



MIGRATING APACHE 1.3 TO APACHE 2.0: THE COVALENT APPROACH

WHY MIGRATE?

Apache 1.3 HTTP servers are recognized as the most secure and reliable web servers available today. With over 20 million Internet sites running Apache, it represents more than half of the total number of installations. Many Apache-based sites report uptimes measured in months or even years, further demonstrating Apache's reliability. Apache has also been unaffected by the security issues that have plagued other web servers. Since Apache 1.3 has clearly met the needs of large numbers of sites, why should current Apache users consider migrating?

Apache 2.0 represents a fundamental re-architecture of serving web-based content. It provides a much more flexible processing model, allowing server performance to be tuned according to the nature of the application and the strengths of the underlying operating system. Architectural changes will allow plug-in modules much more functionality and interoperability. Additionally, the Apache 2.0 architecture provides for plug-in protocol modules, meaning that Apache can support not only HTTP/HTTPS, but also FTP, mail protocols (such as POP3 and SMTP), audio/video streaming protocols, and potentially many others. Apache 1.3 works very well for a wide variety of sites; Apache 2.0 works even better.

Covalent Enterprise Ready Server (ERS), which is based on Apache 2.0, consists of several additional vital components that make Apache 2.0 very attractive for enterprise-wide deployment:

- Centralized browser-based management of hundreds of servers
- Extensible logging application purpose-built for enterprise needs
- Comprehensive security, reliability and management feature set

APACHE 2.0 FEATURES AND BENEFITS

Apache 2.0 architecture is the result of extensive real-world experience with web sites as they have grown and evolved over the past several years, and represents significant improvements in a number of areas. This section highlights those enhancements in some detail. The strong underlying theme that runs through the entire Apache 2.0 architecture is improved performance and greater flexibility, while maintaining the security and reliability that Apache is renowned for.

Multi-threaded Processing

Web servers have historically followed one of two processing models: parent-child processes and multithreaded servers. The Parent-child process model assigns an incoming request to a process, with each process handling one request at a time. In this model, dozens or hundreds of child processes can execute at a time, handling requests concurrently. Each child process manages its own resources like memory and files. The Multi-threaded model creates a single process. Within this process, multiple threads of execution are created, each handling a request. All threads share resources created by the process.

Neither processing model is perfect and both have their own strengths and weaknesses. The primary strength of the multi-process model has been reliability. If an application error causes a process to terminate abnormally, only a single request (and a single client) is affected by the error; other processes remain unaffected and continue processing. However, if an application error causes the process to end in a multi-threaded server, all requests across all threads in that same Unix process are adversely affected.

The primary strength of the multi-threaded model has been performance and scalability on certain hardware platforms. The execution threads within the process are relatively ‘light weight’; i.e. each thread consumes fewer system resources than an entire process does. This means that multi-threaded servers can often handle larger number of simultaneous requests because of the ability to run greater number of threads within a single process as compared to the total number of processes possible on the system. Also, creating a new thread of execution is much faster than creating a complete process, which improves the responsiveness of the server to handle peak loads and spikes.

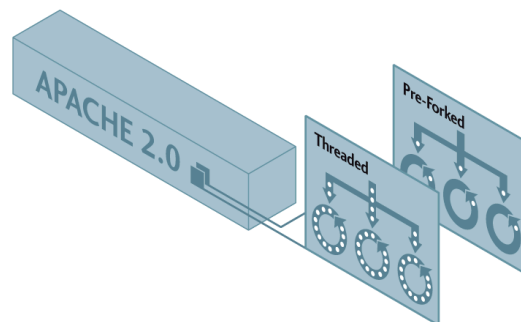
Apache 1.3 uses the parent-child processing model with some enhancements. Rather than creating a new process every time a new request is made, child processes are created in advance (often called pre-forking). Apache continually monitors how many child processes are in use at any given time, and can pre-fork more processes (up to a limit specified in the configuration file) if all the child processes become busy. While this eliminates much of the process starting overhead (except when Apache starts at the outset), there are still limitations set by the operating system and hardware resources that may prevent enough processes being available to handle all incoming requests. The usual approach to this issue has been to add more hardware, either within the server or by adding additional servers.

