Many legs make light work.™

Distributed Computing with the Digipede Network™

Table of Contents:

©Copyright Digipede Technologies, LLC.

Digipede and the Digipede Network are trademarks of Digipede Technologies, LLC. Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

Executive Summary

Distributed computing has moved from academic research to commercial reality. Your organization can use its existing compute resources to improve the scalability and speed of your most demanding applications right now. The benefits of distributed computing have been unavailable to most businesses, agencies, and institutions in the past because of prohibitive cost and complexity. With the Digipede Network™, distributed computing has become affordable, quick, and easy.

The right platform is the key to distributed computing success. Most or all of your servers, computers, and software run on Microsoft Windows. Your developers use Microsoft's Visual Studio.NET software development tools. The Digipede Network, built entirely on the Microsoft .NET platform, is your answer.

As a seasoned Microsoft software developer, Digipede understand the needs of Microsoft customers. Other distributed computing solutions focus on UNIX and Linux, requiring expensive consultants, lengthy implementation, steep learning curves, and a heavy IT burden. In contrast, the Digipede Network is radically easier to buy, install, learn, and use. With its familiar Windows user interface, the Digipede Network allows you to become productive immediately. Unlike competing solutions, no complex scripting, major modification of existing applications, or on-site implementation help is necessary.

The Digipede Network delivers the benefits of distributed computing at any scale. Whether you're a small department with five computers or a corporation with thousands of servers, desktops, and cluster nodes, you can benefit. You can download, install, and configure the Digipede Network in less than an hour, and you'll be on your way to improved productivity and application performance. It's that simple.

Distributed Computing Improves Performance

IT departments must make the most of their budgets. Getting the greatest return-on-investment (ROI) from hardware and software is critical, which is why distributed computing makes sense.

Purchasing new hardware will increase your computing capacity, but it won't apply that capacity to your most demanding applications. Distributed computing software from Digipede takes advantage of all your computing capacity. The Digipede Network uses dedicated and underutilized computing resources in new and existing hardware, so your applications perform better. Increased scalability and speed improves worker productivity, and you can bring products to market faster, significantly improving ROI.

Every organization, small or large, can benefit from accelerated applications, increased revenue, and reduced costs. Industries that can benefit include life sciences, financial services, energy, entertainment, government agencies or departments, academic research, and manufacturing.

The Digipede Network can do the following:

Increase application performance at any scale. Even small installations with a few dozen nodes can see order-of-magnitude improvements in application execution. Minutes-long jobs are finished in seconds. Overnight jobs are finished over a coffee break.

Increase your team's productivity. Your team can focus its energies on key business problems, rather than on developing infrastructure.

Improve and speed up Return On Investment. Improved application performance leads directly to a better competitive position and cost savings. The Digipede Network offers a far lower per-unit cost than alternative solutions.

Support legacy applications. Intelligent agents manage execution of existing Windows applications, with no need to convert or recompile code.

Distributed Computing by Industry

While each industry has its own specialized applications, the performance challenges are similar. Leading companies in the industries listed below rely on sophisticated applications to drive business performance.

INDUSTRY	SAMPLE APPLICATION AREAS	BENEFITS
Life Sciences	• Bioinformatics • Drug discovery • Quantum-mechanical modeling • Statistical analysis	Improved throughput in genomic and proteomic analysis leads directly to faster screening of candidate compounds for new treatments. Productive research teams increase the value of the product pipeline. Cheap and plentiful computing power, dynamically applied to essential applications, gets products to market faster
Financial Services	• Trading strategy simulation • Transactions processing • Portfolio optimization • Pricing simulation and forecasting • Risk management	Improved modeling accuracy leads directly to improved competitive position. The most accurately priced assets enjoy a trading advantage. Those who can evaluate portfolio risks accurately can hedge those risks more cost-effectively. Those who process more transactions each second can scale their businesses more easily. Cheap and plentiful computing power, dynamically applied to essential applications, provides an important edge.

Energy	• 3D modeling and simulation tools • Scheduling, billing and settlement systems • Electric grid simulation • Risk management	Improved simulations of reservoir operations can increase the total resources recovered. More robust simulations of grid operations can improve system reliability. Increased throughput in billing and settlements can reduce the float on billions of dollars worth of electricity and gas. In these and other cases, distributed computing provides immediate benefits that far outweigh the costs.
Entertainment	• Animation, special effects, and video rendering • Asset management and data compression • 3D modeling	Quicker 3D rendering means increased productivity and throughput on film and video projects. The ability to coordinate computing resources – dedicated and shared – supports more and bigger jobs. Cheap and plentiful computing power, dynamically applied to essential applications, leads to faster project completion and more projects.
Academic Research	• Simulation modeling • CAD • Statistical and numerical analysis	Increased application performance opens up new areas of research. Greater scalability results in more accurate modeling and analysis. Distributed computing supports a wide range of applications throughout academia using existing resources – even shared desktops already in place in department offices and labs.
Manufacturing	• Aerospace simulations • Auto and aerospace crash test simulations • Computational Fluid Dynamics	More precise simulations results in improved products at lower cost. In many cases, improved simulations can reduce the need for costly prototyping and destructive testing. Faster simulations and modeling reduces the time lag between the design and manufacturing of new products. A faster time to market enables bigger market share and profits.

