

SDC – The Service Delivery Controller

In his FrankenSOA¹ analysis published in Network Computing, Andy Dorman gave a comprehensive and well-informed assessment of the intermediary software that has sprung up to support SOA (Service Orientated Architecture) deployments.

Some of this software arises to address fundamental problems with SOA, such as a lack of standards, which make it difficult to marry competing vendors' SOA platforms together. Other software arises to address point problems such as application security². Further software products arise to address management and governance problems as the early deployment of the technology outgrows its maturity.



Andy Dorman finishes by assessing the role of Application Delivery Controllers (ADCs), considering how their functionality overlaps with much of this SOA intermediary software. Many users are coming round to the idea that the established, mature ADC market has a lot to offer SOA deployments in terms of performance improvements, fault-tolerance, performance monitoring and future-proof integration in a world of evolving standards and rapidly changing products.

Yes, the majority of SOA deployments can get away without an ADC, just as many websites get away without a load balancer, but the fit between the two is very strong and well worth a further look.

The promise of SOA

SOA rose to prominence promising an agile, flexible application infrastructure, closely aligned with business needs and processes and benefiting from rapid development and code reuse. The reality so far has fallen somewhat short.

Experience so far is that SOA deployments are fragile. Each application depends on multiple components distributed over different application servers on the network. These components may in turn depend on additional components. A failure or a denial-of-service in one component will ripple through to cause failures in the many SOA applications that depend on it.

SOA applications tend to perform more poorly than the monolithic applications they replace, and are more resource-hungry. Network latency affects performance. Bulky XML messages are costly to send, and repeated XML serialization (processing) is resource-intensive. Authentication and validation operations are duplicated across each component.

Finally, as SOA deployments grow, they become increasingly hard to manage and maintain. Administrators have to track and maintain the multiple versions of services depended on by different applications and subtly different standards.

¹ www.networkcomputing.com/channels/appinfrastructure/showArticle.jhtml?articleID=199905816

² In almost any emerging IT technology, security is the last thing to be fully addressed. Security solutions invariably start off as point solutions (and often stay that way, e.g. anti-virus software).

Learning lessons from HTTP – The Application Delivery Controller

Many of these challenges are familiar to web site administrators. For many years, they have been solving similar performance, reliability and management problems. A key tool is the Application Delivery Controller (ADC), also referred to as a Load Balancer or Traffic Manager.

ADCs do a lot more than simply load-balance TCP requests across a cluster of servers to achieve scalability and fault tolerance. They employ performance optimization techniques, ranging from TCP and HTTP optimization to SSL Offload and Content Caching, to improve the performance of the services they manage. Health monitors detect a range of application-specific errors, and bandwidth and rate shaping capabilities ensure that all users get a fair level of service.

Perhaps the most interesting feature of an ADC is its ability to understand the entire of the application traffic – not just the HTTP protocol, but the nature and meaning of the messages sent within. An ADC can use this to perform security checking, authentication and content-based routing. For maximum flexibility, this capability can be exploited by a scripting language within the ADC itself (such as ZXTM's TrafficScript™). Very sophisticated traffic management policies can then be constructed to meet the specific business and technical needs of each organization.

Teaching ADCs to speak the language of SOA...

An SOA deployment needs to manage traffic in many of the same ways as a web site, but with one important difference.

SOA deployments generally use SOAP (Simple Object Access Protocol) to communicate between discrete SOA components (called 'Web Services'). The SOAP protocol uses HTTP as the transport mechanism – something that ADCs are intimately familiar with – but the messages it sends are much more complex, containing large and rich XML data.

XML data cannot be inspected and modified using the simple search and replace operations used for web requests and web content. Instead, standards like XPath and XSLT have been developed in order to inspect and process XML documents³.

Vendors like Zeus (ZXTM Traffic Manager) have added XML processing capabilities into their ADC devices. This capability allows their ADCs to reliably inspect and distinguish

³ **The Computer Science bit**

SOA traffic uses the SOAP protocol, which sends XML messages over the network using the HTTP protocol. In the field of formal languages, XML is a 'context free' language.

Any attempt to parse an XML document with string searches and regular expressions is doomed from the start. That's because regular expressions can only match the limited class of 'regular' languages.

'Context-free' languages are much more sophisticated than 'regular' languages, to the extent that it is logically impossible to parse XML documents with regular expressions.

The XML industry has developed a standard called XPath for this purpose. An XPath expression can analyze an XML document accurately, in the way that a regular expression can analyze simple text. XSLT is a related standard that can modify and restructure XML documents.



between different XML messages, applying security policies and content-based routing within SOA environments.

This XML intelligence means that the benefits that an ADC brings to web site can now be brought to bear on SOA deployments as well. In this context, perhaps it's appropriate to refer to a new term – a 'Service Delivery Controller'.

The Service Delivery Controller

A Service Delivery Controller (SDC) is an ADC that can understand the XML messages that SOA systems use.

The benefits that an SDC brings from the mature, proven ADC market are highly relevant to SOA.

What are the benefits of using an SDC?

Flexibility

An SDC can manipulate the XML-formatted SOA messages. It can assist an administrator in drawing together different applications and web services. These services may use subtly different versions of the evolving SOA standards, and the SDC can perform mediation between these different implementations. This greatly reduces the risk that an SOA deployment may become impeded by the need to support incompatible client implementations, or may require significant rework when new generation components are added.

Content Based routing, load balancing and high-availability clustering

A little like a high-performance Enterprise Service Bus (ESB), an SDC can act as a central clearing house for SOA traffic, routing SOA requests to the correct web service instance. The load-balancing, session persistence, health monitoring and failover capabilities of an SDC let you deploy clusters of SOA web services for improved reliability and performance.

