The AppLogic™ Grid Operating System

April 2006

www.3tera.com

Copyright © 2006 3TERA, Inc. All rights reserved.

Introduction

Web 2.0 and Software as a Service (SaaS) are rapidly reshaping the software industry. For the first time, software applications can be delivered directly to any user, anywhere in the world, as online services accessible at a click of the mouse. But there is no free lunch: today, it takes enormous amounts of money and skills to build and operate the IT infrastructure needed to deliver online services to hundreds of thousands of users.

Getting the cost and complexity of scalable web operations under control requires an entirely new approach to infrastructure. Hardware provisioning and equipment failures need to be dealt with automatically. The infrastructure has to scale dynamically without requiring application downtime. Most importantly, deploying new applications and changing existing ones must be made much simpler.

The basis for this new infrastructure is the grid. Despite the buzz about grid computing, grids have so far been limited to running so-called high-performance computing (HPC) applications such as business intelligence, simulations, derivatives trading, etc. However, the vast majority of web and business applications are not computational in nature – instead, they process large numbers of small concurrent transactions (transactional applications) and/or deliver content (I/O intensive applications).

AppLogic is a grid operating system for scalable web applications and services. AppLogic runs distributed transactional and streaming applications on grids of commodity hardware. It does not require a SAN or other expensive shared storage. What's more, AppLogic is completely compatible with existing web applications, and is open and vendor-neutral.

Copyright © 2006 3TERA, Inc. All rights reserved.

Overview

AppLogic is the first grid operating system that is designed for web applications and is optimized for transactional and I/O intensive workloads. It uses advanced virtualization technologies to ensure complete compatibility with existing operating systems, middleware and applications. As a result, AppLogic makes it easy to move existing web applications onto a grid without modifications.

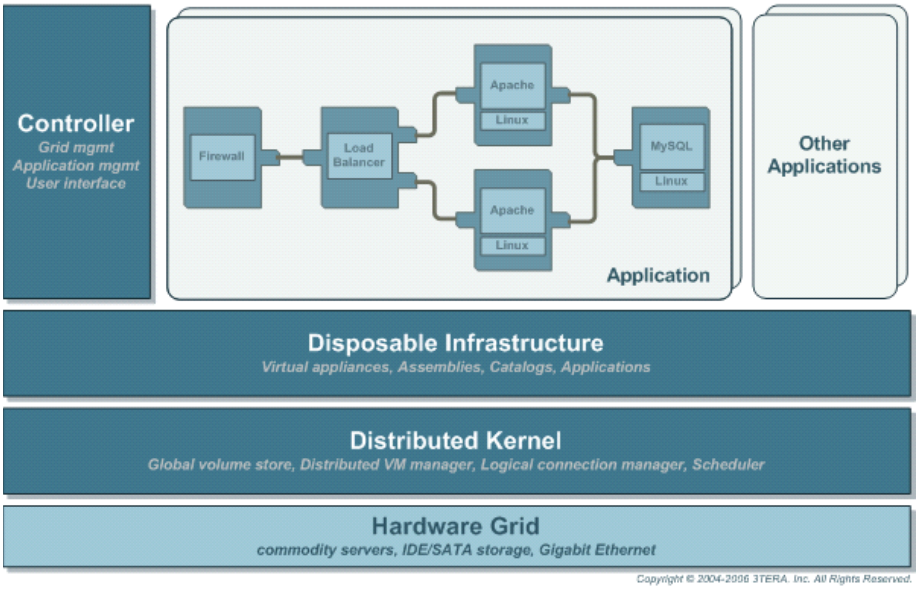


Figure 1. The AppLogic grid operating system

Figure 1 illustrates the architecture of AppLogic. The system runs on grids assembled from commodity servers connected via Gigabit Ethernet. Some (or all) of the servers within a grid are expected to have directly attached storage - inexpensive IDE/ATA/SATA hard drives which AppLogic uses to provide a distributed storage pool for applications.

AppLogic includes three major subsystems:

- **Distributed Kernel** that abstracts and virtualizes the grid hardware, and provides core services such as processing, interconnect and storage;
- **Disposable Infrastructure Manager** that handles the infrastructure needs of each AppLogic application; and
- **Grid Controller** that serves as a central point for managing the grid, creating, running and managing applications and monitoring operations.

