



Akamai

THE TRUSTED CHOICE FOR ONLINE BUSINESS



Improving Application Performance Over The Internet

David Belson
Director, Application Performance Services
May 5, 2005

Application Performance: Utopia vs. Reality



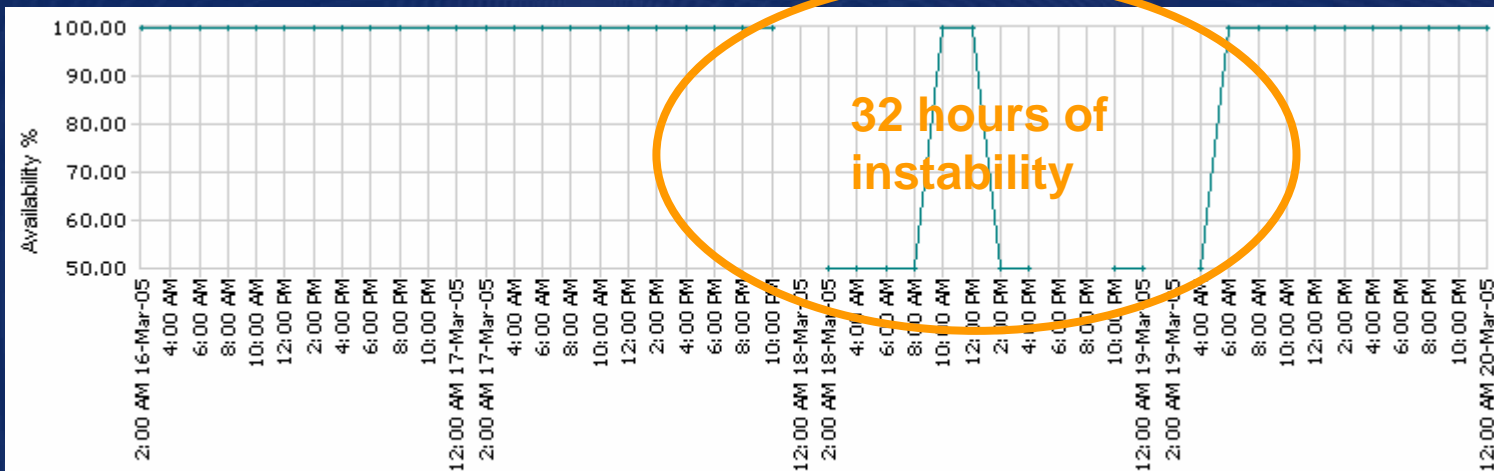
Utopia:



Reality:



Reality Can Be Ugly



© 1998-2004 Keynote Systems, Inc.

What Can Go Wrong?



- Performance for a distributed user community
 - Interactive applications need to make many roundtrips
 - Peering is optimized for economics, NOT QoS
 - BGP is insensitive to congestion
 - Routes take time to stabilize after changes
- One-off events that affect availability
 - Congestion and failures cause outages and slow response
 - Single points of failure with centralized infrastructure
 - DDoS attacks directed at your infrastructure and at networks
- Capacity Concerns
 - Time to manage growing infrastructure
 - Expensive to scale – unusual peaks can overload a site

The Solution



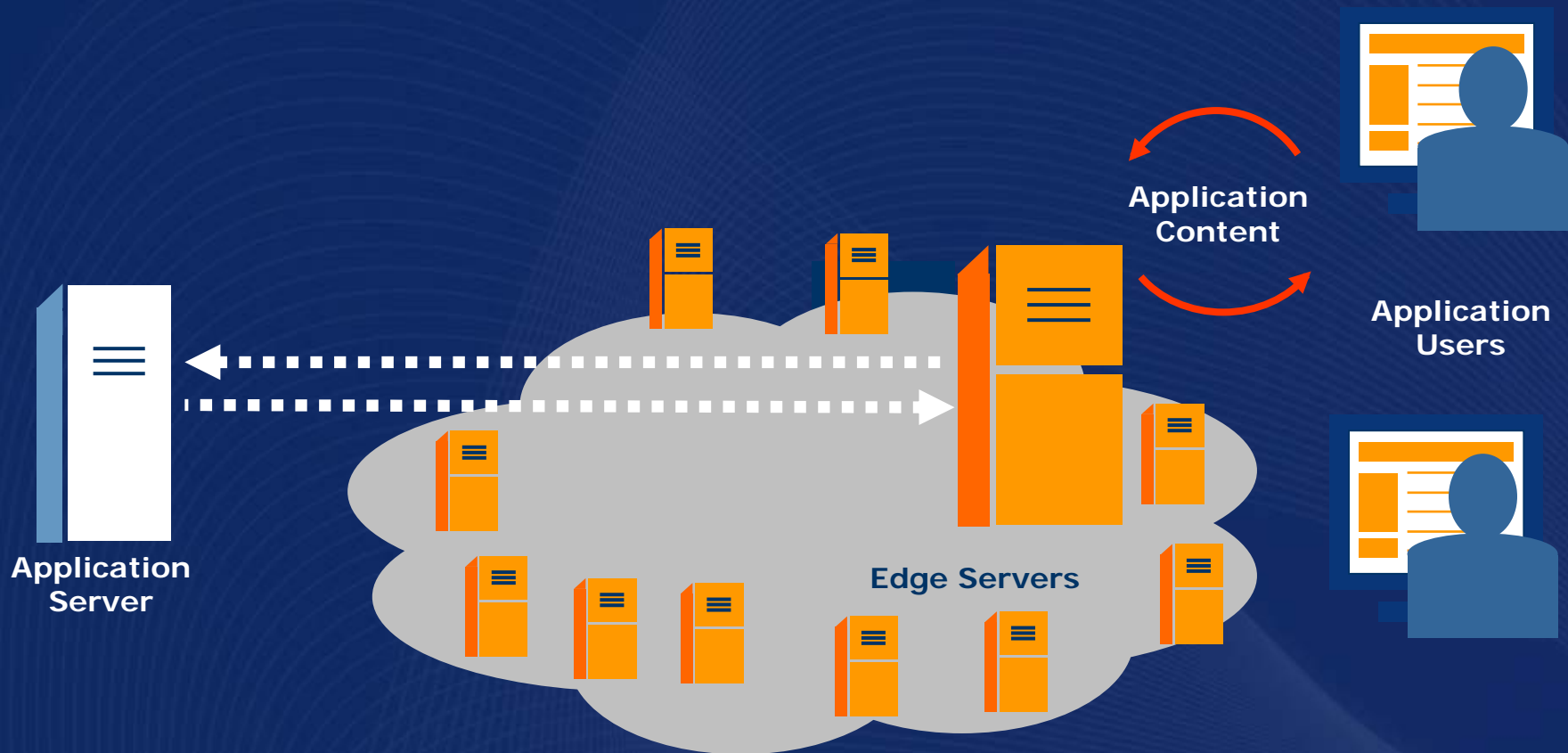
- **Distribute** the infrastructure
 - **Move data closer** to the users
 - **Avoid long trips** through the network
- When long trips are unavoidable...
 - Send data by the **best possible path**
 - **Optimize communications**