Apache 2.0 introduces a new, hybrid processing model (Figure 1). This model uses multiple child processes, each process containing a fixed number of multiple threads. Apache monitors the total number threads in use, and creates new processes as required when the number of unused threads falls below a specified threshold. This model provides superior performance and scalability, while maintaining much of the reliability of the pre-fork model. Apache 2.0 also provides a pre-fork model very similar to the Apache 1.3 model, and a variation of the multi-threaded model especially suited for Virtual Hosting environments that executes child processes under specific user ids for enhanced security.

The result of this architectural change is the ability of Apache 2.0 applications to achieve greater scalability on a given hardware platform, while maintaining the choice of using the familiar and reliable model from Apache 1.3.

Figure 1

Multiple Processing Methods
Apache 2.0's Multiple Processing Models let you balance robustness and scalability, a feature unavailable on Apache 1.3.



Better Support for non-Unix Platforms

Apache 1.3 supported a wide range of platforms, including nearly all UNIX variants such as Solaris and Linux, and the Microsoft Windows platform. While the most recent releases of Apache (1.3.20 and newer) addressed many of the performance and reliability issues of Apache on the Windows platforms, it still lagged behind that of Apache on UNIX. Primarily, this was due to the initial design assumptions when Apache was being coded, namely a Posix-based environment. On non-Posix systems, various coding “tricks” needed to be done to force the operating system fundamentals to fit into the Apache framework. Apache 2.0 overcomes that inconsistency.

Part of the Apache 2.0 project has been the Apache Portable Runtime (APR) project, which facilitates platform-specific implementations of Apache modules (including the processing models) without requiring changes to Apache's core source code. It is the function of APR to map the functionality needed by Apache to what the

underlying operating system provides. In effect, APR provides a compatibility layer between Apache and the operating system. This means that the actual Apache code itself can be much cleaner and operating system independent. If Apache needs to open a network socket for example, it performs that same function call on all operating systems. It's the function of APR to then map that call to the correct implementation for that specific operating system, allowing for optimal use of each operating system's strengths.

Multi-Processing Modules (MPMs), which implement the core request processing features of Apache, can now contain code that is optimized for the specific operating system as well. This is of particular significance for Microsoft Windows (NT and 2000) users, as it allows a processing model to be implemented that takes advantage of native Windows features to dramatically increase performance of Apache. The Windows MPM implements a child process/multi-threaded model that uses native Windows threading mechanism. Early benchmark results indicate that Apache 2.0 performance on Windows NT/2000 is comparable to UNIX performance on systems with similar hardware configurations as well as on par with the performance of IIS on that system.

Plug-in Module Enhancements

One of the most powerful features of Apache 2.0 is "Filtering", which is the ability to combine modules together to perform a series of actions on a given request or response. For example, a module could call functions within the SSL module to provide encryption services. In fact, this has greatly facilitated implementation of Covalent's FTP over SSL. SSL itself is now implemented in a much more straightforward way, due to the changes in the Apache API with 2.0. Additionally, modules can assign functionality to other modules and order among themselves which modules should process the request first. A final improvement on Apache 2.0's module implementation is the concept of per-module hooks. Instead of Apache itself defining the stages of a request that modules have access to the request data (for example, the authentication phase, or the filesystem mapping phase), modules can "hook into" the request handling at any stage that makes sense for its operation. By preventing these hooks from needing to be defined statically by Apache, the actual Apache API is much more robust.

Because of the fundamental changes in the plug-in module architecture, existing Apache 1.3 modules will need to be revised for Apache 2.0. The standard modules available with Apache distributions from the Apache Software Foundation have already been converted, and in many cases enhanced, for Apache 2.0. Other open source modules have already been or are in the process of being converted, as are commercial third-party modules.

Support for Multiple Transport Protocols

Apache 1.3 was designed to serve a single protocol family: HTTP/HTTPS. Much of the code that works with the HTTP protocols in Apache 1.3 is embedded within the core of Apache, making the support of additional protocols difficult at best. As web servers move beyond serving only HTTP protocols into areas such as streaming media, messaging protocols, and more, the base architecture of Apache also needed to change to accommodate these changes.

Apache 2.0 uses a plug-in protocol model that allows additional transport protocols to be plugged into Apache. In effect, the protocol handling aspects are implemented as a standard Apache 2.0 module. This approach greatly simplifies operational logistics. For example, common set of authentication services can be uniformly applied for both HTTP and FTP requests, reducing the risk of any security loopholes. Also, HTTP and FTP requests can safely share directory structures. Other services such as logging can also be shared across protocols, providing a single point for monitoring and auditing. FTP is available as an optional licensed product for Apache 2.0 as part of Covalent's Enterprise Ready Server. Other protocol modules can be expected both from commercial and open source developers.