Together, these three subsystems provide the foundation for executing and scaling existing web applications on grids of commodity servers.

Copyright © 2006 3TERA, Inc. All rights reserved.

What's Unique about AppLogic

The architecture of AppLogic is unique in three important aspects:

Linux and Windows are integral parts of the infrastructure

AppLogic recognizes that conventional OS such as Linux and Windows are very good at managing software on a single computer - what's missing is the ability to manage and operate distributed applications. Rather than trying to replace existing operating systems, or to add a yet another middleware layer with its own complex API, AppLogic treats Linux and Windows as an integral part of the software infrastructure stack.

AppLogic uses virtualization to enable each disposable infrastructure component to run on its own copy of Linux or Windows and focuses on providing the abstractions and services needed at the distributed application level. This approach results in a system that is very robust, while capable of integrating and running existing software unchanged.

The infrastructure is an integral part of each application

Traditionally, you have to build a common infrastructure from firewalls, load balancers, web servers, application servers, database servers, etc. and then deploy multiple applications on it. Disposable infrastructure enables AppLogic to reverse this process by including the required infrastructure within the application. Whenever the application is started, the system manufactures and assembles the infrastructure required to run it. Once the application is stopped, AppLogic tears down the infrastructure built for it.

This dramatically simplifies both the construction and the operation of N-tier applications: building infrastructure for each individual application is much simpler than building and managing shared infrastructure. More importantly, including the infrastructure within each application makes applications self-contained and portable, enabling AppLogic to instantiate them on demand and migrate them from one grid to another.

Distributed web applications are first-class objects

AppLogic treats the entire N-tier application as a single logical entity that can be copied, instantiated, configured, started, stopped, cloned, exported, imported, etc. As a result, once the application has been integrated and tested, it can be manipulated with remarkable ease.

The user can scale an application instance from a fraction of a server to dozens of servers simply by defining how much CPU, memory and bandwidth are to be allocated to that specific instance. Any number of instances of the same application can be executed simultaneously on the same grid. Multiple, unrelated applications can share the grid.

Finally, an instance of an application can be cloned, together with its state, database and content, and exported to run on another grid that may be located half-way around the world.

Copyright © 2006 3TERA, Inc. All rights reserved.

A Grid Operating System for Web Applications

These capabilities enable AppLogic to provide the core set of functions that are essential for running mainstream web applications. Those include:

- Aggregating commodity hardware into a single scalable grid
- Native support for transactional and I/O intensive workloads
- Running an application on different grids without modifications
- Concurrent execution of applications, each with its own resource quota
- Scaling applications from a fraction of a server up to the full grid
- Supporting hardware, middleware and applications from many vendors

In addition, AppLogic implements a number of key services that enable the building of real-world utility computing systems. These include a resource metering system that enables pay-per-use models, a catalog delivery system for distributing and sharing applications and a grid management system that manages a datacenter as a single system.

The following sections will cover in more detail some of the unique technologies that make all this possible. Those include the structure of AppLogic applications, key elements of the disposable infrastructure layer that makes the applications possible and the scalable system services required to support application creation, execution and management.

AppLogic Applications

Applications are the main entities in AppLogic. Making distributed scalable web applications easy and logical is the main purpose behind AppLogic.

An AppLogic application is a single system object that includes everything necessary to run a specific distributed application such as a CRM system, a PBX or an e-commerce application. It includes the application code, HTML pages, templates and scripts, databases and content, but also operating systems, middleware, file storage, load balancers, firewalls and all configuration information needed to reconstruct and run the application on an AppLogic grid. In addition, each application carries a defined resource budget - a minimum set of hardware resources including CPU, memory and bandwidth that are required to run the application, and the maximum resource quota allowed for it.

AppLogic applications are hardware-independent and do not include any references to hardware such as servers, IP addresses and host names, storage LUNs, etc.

This enables AppLogic to provision hardware resources for the application on the fly, to run multiple instances of the same application on the same grid, and to start the same application on different grids without requiring modifications. The process of preparing an application to run is not unlike the "fix-up" performed by traditional operating system loaders when preparing an executable file to run.

Copyright © 2006 3TERA, Inc. All rights reserved.

