File searching/sharing
- P2P/PAN communication
- Dynamic service enabling applications
- Resource balancing
- Application and service sharing

In general, GT can be used in any applications that exhibit the characteristics of Resource Aggregation and/or Resource Constraints.

For example, surplus/idle computing power can be collected and sold to other organizations and/or individuals that need super-computing power like the electricity power. In the meantime, individuals and/or organizations can trade their extra computing or network resources to the GT service provider (e.g. idle computing and

network resources during the night time), simply by pointing their web browser to a certain GT service provider web site, or registering their IP addresses/machine names to the GT service provider web site. The immediate impact of GT platform is that organizations that need super-computing power such as search engine companies can benefit most from using the GT technology. All their networked machines in the company can participate in providing search-engine services to the customers during the machines' idle times. As GT becomes more and more popular and mature, GT Applications will become pervasive enough to make GT service a standard service in all OS packages (Windows, Unix, Mac, all kinds of real-time OS, etc). Applications such as the computer in the car can have super-computing power, which opens the door to applications that have never been imagined before. The immediate applications will be those in controlled environment for mission critical applications, such as Intranet environment.

The GreenTea objectives are:

- To maximize utilization of the network bandwidth and idle CPU resources, based on fast network and high CPU performance.
- The GT platform is able to detect that if it is more costly to send GT subtasks to other GTHelpers to execute, then it will execute the subtask locally. Otherwise, it should be done cooperatively on the network. This is a dynamic process.
- GT platform should be transparent to the underlying machines, using the fastest technology and Java to implement platform transparency.
- Parallel-Tasking. Some tasks could be resident to remote helper computers. The tasks are executed cooperatively in a network environment.

The end-result is that, overall, the individual PC appears to have better system throughput performance than non-GT environments. The idle networked resources are fully utilized. This is not limited to computing resource. It can be networked memory resources, hard disk resources, etc.

2 GreenTea Platform

The GreenTea (GT) Distributed Network Computing Platform is a Peer-to-Peer (P2P) based generic runtime operating system platform that facilitates the parallel computing service by harnessing the existing idle computing resources on the network. GT is written entirely in the Java(TM) programming language. It runs on any Java-enabled computing platforms, such as PCs, Unix, Linux, Macintosh, etc. GT enables organizations to fully utilize their existing heterogeneous computing resources to meet their supercomputing needs without buying expensive supercomputers and super servers. This means that any computers in the organization can contribute to the supercomputing by using GT.

GreenTea platform stack is depicted in Figure 1. GT runs on top of Java platform to achieve platform transparency. On top of GT, various applications can be developed to achieve P2P computing. GT has built-in support for CPU aggregation applications.

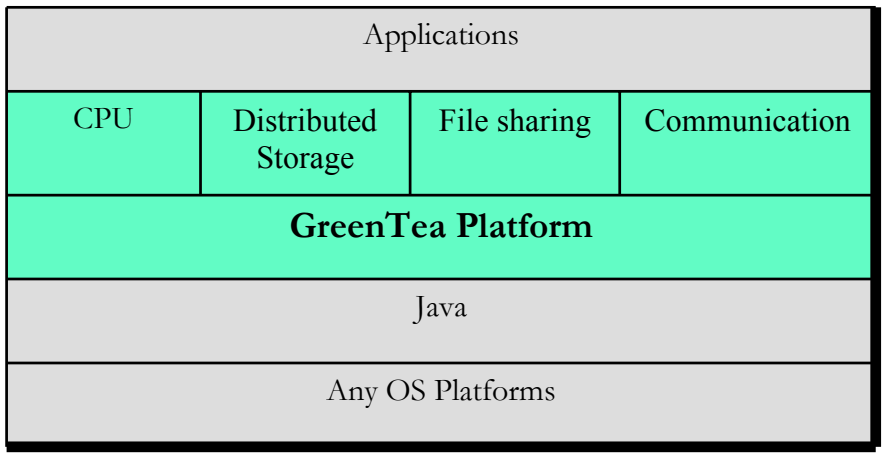

Figure 1 GreenTea platform stack

3 GreenTea Network

Each computing device running GT platform forms a GreenTea network. Each machine is a peer to any other machine in the network. Each machine can have

different views of the GreenTea network. An individual machine's view of the GreenTea network is called a GreenTea Federation. The members of the GT Federation help the peer to perform tasks. The collective result makes any individual machine in the GreenTea network appear to be a virtual super-computer.

4 GreenTea Benefits

The GreenTea platform is designed to achieve the following advantages:

- Cross platform. In order to maximally utilize different computing devices on the network, cross-platform is a very important requirement for P2P network. GT is written entirely in Java to achieve platform transparency.

- N-way resource sharing. The sharing is bi-directional. Any GreenTea node in the network can contribute as well as benefit from the network.

- Total resource sharing. The shared resources not only include hard resources such as CPU, RAM, hard disk, bandwidth, but also software resources, such as programming code, files, and services.