It is Still Apache

Even with all of these architectural changes, Apache 2.0 retains the advantages that Apache users have enjoyed for several years. The open source development process yields a product that has been scrutinized and optimized by a large body of developers with no other motive than to make it the highest quality product. Although much of the code has been rewritten for efficiency and to provide a more flexible architecture, the combined experience of the core Apache development team has yielded a product with all of the reliability and even more performance than the Apache 1.3 generation products. Apache users will be able to deploy Apache 2.0 based systems in critical applications with even greater confidence than before.

MIGRATION PROCESS

Once an organization has made a decision to migrate some or all of its current sites from Apache 1.3 to Apache 2.0, Covalent recommends that a structured process be followed to ensure the best results with minimum of risks. This section describes some of the migration best practices followed by the Covalent Professional Services organization.

Catalog Existing Applications

It may seem obvious, but the best place to start is to understand what you have. Because many Apache sites have grown organically over time, it is a good idea to take a comprehensive inventory of applications and to document the dependencies of those applications.

Module Use

As mentioned earlier, Apache 2.0 requires the plug-in modules be revised for Apache 2.0. The good news is that all of the most common open-source modules have already been revised by ASF. Also, key third party modules from companies like Covalent are also available for Apache 2.0 today. As you examine your applications, note each module in actual use on the site, remembering that you may have some modules in your configuration files that are actually not being used at all. For each module, note whether an Apache 2.0 version is currently available, or if the module has been replaced by a new module with equivalent or greater functionality. For example, the Apache 1.3 modules *mod_log_agent* and *mod_log_referrer* are no longer included with Apache 2.0; their functionality is available through the *CustomLog* directive of *mod_log_config*.

Configuration File Structure

Many configuration directives are unchanged from Apache 1.3. There are some directives that no longer exist, including *BindAddress* and *Port*, which have been replaced by the much more flexible *Listen* directive. Likewise, *AddModule* and *ClearModule* directives are no longer required, because the order in which modules load can now be determined by the modules themselves. As part of reviewing your configuration files, it may be useful to remove directives that no longer apply to your current installation, and perhaps to restructure the file to make it clearer to work with during the migration process. Covalent provides a useful Perl script which will take your existing Apache httpd.conf file and “port” it to the new Apache 2.0 httpd.conf format. This script is available as a free download from <http://apache.covalent.net/> or as an online service.

This is also a good time to review, and document if needed, the structure of content on your servers. Particularly if you want to restrict access to portions of your site to authorized users, you can often gain efficiency and clarity in the configuration process by rearranging the structure of content directories.

Plan New Installations

Once you’ve got a good idea of where you are, it is time to plan how you will be performing the migration to Apache 2.0. Here are some considerations that may help your migration go smoothly.

1. Module Availability

As mentioned earlier, you need to confirm that the Apache plug-in modules that you’re using today are available for Apache 2.0. You also have to be aware of some dependencies that may not have existed earlier. For example, if you use *mod_perl* and decide to use the multi-threading Apache server, you

have to make sure that you have a multi-threading compatible version of Perl available (5.6.1 or later) on your system. Covalent's ERS ships with such a version of Perl.

2. Multi-threading vs. Pre-Fork

Deciding on which processing model (multi-threaded or pre-forked for UNIX systems) you're going to use should be based on weighing performance versus application reliability. While the multi-threaded processing model is in itself very reliable, like all multi-threaded models application errors may affect a number of requests in progress at the time. By adjusting parameters in the configuration file, you can adjust the balance of processes vs. threads. If you specify relatively few threads per process but a large number of total process, you gain some of the advantages of multi-threading while limiting the affect of any application errors to a smaller number of requests.

For some sites, staying with the pre-fork module will make the most sense. This creates a site with similar performance characteristics as the current Apache 1.3 site, with all of the site stability. Some third party modules, which are not thread safe, may also cause you to stay with the pre-fork implementation. Since Apache 2.0 uses the threading implementation of the underlying operating system, your choice of processing module should consider the robustness and reliability of those core thread libraries.