Distribute The Infrastructure



- Move content into the network
 - Caching and delivery from the edge provides performance improvements and transparent scalability
 - Enterprises can often cache upwards of 50% of application content
- Move processing into the network
 - Simple template structure (ESI, XSLT)
 - Can render many dynamic sites (e.g., portals) highly cacheable, reducing long-haul bandwidth significantly
 - Edge processing
 - Move application components to the edge
- **Scale matters**
 - You're only as good as the number of nodes near your users

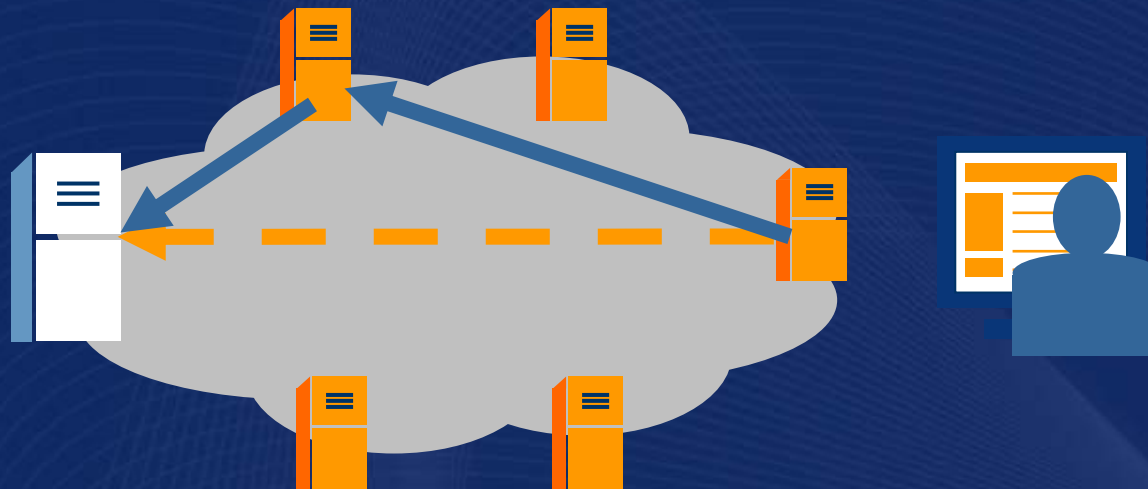
Avoid Long Trips



- Dynamic mapping – send users to optimal edge servers
- Use real time knowledge of Internet conditions
- **Scale matters**: more locations to measure from

When Long Trips Are Unavoidable

- Send data by the best possible path
- BGP ignores congestion and latency, and is slow to route around failures



- Optimize paths by finding alternate routes through intermediate servers in the distributed infrastructure

Optimize Communications



- Persistent Connections
 - Avoid TCP setup and slow start over long-haul
 - Avoid repeated SSL session negotiation
- Prefetch data not in cache
 - Stream data over long-haul in advance of user requests
- Transport Optimization
 - Streamline communication between controlled endpoints
- Compress whatever you can
- **Scale matters:** more pairs of nodes to optimize between

Summary



- Distribute Infrastructure
 - Move content and processing into the network
- Avoid Long Trips
 - Deliver application content from servers close to users
 - If you can't avoid, send data by the best path possible
- Optimize Communications
 - Leverage persistent connections
 - Prefetch embedded content
- Each solves a piece of the problem – together they make a much bigger impact on performance
- **Scale matters** for user proximity, network measurement, and connection optimization