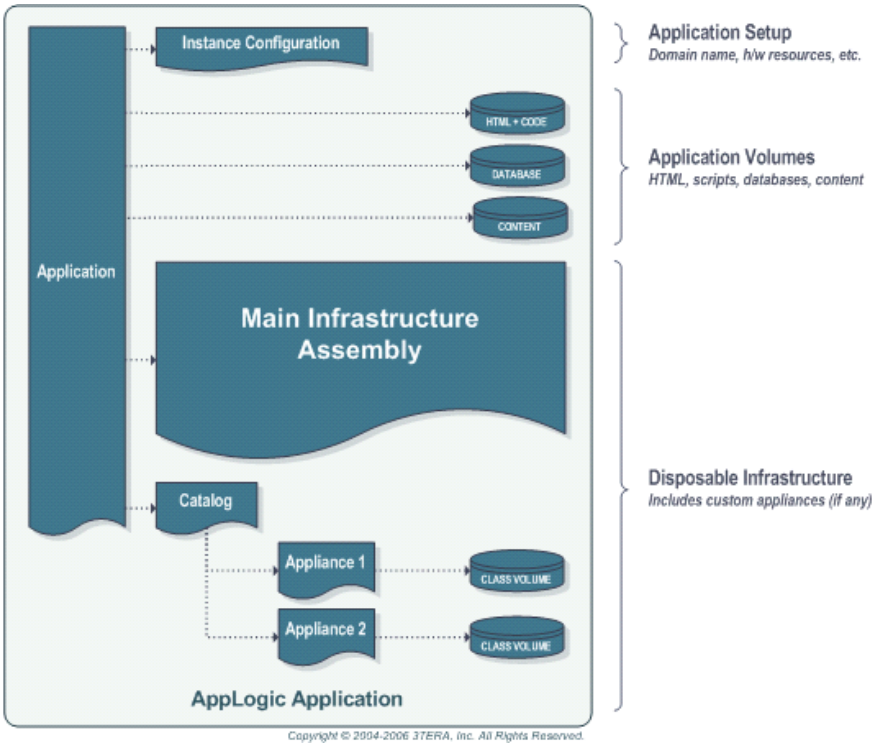


Figure 2. The structure of AppLogic application

Figure 2 shows the internal structure of an AppLogic application. A typical application includes the following three major elements:

Application Setup

The application setup includes configuration parameters such as DNS domain name, public IP addresses, specific hardware budget, admin user name and password and similar values that vary from one instance of the application to another. When instantiating the application, users can specify setup parameters without the need to know how the application is built.

Application Volumes

Each application has one or more volume images containing the application-specific code and content. A typical application will have a single volume on which HTML files, scripts, Java beans, etc. are deployed, another volume for the database and a third for the visual/video content. This is by convention only - any combination of volumes is possible.

Disposable Infrastructure

A key element of every AppLogic application is a set of disposable infrastructure appliances, such as gateways, firewalls, load balancers, web servers, application servers, database servers, file servers, mail servers and so on, and a main assembly that ties them together into a logical structure

Copyright © 2006 3TERA, Inc. All rights reserved.

capable of running the application. This includes all information required to configure each appliance and tie them together.

AppLogic provides a powerful interface that makes it easy to manipulate a distributed N-tier application as one whole. For example, you can create, destroy, instantiate and clone applications. Instantiation will create a new copy of the application ready to run from a known initial state of the databases and other persistent storage, while the clone operation creates a new application that inherits a snapshot of the state from the original application at the time it was cloned. An applications can be exported into a portable, compressed archive and then imported and ran on a different grid.

Since each AppLogic application is stored as a set of text and binary files, it can be checked into any existing version control system, enabling complete control over the configuration, code, data and structure of the application. This makes it possible to extend the well-understood software change management tools to the entire distributed applications.

Disposable Infrastructure

Making infrastructure disposable is a unique capability of AppLogic. Disposable infrastructure is enabled by the recent rapid growth in the speed of commodity x86 processors in comparison with the performance of network and storage hardware. The excess CPU cycles make it possible to implement in software most of the infrastructure functions that needed hardware assistance even a couple of years ago. This enables AppLogic to take the "smarts" out the smart infrastructure boxes, and package them into software objects that are created on demand and composed into complete application infrastructures.