Table 1: Distributed Computing Applications in Industry

Inside the Digipede Network

The Digipede Network uses the power of your Microsoft Windows desktop computers, servers, and clusters to improve the scalability and speed of the most compute-intensive, transaction-intensive, and data-intensive applications. This powerful software parcels out complex computing jobs across the network, dynamically allocating the computing power of both dedicated and idle resources.

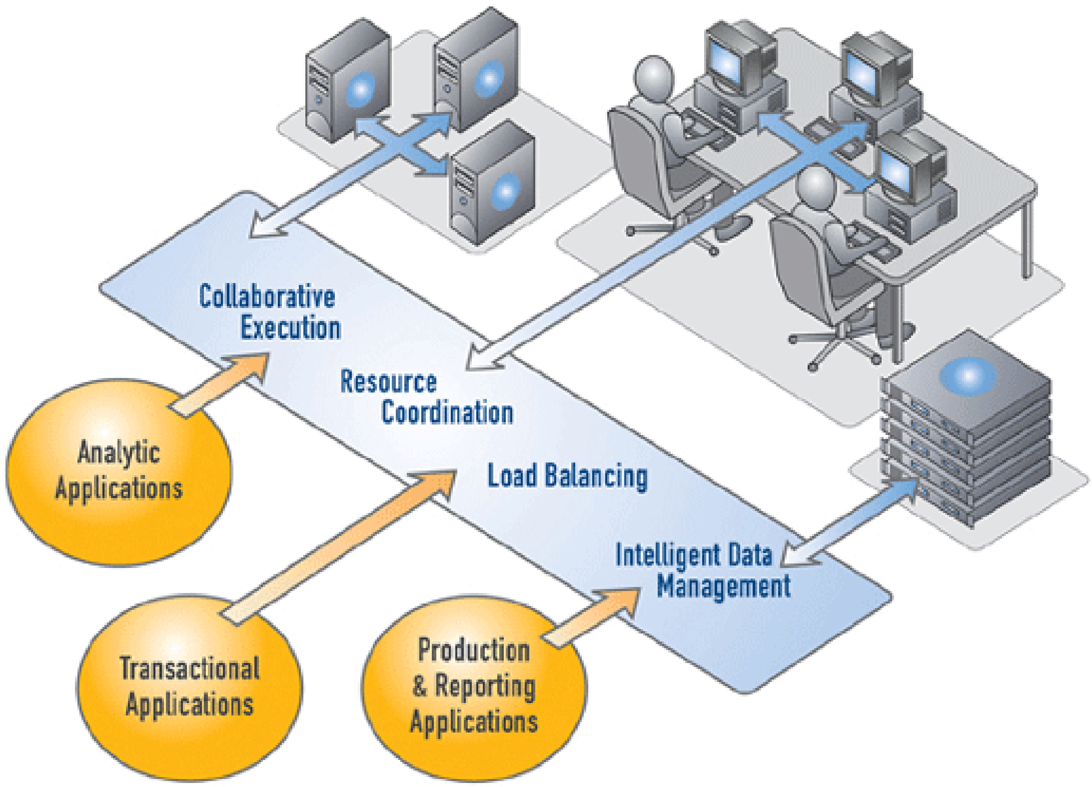

Figure 1: The Digipede Network

The Digipede Network has several components:

- **Digipede Agents** manage each of the individual desktops, servers, or cluster nodes, and the tasks that run on them;
- **Digipede Server** manages the workflow through the system;
- **Digipede Workbench** lets users define and run jobs; and
- **Digipede Control** provides a browser-based tool for system administration.