Multi-threaded versus pre-fork choices are made when the Apache server binary is compiled, as opposed to the run time directives stored in the Apache configuration files. This compile choice is required because compilers for operating systems can make substantial performance enhancements at the code level if the compiler knows that it is compiling for a threaded environment. For users who build their own Apache servers, this means that you'll need to create both a threaded and pre-forked binary if you intend on supporting both processing models. Covalent Enterprise Ready Server comes complete with the pre-compiled binaries for all processing models optimized for all supported hardware and operating system platforms.

Windows NT/2000 users have a simpler decision to make: only a single processing model, a variant of the multi-threaded model, is available. This model offers significantly better performance and scalability for Apache 2.0 than the Windows versions of Apache 1.3.

Testing

Before committing to any changes in a production environment, prudence dictates that a test plan be created and executed. If no changes in site functionality are planned, testing can be based on a regression test of the current application, along with confirming performance and rehearsing the physical migration of production. In many cases, a migration to Apache 2.0 will be accompanied by additional site functionality. Best practice in these circumstances suggests performing the migration to Apache 2.0 of the current (unchanged) site, and then adding the new functionality will likely produce the best results. This method allows an easier debugging process, since you're only attempting to change one thing at a time.

1. Performance

The migration process is a good time to repeat previous performance and load tests to confirm that you're getting the expected results from Apache 2.0, or to create a new test harness that reflects the current and expected traffic patterns for your site. There are many open source and commercial performance and load testing tools available today, but perhaps the most critical element in testing is to understand what you're attempting to test, and to understand all of the elements that make up the performance characteristics of your site.

While web site testing is an art and science of its own, in general two characteristics are generally of interest: site response time, which is how quickly a transaction of a given type can execute when the site is idle, and site scalability, which is how the transaction duration varies under increasing load.

Within those categories are a number of individual measurements of interest, including how long from the time a request is sent until a response is generated, and how long a complete web page response takes to complete.

2. Rehearsing for Migration

In most situations, production web sites will need to be migrated using the same systems that currently host the Apache 1.3 sites. Since the downtime of production systems generally needs to be minimal (or non-existent if possible), a carefully planned and tested production migration process needs to be created.

In general, the following steps will need to be followed for each server to be migrated:

- The Apache 2.0 system and associated modules are installed in a directory structure parallel to the currently executing server. It is important that the existing installation isn't overwritten during this process.
- The configuration file from the existing site is migrated, making whatever changes are required.
- If possible, the site is brought up on port numbers different from the production site. For example, if the production site is running on ports 80 and 443, use ports 8080 and 8443 for the new installation. At this point, a sanity check can be done on the new site. This final check isn't meant to be replacement for a more complete testing process.
- The old Apache server is halted, and the new web server started, using the production port numbers.
- If all has gone well, the interruption in the site availability should be minimal.

This process should be thoroughly tested in a non-production environment before attempting it in a production environment.

Cut-over

Moving an entire web farm to Apache 2.0 may be somewhat time consuming, particularly if the site is powered by multiple web servers. If you currently use load balancing and/or fail over environments for your web site, it is possible to perform an entire migration without leaving production status, as servers still active under previous configurations continue to operate while the new servers are brought on line.

Because converting, and then managing and monitoring large Apache web sites can be challenging, you should use automation wherever practical. Covalent's Enterprise Ready Server consists of components that assist in centrally configuring, installing, managing and monitoring multiple web servers in production.

Covalent Technologies hosts several Apache-related discussion forums on its website at <http://apache.covalent.net/discussions/intro.php>. These forums are moderated by several Covalent employees, who are generally regarded as authoritative experts in their respective domains. One of the forums focuses specifically on Apache 1.3 to 2.0 migration.

COVALENT ENTERPRISE READY SERVER, POWERED BY APACHE 2.0

Covalent Technologies, Inc. has created a comprehensive set of products based on Apache 2.0. They are designed to add value to the Apache 2.0 platform, adding functionality required in the enterprise-computing environment. If you're contemplating a migration to Apache 2.0, Covalent Enterprise Ready Server can help make the migration process smooth, while adding significant new capabilities to your Apache environment.

Covalent ERS is available in the following two flavors:

Covalent Enterprise Ready Server, Fast Start Edition

The Enterprise Ready Server, Fast Start Edition (Fast Start) is designed for use in development environments and some intranet applications. It features the Apache 2.0 HTTP server, Covalent's famous SSL product (née "Raven"), and a variety of development environments. All components are supplied in pre-built (binary) form, and are optimized for the target platform (Linux, Solaris, HP-UX and Windows NT/2000).