The technological foundation of AppLogic disposable infrastructure is a new distributed component model. The AppLogic component model provides several powerful and unique capabilities:

Existing software runs in components

An AppLogic component is called a virtual appliance. Unlike any other component model in existence, each instance of a virtual appliance executes in a completely virtualized environment, boots its own operating system, application services and other software. This enables AppLogic to package existing software into components that are easy to manufacture on demand from a common class definition.

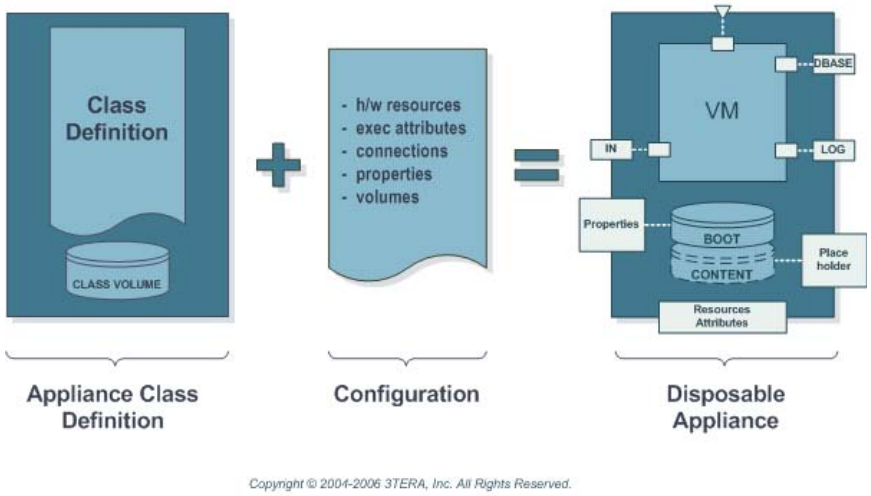
Components are assembled visually

The AppLogic component model enables assembling structures of virtual appliances by configuring them and wiring them together in a browser. This makes it easy to build custom infrastructure for N-tier applications, as well as to visually debug, monitor and operate such applications.

Copyright © 2006 3TERA, Inc. All rights reserved.

Hierarchical composition

The AppLogic component model makes it possible to package structures of interconnected components into new appliance classes that can be manufactured on demand and used in exactly the same way as one would use a virtual appliance. For example, one can assemble a "clustered database appliance" from regular database servers, load balancers and storage, and make a complex cluster as easy to use as a single database appliance. The assembly support is hierarchical, allowing users to create assemblies of assemblies of assemblies and so on, enabling the construction of truly gigantic applications. Indeed, every AppLogic application is an assembly.



Copyright © 2004-2006 3TERA, Inc. All Rights Reserved.

Figure 3. Disposable infrastructure components

Figure 3 illustrates the process of manufacturing infrastructure components on demand in AppLogic. A class definition, consisting of a class descriptor and one or more class volumes, defines a class of disposable virtual appliances, such as a web server, or a load balancer.

The class volumes contain all of the software required to boot and operate an instance of this class, including an operating system (e.g. Linux), application services (e.g. Apache httpd) and anything else the appliance may need. The class descriptor defines the characteristics of the appliance as component, including its inputs, outputs, configuration properties and their default values, volumes, and hardware resources.

When requested, AppLogic produces a running instance of the appliance from the class definition by applying configuration information specific for that instance to the class definition. Each instance consists of a virtual machine, one or more virtual storage volumes and one or more virtual network interfaces. The software included in the instance executes in a completely virtualized environment, equivalent to running on a separate

Copyright © 2006 3TERA, Inc. All rights reserved.

physical server. Network interactions are also virtualized, providing a secure and flexible way to interconnect components.

This process makes it easy to create infrastructure on demand, custom for each application or even application instance, and tear it down when the need for it is no longer there.

System Services

The AppLogic Distributed Kernel provides a set of system services required to implement the distributed infrastructure and application model of AppLogic. The three most important system services include:

Global Volume Store (GVS)

GVS implements a new type of distributed storage subsystem that combines the advantages of a global file system with an object store. The key object supported by GVS is a virtual volume. Virtual volumes exist in a hierarchical namespace, can be created, destroyed and cloned on demand, and are mirrored on multiple servers for read performance and availability.

GVS scales linearly to hundreds of servers in both performance and capacity. Due to advanced caching and network access algorithms, GVS volumes are generally 10-15% faster than local physical disks. In addition, GVS supports the ability to take snapshots of a volume or a set of volumes, which makes it easy to backup entire applications.