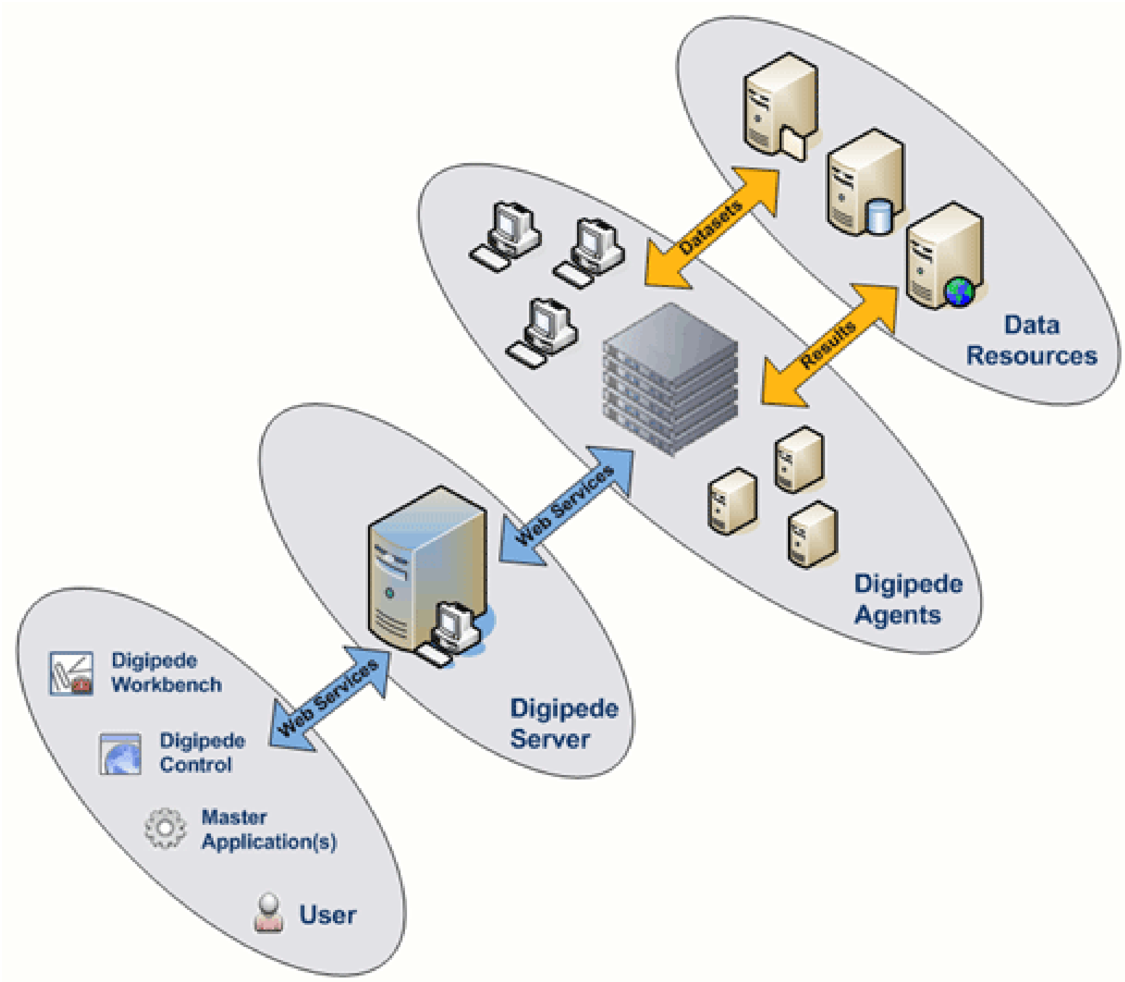

Figure 2: How the Digipede Network Works

Digipede Agent

Each Digipede Agent manages the computer on which it runs. The Digipede Agent uses Windows Management Instrumentation to gather information about the computer (hardware, software, and data resources available locally), including:

- Type and speed of processor
- Amount of RAM
- Local disk space available for distributed applications and associated data
- Network throughput availability
- Installed applications
- Locally resident data
- Network proximity to required data

After analyzing this information, the Digipede Agent checks the Digipede Server to assess the tasks available for execution. Based on the requirements and priority of these tasks, and the abilities of its compute resource, the Digipede Agent chooses the most appropriate task. For many real-world business applications, this approach is a significant improvement over "command and control" scheduling systems.

For resources that are not dedicated to distributed computing (for example, a desktop computer used for business during the day, or a shared departmental server), the Digipede Agent monitors the resource's availability. The Digipede Agent can be configured to perform work in any of these ways:

- Constantly, in the background, at low priority;
- Only when the screen saver is active; or
- Only during scheduled hours, such as nights and weekends.

The Digipede Agent is unobtrusive to a shared resource user. If there is any degradation in performance, the user can click configure the Digipede Network to run only when the user is not present.

Tasks and Jobs

In the simplest case, a "task" is a single run of a set of instructions. A "job" is a collection of tasks that run that same set of instructions with different input values. Examples of tasks include:

- A single simulation model run
- A single Monte Carlo trial
- A single search of a genome
- The production of a single invoice or report.

A Digipede Agent executes tasks with a command-line executable. It can take different inputs (from a file, as command-line parameters, or a combination of both). The Digipede Agent calls the distributed application with the appropriate input parameters, and reports back to the Digipede Server via the Agent Management Web Service when that task is completed.

For finer control, the agent can also manage applications developed with our API (the Digipede Framework), which also provides progress reporting and workflow control.

Digipede Server

The Digipede Server manages the flow of work to and from the Digipede Agents. The Digipede Server keeps a prioritized work queue of tasks for one or more jobs. It publishes this work queue periodically in the form of a job table, and agents check this job table to see the jobs and tasks that are available.