1. Components

- Apache 2.0 HTTP Server
- Covalent SSL
- Standard Apache 2.0 plug-in modules
- Mod_perl for developing Perl based applications
- Mod_PHP for developing PHP based applications
- Tomcat 4.0 for developing Java Servlet and Java Server Page (JSP) applications

2. Installation and Migration Process

ERS, Fast Start Edition includes a comprehensive, easy to use graphical installer that provides a point and click interface to chose the specific modules to be installed. To facilitate the conversion of Apache configuration files, Covalent has made available a "Configuration Converter" tool on its Web site: (<http://apache.covalent.net/tools/index.php>). It is available free of charge. It backs up the original configuration file and edits it for syntactical changes between Apache 1.3 and Apache 2.0.

Covalent Enterprise Ready Server

Covalent's Enterprise Ready Server (ERS) provides a full set of Covalent products that make it ideal for production web environments. In addition to the tools provided in the Fast Start Edition, ERS also includes new capabilities in system management and monitoring, authentication, state and session management, and logging. Like Fast Start, ERS features pre-built, optimized binaries for Apache and all open source modules.

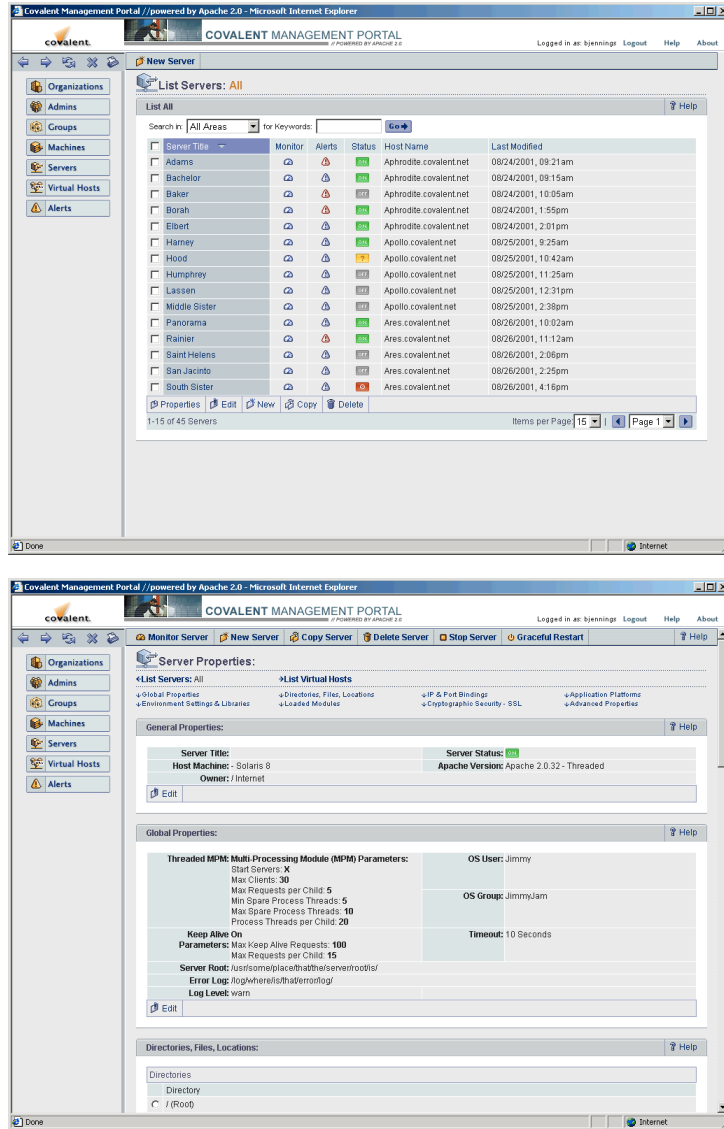
1. Components

- All of the components of the Fast Start Edition mentioned above.
- Covalent Management Portal (CMP): Comprehensive configuration management for Covalent ERS servers, including creating and modifying configuration files, and updating configuration files and Covalent software on servers under management. Using a web-based graphical interface, all the data pertaining to management of the web farm is stored in a relational database. Using role-based authorization model, configuration templates can be applied across multiple web servers, and all the required installation binaries can be pushed out to the designated server machines. All instances of installed web servers can subsequently be controlled via the portal: start/stop, graphical monitoring of real-time statistics, alerts etc. This is the only tool of its kind for Apache web servers, and offers a bounty of operational features for effective centralized management of hundreds of web servers.