- Heterogeneous network. GT is able to run in a network of heterogeneous computing devices that support Java.

- True peer to peer. GT applications can be started from any desktops/peers in the GT network. The collective effect is that each desktop is a virtual super-computer.

- Serverless network. There is no central server in the GreenTea network. Serverless network provides robustness and redundancy.

- Making P2P computing easy by simply using a web browser to join the GreenTea network.

- Simple API for application development.

- Dynamic code transmission enables dynamic services.

- Fine-grained security measures in place.

GT system is the freest form (Multiple Program Multiple Data) of distributed parallel system. It allows both data and task parallel. It is very simple, light-weight, generic, and platform-transparent. GT system provides a very simple and minimum set of APIs to enable distributed parallel computing, and yet, provides a complete set of APIs to allow intuitive GUI system management. It provides maximum flexibility of a distributed parallel system.

- GT provides a system infrastructure that allows web browsers to participate in P2P computing.
- GT provides the most flexible runtime framework such that users can easily override most existing functionalities, such as task scheduling.
- GT provides a management software that can be run as either a CLI interface or a GUI application, to facilitate management functionalities such as add/delete a helper, list all tasks and helpers, system resource status, etc.
- GT provides support for inter-peer (or inter-task) communication functionalities. This is part of feature that exhibits the totally liberal side of GT systems.

5 GreenTea Platform Features

The GT 1.1 offers the following features:

- Priority Queue scheduling - the subtask is scheduled to the fastest helper first. A fastest helper is determined by the priority that was computed based on a number of parameters known to the GreenTea node.
- Eager scheduling - all helpers will be given subtask to help even though a helper is already helping with other subtasks.
- Mechanisms to facilitate communication between any subtasks.
- Allows GTApplications to send subtasks to any machine.
- Work-stealing algorithm for better system throughput and runtime performance. The work scheduler attempts to steal work when there are no subtasks left in the subtask queue, and there are available idle helpers.

- Multi-Queue scheduling - Compute-bound Queue and Network-bound Queue scheduling. It allows the GT application to specify what type of subtask it is - compute or network bound. This helps GTServer' work scheduler to determine how to schedule this subtask.
- Adaptive parallelism – Tasks can be executed in different machines in parallel. That one helper is down will not jeopardize the integrity of GT network. GTServer will steal the work, or reschedule it when it finds out the remote helper is down.
- Fault tolerance - That one helper node is down will not crash the whole GT system. Each node in the GT Federation will have its own view of the world. The GTServer will remove the helper node from its internal helper queue when it finds out the remote helper is down.
- Multi-user GT shell. Multiple users can be logged in to the GT System from local or remote GT node to administer any node in the GT network.
- A number of commands in GT shell are implemented to administer any node in the GT network.
- Join-via-web: a computer can join the GT Network by pointing the Java-enabled web browser to any of the computers in the network that has installed GT. Then the computer instantaneously becomes part of the GT network, sharing its resources.
- GT can be run as a standalone console application, or a GUI application, or a daemon, or an applet.
- Remote Login feature: Multiple instances of GT can be run in one computer. The first instance will start the GT Services. The rest of the GT instances can be used to remote login to any other computers in the GT Network for remote monitoring and management purposes.
- Granular security measures. Currently a Java security policy file is used to govern what resources a mobile code can utilize on the local system. For example, the security policy file can specify read and/or write permission of a specific directory for a certain piece of mobile code.
- GUI interface is used to configure GT system.

6 GreenTea Programming Framework And API

GT platform provides a very simple programming framework that allows programmers to write GT-enabled P2P application programs. Using GT API, programmers do not need to be concerned with how many machines on the GT network, and how to communicate with other machines. GT API provides the basic resource scheduling and communication functionalities. From the programmer's point of view, the programmer is simply programming to a virtual supercomputer. He does not need to be concerned with how to communicate with other peers, how the tasks are scheduled to what resources. All he needs to do is to send tasks to GT system, and get results from the system. This design releases the programmer to concentrate on domain-specific programming.

6.1 A basic GT Application model

A GTApp is a coarse-grain parallel Java application with a basic parallel formula like this:

```
main()
{
        register(GTServerID);
        for (n subtasks)
                send(GTTasks)
        for (n subtasks)
                receive(Results, GTAppID, GTTaskID);
        deregister(GTServerID)
}
```

GTApps submit GTSubTasks to GTServer that manages and schedules the GTSubTasks to the GTHelpers in the GT Federation to which the GTServer belongs.

GTApp can also be a message relay point acting as a message-forwarder or proxy among GTSubTasks if the tasks need to synchronize with each other, or if they have data dependency. GTSubTasks can also directly communicate with each other using GTSubtask API.

GTApp creates a number of GTSubTasks. GTSubTask is an abstract class or interface that has a "run" method. GTSubTasks encapsulates the lifecycle primitives that can be managed by the GTHelper host machine. GTSubTask is analogous to an applet but less restrictive governed by user-defined security policy.