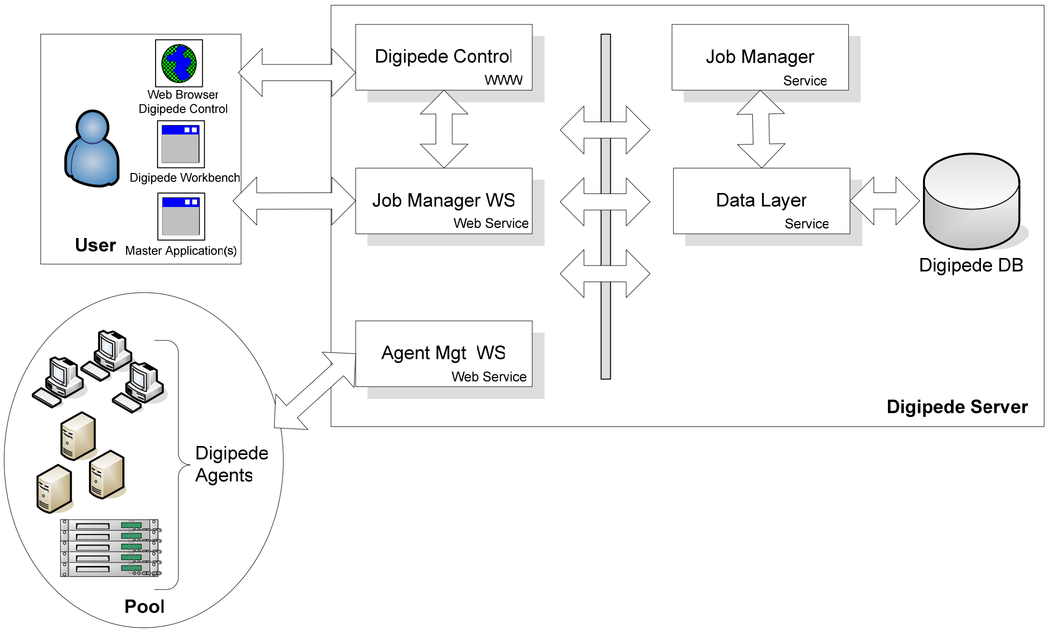

Figure 3: Server Components

Server Components

The Digipede Server has several components:

- **Job Manager** controls the priorities of jobs and tasks, and tracks task allocation to agents.
- **Digipede DB** stores the history of jobs and tasks taken by agents.
- **Digipede Control** is the Web site for administering the Digipede Network: managing user accounts and privileges. Users can also submit jobs through Digipede Control.
- **Agent Management Web Service** manages all communication between the Digipede Server and the Digipede Agents.
- **Job Manager Web Service** manages all communication between the Digipede Server and user applications, including Digipede Workbench, Digipede Control, and any master applications written by the user or by third parties. (Master applications can start jobs on the Digipede Network.)
- **Data Layer** moderates traffic between the Digipede DB and all other Digipede Server components. This layer provides for two-way caching, so that the Job Manager, Job Manager Web Service, Agent Management Web Service and Digipede Control can access recent data without hitting the database directly, and Digipede DB updates can be performed asynchronously.

These components are logically separated to improve performance and to support back-end scaling. Communication among these components is handled via Web Services. For most real-world applications, Digipede Server running on a single-processor commodity server can support thousands of Digipede Agents. For very large implementations (hundreds of thousands of Digipede Agents), individual Digipede Server components can run on separate machines, and each component can run on multiple machines simultaneously.

Digipede Control

Digipede Control is a browser-based user interface for administering the Digipede Network. Digipede Control gives you a complete view of the distributed-computing operation. It displays system status, including each individual compute resource, all currently running jobs, and all completed tasks. You can use Digipede Control to:

- **Administer user privileges**. Manage users for the Digipede Network. Create user names and passwords and assign roles, so users can log in and submit jobs.
- **Administer computers**. Add specific computers to pools of resources, to which jobs are submitted. You can choose all the compute resources in your company, or define a pool limited to a set of machines in one server room or department. Low-priority jobs can be limited to a resource pool available only at night, for example, while dedicated clusters are reserved for higher priority jobs.
- **Set schedules for jobs**. Determine when jobs run on the compute resources in a given pool.
- **Abort and restart jobs**.
- **Manage bandwidth**. Limit the bandwidth for jobs so you don't overload the network, allowing other computing operations to function properly.