- Covalent Logging Services: Centralized logging facilities for all web servers in an installation, including optimized log transmissions and customized log extraction formats. All web servers transmit their logs (via HTTP) to a central logging server, which can process the incoming log data as per the specified rules. This can lead to near-real-time analysis and reporting of log data, permitting organizations to have a centralized view of their web-farm. All of these out-of-the-box features eliminate the need for tedious scripting, file transfers, manual log-rotation, cron jobs etc., resulting in best practices logging procedures, higher productivity and lower operational costs.
- Covalent Authentication: Comprehensive authentication features including file, database, and LDAP based authentication. Pluggable authentication mechanism for custom integration.
- Covalent State and Session Management: User state and session tracking via cookie or URL based mechanisms.

2. Installation and Migration Process

ERS can take advantage of the CMP throughout the installation process. Configuration sets can be created and modified, and Covalent Apache 2.0 software and supporting modules installed across multiple servers, all through the centralized CMP. It provides an easy to use interface for configuring hundreds of Apache module directives (Figure 2). Updates and changes to both software and configuration can be easily made as your environment changes. Servers can be commissioned and de-commissioned at will through the portal.

Figure 2: CMP can be used to centrally configure and deploy hundreds of web servers.



3. Monitoring and Alerting

To round up the enterprise-class characteristics for deploying Apache 2.0, CMP includes a comprehensive set of features for central monitoring of all the deployed web servers (Figure 3). For each web server, you can view real-time statistics like the number of concurrent connections, process capacity utilization etc., which are essential pieces of information to ensure efficient server performance. Additionally, you can view graphs on data accumulated over a periods of time, to make informed decisions about the website's scalability, visibility, reliability etc.

Webmasters can also setup alerts to be notified in real-time about certain critical server events like shutdown and capacity over-utilization (Figure 3). These alerts can be transmitted to mobile devices via email, pager, SMS etc., or viewed within CMP.

Figure 3: CMP allows you to monitor web servers' real-time and historical statistics graphically.