Distributed Virtual Machine Manager (DVM)

DVM aggregates the CPU and memory resources of the servers within a grid into a single scalable pool and makes them available to applications. It uses the open source Xen hypervisor to virtualize computing resources. DVM provides the ability to create, schedule, execute and monitor dynamic distributed sets of virtual machines across the entire grid, enforcing their execution attributes, security and hardware resource quotas.

Logical Connection Manager (LCM)

The logical connection manager implements a key service that abstracts inter-component communications. It enables AppLogic to define all interactions between components of an application in terms of point-to-point logical connections between virtual appliances. The interactions are controlled and tunneled across physical networks, allowing AppLogic to enforce interaction protocols, detect security breaches and to migrate live TCP connections from one IP network to another transparently.

Together, GVS, DVMM and LCM encapsulate the distributed nature of the underlying grid, virtualize its hardware resources, and provide the runtime abstractions necessary to implement the rest of AppLogic.

Copyright © 2006 3TERA, Inc. All rights reserved.

A Platform for Utility Computing

The architecture of AppLogic makes it a great foundation for building utility computing services targeted toward general-purpose web applications with transactional and streaming workloads. To further enable utility computing models, AppLogic provides:

Ability to add and remove hardware on the fly

Adding servers requires little more than connecting them to the grid and entering their IP addresses in the grid configuration. The new server is rebooted automatically, made part of the grid and its recourses are immediately provisioned to run applications. To remove a server (or several), the grid operator only needs to mark them as disabled; AppLogic will migrate automatically the virtual appliances and volumes away from those servers and notify the operator when they are ready for removal.

Real-time resource metering infrastructure

The AppLogic resource metering subsystem is an integral part of the kernel. It interacts directly with the scheduler, and the storage, virtual machine and connection managers. This makes it easy to measure and report hardware resources assigned to each application and component, enabling 3TERA customers to implement sophisticated billing systems based on actual resource consumption.

Visual interface in a browser

AppLogic implements an AJAX-based visual interface for creating virtual appliances, assembling application infrastructure, monitoring and troubleshooting applications and managing the system. AppLogic users can integrate, deploy and manage their applications from anywhere in the world using only a browser and basic IT skills.

Catalog delivery system

The AppLogic catalog delivery system provides global distribution, caching and sharing of virtual appliance classes, standard application infrastructure assemblies, and whole prepackaged applications ready to run. The delivery system plugs every AppLogic user into a global network that enables sharing of applications and application infrastructure.

Grid management system

For massive datacenter deployments, AppLogic includes a separate system for managing multiple grids. The grid management system is implemented as an AppLogic application and provides a unified view over the datacenter, enabling the operator to see and manage hundreds of deployed grids as a single system.

Copyright © 2006 3TERA, Inc. All rights reserved.

Summary

The forefront of IT technologies is shifting away from traditional large enterprises. The dynamic nature and scale of Web 2.0 and SaaS pioneers are stressing the abilities of existing IT infrastructure, systems and practices. Each of today's successful web companies have been forced to invent their own ways to manage and scale infrastructure by building their own tools and using a lot of manpower. Those that couldn't scale have seen their market opportunity evaporate.

AppLogic is the first system designed specifically to address the needs of the new IT vanguard. By combining scalable grid architecture, virtualization and disposable infrastructure, AppLogic enables users to deploy existing web applications on grids and operate on them with remarkable ease.

For the first time, the underlying hardware can be treated as an easily scalable resource for applications, instead of a deadweight that prevents them from evolving. Applications can be deployed in minutes, not in weeks. They can be executed, cloned, exported and even scaled dynamically without concern for the underlying hardware.

With AppLogic Web 2.0 and SaaS providers can focus on their business instead of their infrastructure.

About 3TERA

3TERA is a leading provider of grid and utility computing solutions that simplify the deployment, operation and scaling of online services. Customers use 3tera's systems to offer online applications without the need to build and maintain IT infrastructure. For more information, visit www.3tera.com.

Copyright © 04/2006 3TERA, Inc. All rights reserved. 3TERA, 3TERA logo and AppLogic are trademarks or registered trademarks of 3TERA, Inc. in the United States and other countries. All other product names, service marks and trademarks mentioned in this website are trademarks of their respective owners.