Run-time monitoring and reporting

Health monitors report on the correct operation of SOA web services, and service level monitoring allows you to chart the responsiveness of each Web Services component used in an extended SOA transaction. This performance information – knowing which SOA components are slowest - is vital when diagnosing bottlenecks, determining critical paths and planning resource allocation.

Security

The majority of SOA web services traffic comes in through port 80. The traffic inspection and security capabilities of an ADC are a good starting point when constructing an application-aware security perimeter and the XML inspection allows sophisticated security policies to be created. The security policy can be used in conjunction with the content-based routing. Traffic from trusted sources or for secured services can be routed directly into the SOA application, and other traffic can be routed to a specialized security gateway as required.

Performance

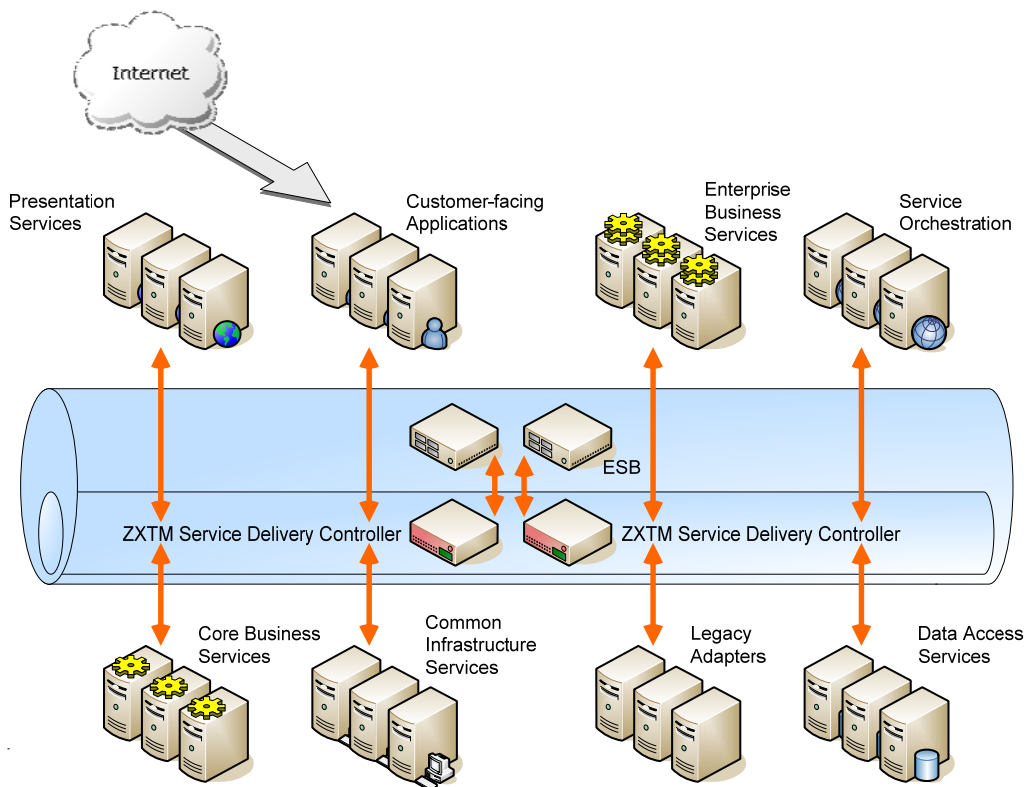
Much of the technology in an ADC is geared towards improving the performance of HTTP-based services, and this technology is directly applicable to SOA web service. TCP offload and HTTP optimizations improve service responsiveness, and SSL offload frees up CPU resources on SOA servers. By performing XML pre-processing and normalization in the SDC, this can further reduce the processing load on the SOA servers, improving performance and capacity.

Conclusion

Using an XML-aware ADC as a Service Delivery Controller can address many of the problems that hold back large or high traffic SOA deployments.

An SDC fits best when considered as a front-end to an Enterprise Service Bus (ESB). ESBs implement many key capabilities within an SOA system – orchestration, routing and asynchronous messaging – but sitting in the middle of the SOA network, they create their own choke point and point of failure.

An SDC can sit in front of an ESB, or better still, a high-availability cluster of ESBs. It can manage traffic on behalf of the ESB, performing tasks it is specialized for (such as SSL offload, XML processing, health monitoring, performance monitoring and content-based routing) without having to invoke the ESB. Only messages which need special handling need be passed up the stack to the ESB.



The ZXTM Service Delivery Controller provides XSLT Transformation, Routing, Failover, Scaling, Draining, Offload and Acceleration, Prioritization, Service Level Monitoring and Security and Validation services on behalf of the Enterprise Service Bus.



Customers who have used Zeus' ZXTM traffic manager in this way have reported dramatic performance improvements in their ESB deployment, including:

- 15-fold increase in transactions per second and 15-times reduction in response time when performing XSLT transformations between schemas
- 15-to-20-fold increase in transactions per second for content-based routing

Application Delivery Controllers are mature, proven systems that bring huge benefits to networked services like web sites and applications. With the addition of XML awareness, and the power to implement intelligent traffic management policies using configuration languages like ZXTM's TrafficScript™, they promise to bring the same benefits to administrators and developers rolling out SOA deployments.

Find out more

- www.zeus.com/products/zxtm/ - Zeus' ZXTM Traffic Manager
- Building a Service Oriented Network using a Service Delivery Controller: Zeus White Paper, www.zeus.com/library/white_papers.