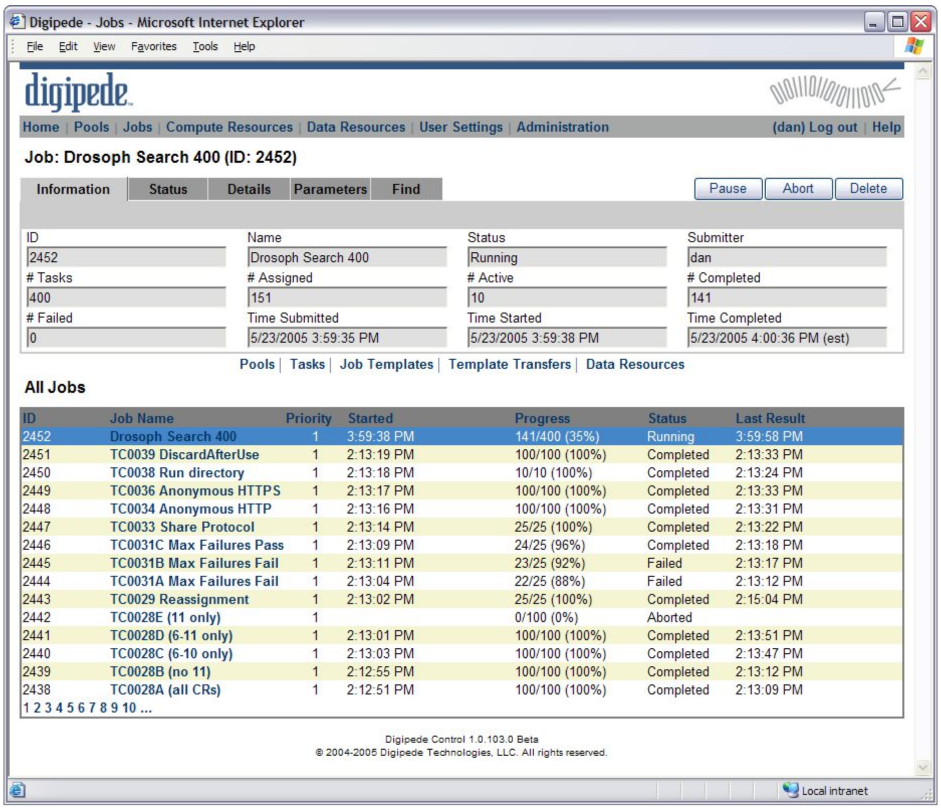

Figure 4: Digipede Control

Digipede Workbench

While Digipede Control is used primarily by administrators to manage the system, Digipede Workbench (a .NET Smart Client) is a tool for every user who wants to submit jobs to the Digipede Network. With its familiar Windows user interface, Digipede Workbench helps every user to become productive with the system immediately.

To design and run a distributed computing job, you must do the following:

- Tell each computing node what software and data are required to complete its assigned tasks and where to find that software and data.
- Move and install the required software and data on each node.
- Tell each node when to start processing the data required to complete its tasks, and what to do with any results.
- Move the results to their specified destination.

Until now, distributed computing users had to spend a large amount of time on complex scripting before accomplishing even the most basic tasks. Bewildering configuration options and inconsistent interfaces created needless complexity.

Digipede Workbench removes this complexity, accomplishing the steps described above with no scripting. Wizards guide users through the process of submitting a job. Users can go from installation to productive use in less than an hour.

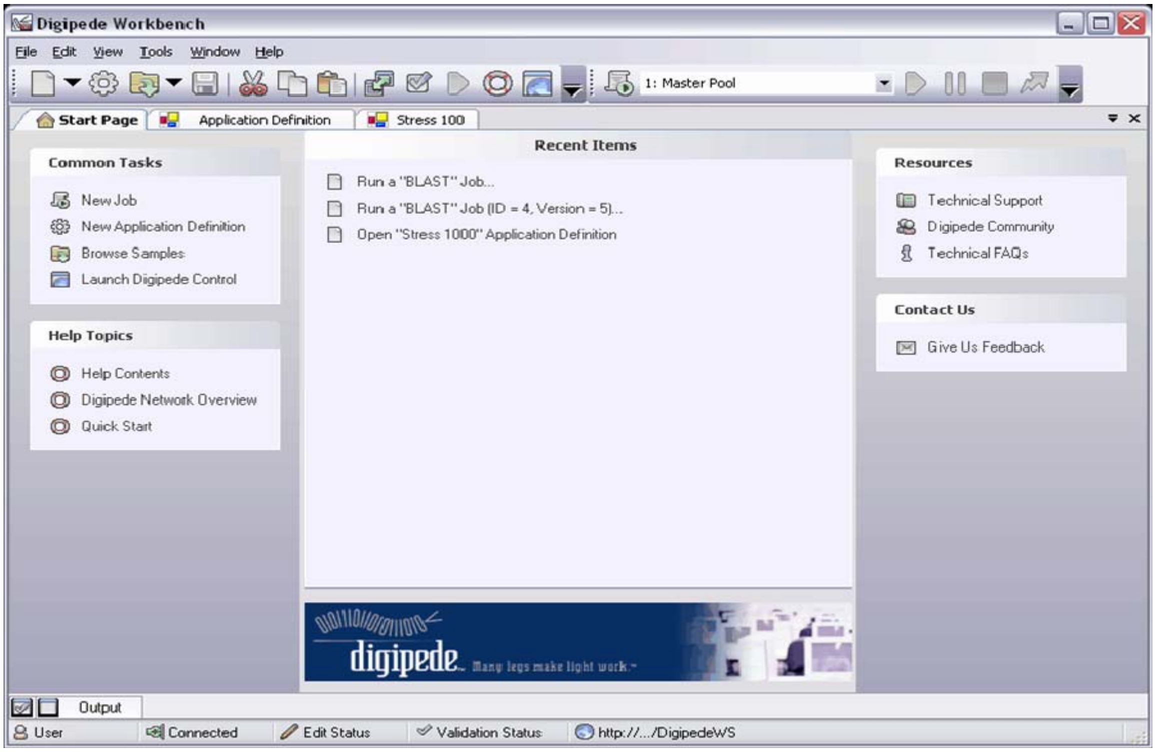

Figure 5: Digipede Workbench

Beyond Digipede Workbench: Distributing Your Application

Your developers can use the Digipede Framework API to take the fullest advantage of distributed computing for your application. They can make the adjustments in a matter of hours—a fraction of the time it would take to develop a distributed application from scratch.