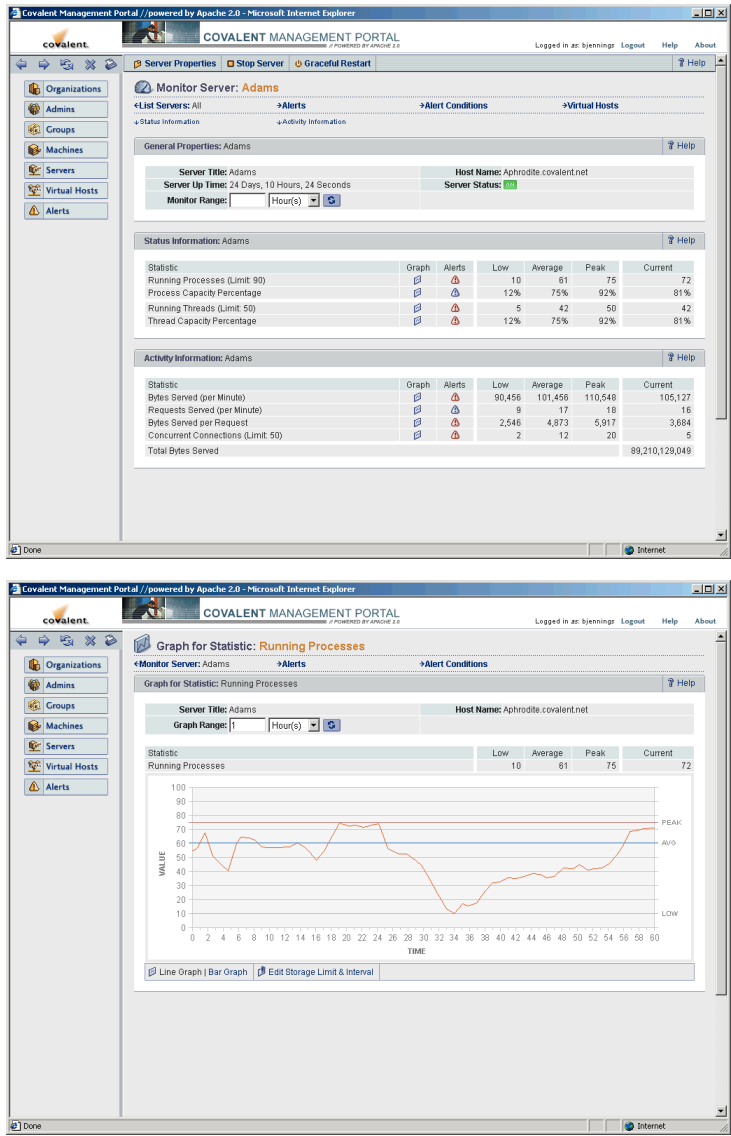
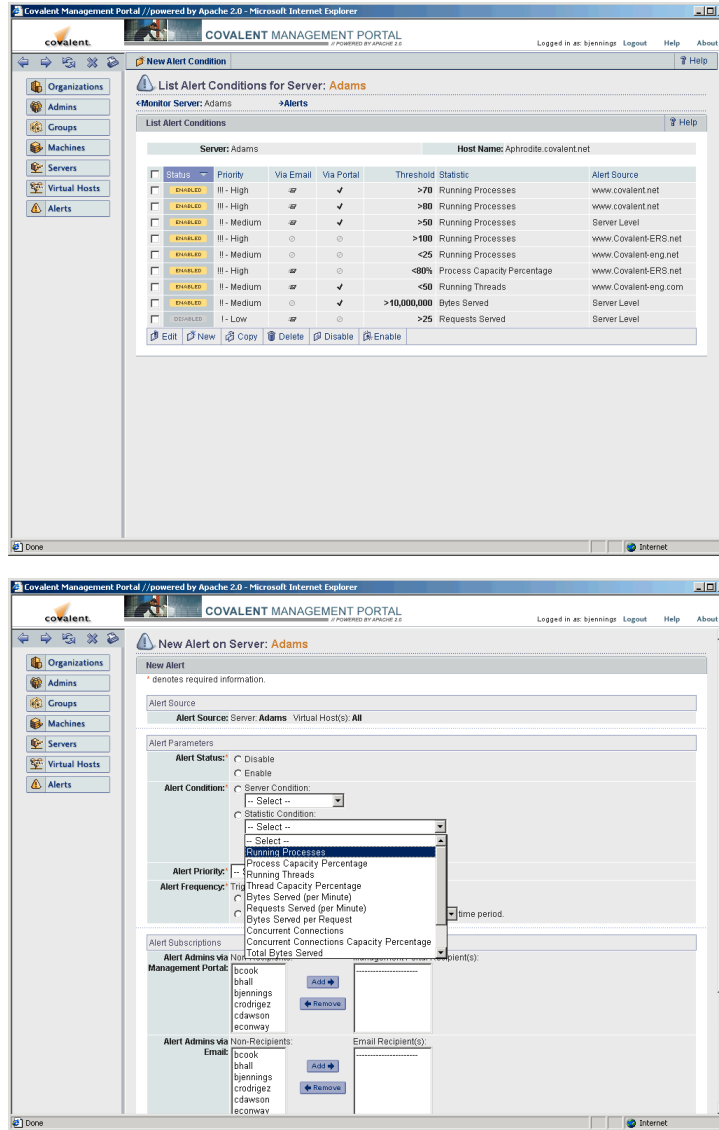


Figure 4: Real-time alerts can be defined and reported in CMP.



The Covalent Advantage

Covalent Technologies, Inc. offers an unmatched level of expertise in Apache products, and has augmented Apache products to be truly ready for the enterprise. These products, combined with Covalent's expert professional services team and world-class support, provide your organization with a unique ability to take advantage of the new generation of Apache.

SUMMARY

Covalent Technologies, Inc. offers an unmatched level of expertise in Apache products, and has augmented Apache products to be truly ready for the enterprise. These products, combined with Covalent's expert professional services team and world-class support, provide your organization with a unique ability to take advantage of the new generation of Apache.

whitepaper

Covalent
technologies inc.



Covalent Technologies, Inc.
303 Second Street, Suite 375 South
San Francisco, CA 94107
415/856-4200
www.covalent.net

Sales 415/856-4245 or 800/444-1935