Another critical difference between GTSubTasks and applets is that GTSubtasks are dynamically pushed to remote machines for execution. This is the foundation to the dynamic service enabling feature of GT platform.

GTApp is not restricted to only be a parallel Java application. The GTApp can do anything else that a Java application can do, e.g. a GTApp can be a server application that talks to a web applet or other web pages to allow the user to issue complex commands or queries for the GTApp to execute and return results to web page or applets. GTApp can also issue subtasks to specific machines, to dynamically enable a machine to perform certain tasks.

6.2 Register and Deregister to GT platform

A GTApplication must first registers to the GTServer to get the pid for the GT application. This method is called in the constructor of GTApplication to save developers from calling this method explicitly.

A GTApplication deregisters from the GT platform in order for GT to clean up data structures related to GTApplication.

6.3 GTSubtasks and their Communications

GTSubTasks are of the type `GT_COMPUTE_BOUND` and `GT_NETWORK_BOUND`. GTSubtasks has a stealable property that can be set be the user. GTSubtask API provides inter-subtask communication mechanisms so that any subtask running on any machine can communicate with any other subtasks in GT network.

6.4 Send subtasks and get results

GTApp calls `putSubTask` () method to send subtasks to GTServer for scheduling. The GTServer schedules the subtasks to the fastest machine in the GT network based on a set of proprietary algorithms. A GTSubtask can be put to the GTServer in two ways: 1. GTSubtask lets GTServer to schedule to which machine the subtask is executed, 2. GTSubtask specifies which machine is to execute this subtask. The 1st approach leaves the scheduling of the subtask to the GTServer, which will automatically transmits the subtasks to remote machine in an optimal fashion. The 2nd approach gives programmers more control as to where the subtask will be executing. This is useful in certain P2P applications.

GTApp calls getResult() method to get results back from GTServer, provided the subpid of a GTSubTask.

6.5 Interface to native code

GTApplications are not limited to Java applications. Java Native Interface API can be used to interface native code on a specific machine architecture. Local programs can be invoked through GTSubtasks. Although it could compromise GT's platform transparency, it provides freedom to the user and the available applications. Developers can choose to trade platform transparency for making use of available native applications.

7 GreenTea Applications

Areas of applications for GT are:

1. distributed network parallel super-computing
2. distributed storage
3. bandwidth sharing (distributed network tasks): optimal network bandwidth usage
4. bi-directional: individual empowerment, distributed access to resources
5. de-centralized system
6. resource constraints applications
7. resource aggregation applications
8. distributed computing,
9. file-sharing/searching
10. collaboration

GT can be applied in many other areas of applications such as bioinformatics, Electronic Design and Simulation.

7.1 Distributed Storage

GT can be used in distributed storage applications. From any originating machine, a subtask is transmitted to other nodes on the GT network, along with the data to be stored in remote machines. The GT application keeps track of which piece of the file is stored on which GT node. This in effect is making use of remote unused hard disk resources. It saves organizations from purchasing expensive data center machines.

7.2 Distributed Bandwidth I/O Sharing

GT can also be used in Distributed Bandwidth I/O sharing applications. From any originating machine, a subtask is transmitted to other nodes on the GT network that have broadband network access. The originating machine can then effectively make use of the under-utilized broadband network of the remote machine. This in effect is making use of remote unused bandwidth resources.

7.3 File Sharing/Searching

GT can also be used in File Sharing/Searching applications. From any originating machine, a subtask is transmitted to other nodes on the GT network to search for the file on the local hard disk, and then propagate the search request to other peers on the network.

7.4 File Sharing/Searching

GT can also be used in content and code delivery applications. From any originating machine, GT can send a subtask that combines the mobile code and mobile data along with it, to a remote machine for execution. This dynamically enables the remote machine to do the things it was not able to do before, at runtime.

8 GreenTea Network Management

The GT System can be managed through a standalone GT System Management software, from any node in the GT network, or from any Java-enabled web browser. The administrator can login to any specific GT machine to perform admin functionalities.

The owner of the computer will be able to monitor activities and status of the GT environment for his machine. The user can see the status of his GT subtasks, i.e. where they are executing, how long they have been resident, are they running or prematurely stopped, how many GT subtasks that this machine is helping and where they come from, etc. The owner of the PC can terminate a specific running GTSubtask at anytime as he wishes. The management feature also controls how much of the machine's resources be used for GT services. The user has complete control over who to help,

when to help for parallel execution of remote code. The GT network admin or individual users are able to monitor activities and status of the GT environment. All user admin and management features can be managed from a web management UI interface.

9 Summary

GreenTea (GT) is a P2P software infrastructure that harnesses the idle computing resources such as CPU, RAM, and hard disk on the network. It is a generic runtime platform on which numerous P2P and server applications can be developed and executed. GreenTea platform addresses P2P computing issues in a novel approach. GreenTea achieves the following:

- True edge to edge resource sharing (no central server)
- Fully cross platform
- N-way resource sharing
- Small footprint – extendable to PDAs
- Built-in redundancy
- Granular security