Since it is based on the Microsoft .NET platform, developing applications that can start jobs on the Digipede Network (master applications) can be as easy as adding a few lines of .NET code to an Excel 2003 spreadsheet. For many users, Excel is the tool of choice for consolidating and reviewing analysis results, so it makes a natural front-end for distributed applications.

Your developers can use Visual Studio to designate objects to execute on the Digipede Network. Our software takes care of the difficult parts: finding the appropriate computers to use for execution, migrating the object to those machines, starting and managing the execution, and migrating the object back into your original code. From the developer's perspective, it is no different than any other object-oriented programming, but the objects take advantage of your whole network. Your developers don't need to apply SOAP, Web Services, or .NET Remote to distributed applications. They don't need to manage the complexities or security issues of a dynamic network environment.

Here's a brief technical summary:

- You add a reference to the Digipede Framework in your application. The Framework gives your application the ability to communicate with the server via Web Services.
- You derive a class from our Worker class to create an object that will do the heavy lifting.
- When your program runs, your application instantiates as many of the Worker objects as you need to do your work. Each Worker object represents a task in the Digipede Network.
- The objects are serialized on your machine and streamed to the Digipede Server.
- When the agents check in, they see that there are tasks waiting to be executed. The agents grab the tasks and de-serialize the objects on their local machine.
- The Digipede Network calls a method that tells your object to start executing. Your object executes simultaneously on many machines on your network.
- When it has finished its work, your object tells the agent that it is done. It is then re-serialized and returned through the Digipede Network to the calling application.

Consider an engineering simulation application containing a Simulation object. It has twenty different properties that function as inputs to your model. Currently, your programmer sets the properties and calls the PerformCalculations() method. The model takes ten seconds to run each time, and it generates over a hundred parameters as results. A typical simulation requires 1,000 model runs and takes nearly three hours to complete. How can Digipede Framework make this faster?

By deriving the Simulation object from the Worker class, your developer can execute this object all over your network. Set up the object just like you do now, but instead of calling PerformCalculations() directly, call the DoWork() method. DoWork() wraps the PerformCalculations() method and readies it for distribution across your network. You can instantiate all 1,000 objects quickly, and the DoWork() method tells the Digipede Network to create a task in the system for each of them.

The Digipede Network manages the execution. On each machine on your network, your Simulation object is created and the PerformCalculations() method is invoked. Your model runs, and the object is passed back up to the server. While it has your process running, the agent checks to see if there is more work to do. If there is, the agent instantiates a new object without unloading the executable. This makes the most efficient use of the resource.

Back on the calling computer, the Digipede Network hands each of your Simulation objects back to your application. In each object, all 100 result parameters are set. You get the same results that you did before using the Digipede Network, but instead of taking three hours, the fifty machines in your office complete the job in less than five minutes.

Running a Job

Here's a typical scenario of how all the different components of the Digipede Network work together to run a job.

- **A user defines and runs a job using Digipede Workbench.**

  Using a wizard, the user tells the system which files to install, where they're located on the network, how to start a task, and what to do with results. During this step, the user also specifies the minimum requirements for computers that run this job (minimum RAM and processor speed and any software required).

- **Digipede Workbench uses Web Services to communicate with the Digipede Server.**

  Specifically, Workbench sends a SOAP message to the Server with details of the job to run. The Digipede Server consumes the SOAP message and authenticates the user who submitted the job.

- **The Digipede Server alerts the Digipede Agents that the new job is available.**

  The Digipede Agents communicate frequently with the Server, looking for new jobs to tackle. An administrator sets the frequency for agent check-in using Digipede Control.

  When an individual agent determines that its computer fits the requirements of a particular job, it can claim some of the tasks for that particular job. At this point, the agent checks to see if it needs to download files or install software to process the tasks.

- **The agent processes the tasks, delivers the results to a specified location, and notifies the Digipede Server that these tasks have been completed.**

  As the job runs, the Digipede Server monitors the agents. It knows if an individual agent took tasks but failed to deliver results. The agent's computer may have been turned off, for example. In that scenario, the Digipede Server reassigns the tasks to another Digipede Agent. The Digipede Server guarantees that all tasks will be completed, and reports progress back to Workbench so the user can monitor progress. Workbench can display a graph showing the tasks completed, in process, and remaining. The graph updates frequently, showing the user when all tasks have been completed.

Through this distributed computing model, jobs finish faster. For many applications, the more Digipede Agents you add, the faster your jobs will run.

Cluster and Grid Configurations: the Right Balance

For most of the applications that can benefit from distributed computing, dedicated cluster computing is too restrictive and full general grid computing is too complex.

	HomoGeneous Hardware (Dedicated)	HeteroGeneous Hardware (Dedicated or Shared)
Single Operating System	Cluster Computing	Extended Cluster: The Digipede Network
Multiple Operating Systems	N/A	Full General Grid Computing

Table 2: Clusters, Grids, and the Digipede Network

As shown in Table 2, traditional cluster computing requires homogeneous hardware. In almost all cases, such hardware is exclusively dedicated to cluster computing.[1] Expanding the computing capacity of a cluster requires new hardware and reconfiguration of the cluster, which can get expensive and complicated fast.

Full general grid computing provides tremendous additional flexibility, including the ability to run applications across multiple operating systems, and even multiple organizations. This approach assumes that applications are configured to run across multiple operating systems, which is valid for some Java applications. Some solutions espouse this multiple-OS approach. But Figure 6 illustrates how this can lead to implementation nightmares, requiring custom configuration and on-site help from expensive consultants.

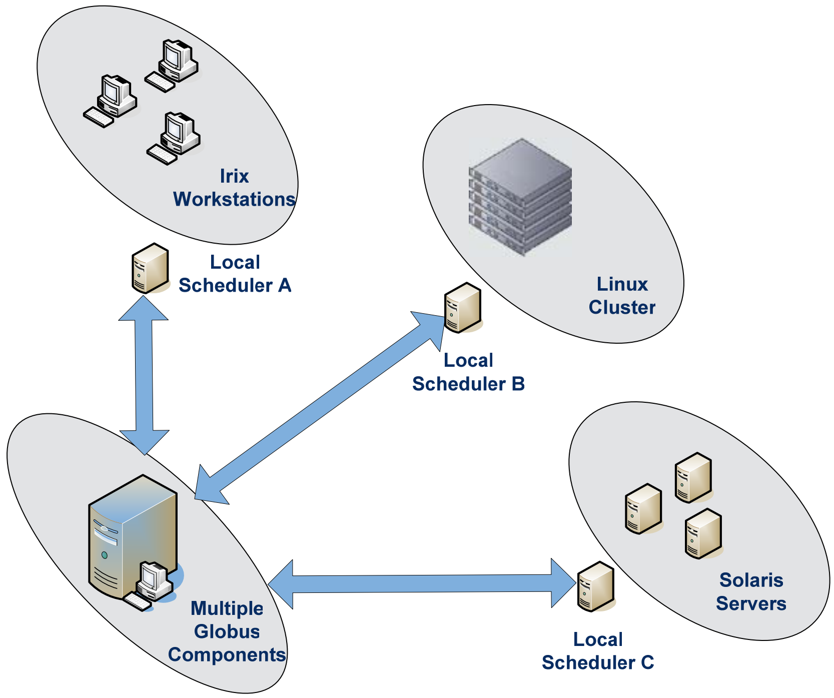

Figure 6: Multi-OS Grid Computing

The Digipede Difference

Most applications (and most computing resources) use a single operating system—Microsoft Windows. Your organization can deploy the Digipede Network in several Windows-based cluster and grid configurations.

A cluster is a great tool, but it can be made far more powerful and cost-effective by extending it to include additional shared desktops and underutilized servers throughout the enterprise. A grid is a great idea, but most organizations don't need the complexity of integrating multiple operating systems to achieve grid benefits. The Digipede Network takes full advantage of both dedicated and shared Windows resources without introducing the needless complexity of multi-OS systems.

[1] An exception is the practice known as "Beowulf at Night." In a nutshell, Beowulf at Night requires each machine to have two operating systems: Windows, which is run during the day, and a selected version of Linux, which is run at night. In theory, every person in an office sets up his or her machine as a dual-boot system, and remembers to reboot the machine in Linux before he or she goes home, so that the machine can join a Beowulf cluster for the night. In practice, our experience indicates that this approach is too complex and unreliable for most businesses.

Dedicated Cluster Configuration

When the Digipede Network is configured to run jobs on a dedicated cluster, it is more flexible than traditional "command and control" scheduling systems, and it adds value to existing cluster management tools.

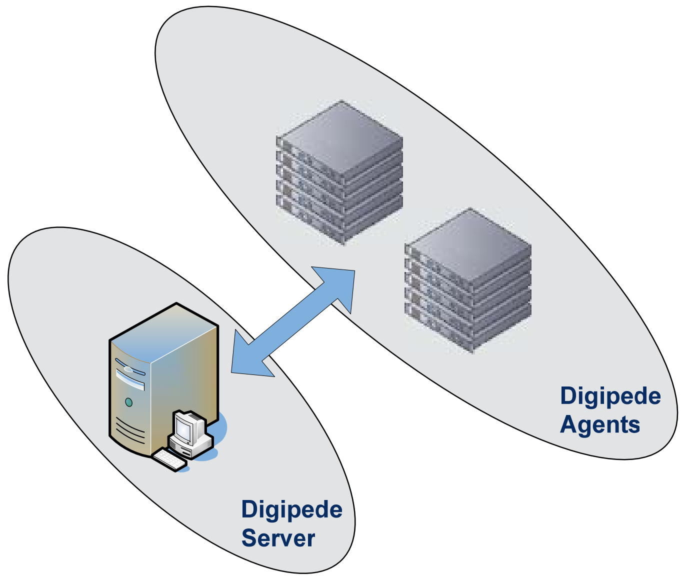

Figure 7: The Digipede Network in Dedicated Cluster Configuration

Desktop Grid Configuration

When the Digipede Network is deployed in a "departmental desktop grid" configuration, with a single Digipede Server and multiple Digipede Agents running on shared resources throughout a department, the department uses available resources more efficiently. Agents run unobtrusively, so desktop users are never inconvenienced by distributed computing jobs.

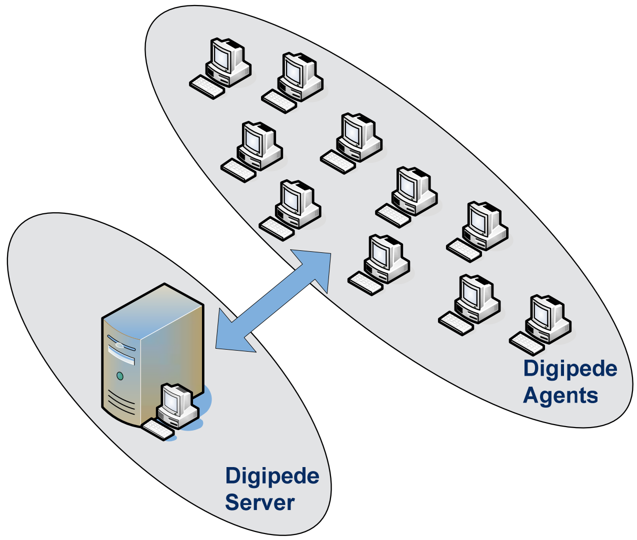

Figure 8: The Digipede Network in Desktop Grid Configuration

Extended Cluster Configuration

When the Digipede Network is deployed in an "extended cluster" configuration, it adds even greater value.

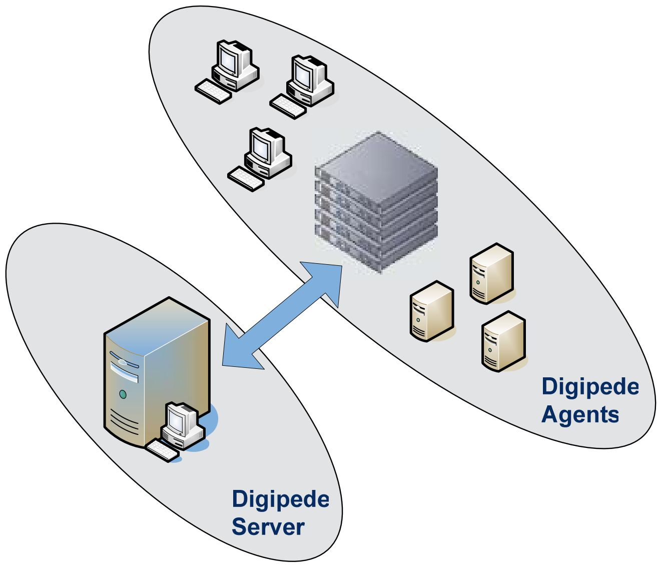

Figure 9: The Digipede Network in Extended Cluster Configuration

A cluster of Windows servers (16-32 nodes), supplemented by hundreds of shared desktops throughout the organization, can be as powerful as a far larger cluster by deploying the Digipede Network. Many real-world business applications require guaranteed execution of jobs within a specified period. A combination of dedicated and shared resources is often the most cost-effective way to meet this requirement.

Economic Benefits

The Digipede Network changes the economics of cluster deployments. Given a choice between purchasing a 100-node Linux cluster and a 100-node Windows cluster, good arguments can be made for both sides. In most cases, the cost and performance differences between them is relatively small, but bring the Digipede Network into the picture and Windows gains the advantage.

Compare a 100-node Linux cluster to an "extended cluster" configuration of the Digipede Network. This configuration might include a dedicated 32-node Windows cluster augmented by 150 shared Windows desktop computers and servers already owned by the organization, with Digipede Agents running on all 182 nodes. The 182-node extended cluster has higher throughput than a 100-node dedicated cluster, whether Linux or Windows. The cost of 32 new nodes, combined with a 182-processor license for the Digipede Network, is far less than the cost of a 100-node dedicated cluster running Linux or Windows.

The Easiest Path to Scaling Your Application

Whether your organization is conducting biotech research, creating new aerospace designs or processing millions of financial transactions a day, you need access to cheap and plentiful compute power, dynamically applied to your most important applications. Distributed computing is the answer. The Digipede Network, built on the Microsoft.NET platform, dramatically reduces the cost and complexity of distributed computing.

Digipede's powerful software allows companies, institutions, and agencies of all sizes to improve application performance, so workers can innovate and bring products to market faster without expensive investments in hardware and consulting services. You probably own all the computing power you need to accelerate your most important applications. With the Digipede Network, you can put that power to work without consultants, deep IT knowledge, or expensive implementation. Visit www.digipede.net to get started with the Digipede Network today.