Capacity Planning Tool for Streaming Media Services

Ludmila Cherkasova
Hewlett-Packard Laboratories
1501 Page Mill Road,
Palo Alto, CA 94303, USA
cherkasova@hpl.hp.com

Wenting Tang
Hewlett-Packard Laboratories
1501 Page Mill Road,
Palo Alto, CA 94303, USA
wenting.tang@hp.com

Abstract *The goal of the proposed capacity planning tool is to provide the best cost/performance configuration for support of a known media service workload. There are two essential components in our capacity planning tool: i) the capacity measurements of different h/w and s/w solutions using a specially designed set of media benchmarks and ii) a media service workload profiler, called MediaProf, which extracts a set of quantitative and qualitative parameters characterizing the service demand. The capacity planning tool matches the requirements of the media service workload profile, SLAs and configuration constraints to produce the best available cost/performance solution.*

Categories and Subject Descriptors: *H.5.1 Multimedia Information Systems.*
General Terms: *Measurement, Performance, Design.*
Keywords: *Media server benchmarks, media server capacity, measurements, workload profiling, SLAs, capacity planning.*

1. INTRODUCTION

The ability to plan and operate at the most cost effective capacity is a critical competitive advantage. In this paper, we consider a scenario where a service provider, supporting a busy media site, faces a necessity to migrate the site to a new, more efficient infrastructure. We assume that a service provider collects the media server access logs, reflecting processed client requests and client activities at the site. Thus the problem is to find the best cost/performance configuration for a support of a known media service workload.

Traditionally, network bandwidth or disk system throughput has been the target of optimizations and sizing for streaming media services [10, 5, 1, 6, 8, 7]. Most of designed models deal with a complexity of real-time delivery of the variable bit rate content. We assume that files are encoded at constant bit rates (that is typical for commercial media sites). Our work addresses another critical system resource that has not received much attention in the past: the impact of main memory on media system performance. A typical media workload has a high degree of reference locality [2] (i.e. a high percentage of requests are accessing a small subset of media files) and exhibits a strong sharing pattern (i.e. accesses to the same file come in "bursts"). Intuitively, it means that a media server may serve most of accesses to popular files from memory, even when a media server relies on a traditional file system support and does not have additional application level caching. Thus, an efficient memory support should be considered by service providers as an additional critical parameter in choosing the best targeted configuration.

Among the *prerequisites* to a design of a capacity planning tool is the ability to measure and to compare the capacities of different media servers. In our recent work [3], we proposed a set of benchmarks for measuring the basic capacities of streaming media systems. The benchmarks allow one to derive the scaling rules of server capacity for delivering media files which are: *i)* encoded at different bit rates and *ii)* streamed from memory versus disk. Our measurements for RealServer 8.0 from RealNetworks[9] demonstrate that the amount of bandwidth a server is capable of delivering is variable, and depends on the encoding bit rates of current streams in progress. Modern media servers, such as RealServer 8.0, relies the native operating system's file buffer cache support to achieve higher application throughput when accessed files are streamed from memory. Our measurement results show that media server throughput is significantly higher (3-5 times) [1] when media streams are delivered from memory versus from disk.

The capacity planning framework is shown in Figure 1. In order to derive the media site workload profile, which can be used by our *Capacity Planner*, we designed and implemented a new tool, called *MediaProf*, which characterizes the site access profile and its system resource usage in both a quantitative and qualitative way. The *MediaProf*'s analysis is entirely based on media server access logs (they can be from one or multiple servers in a cluster).

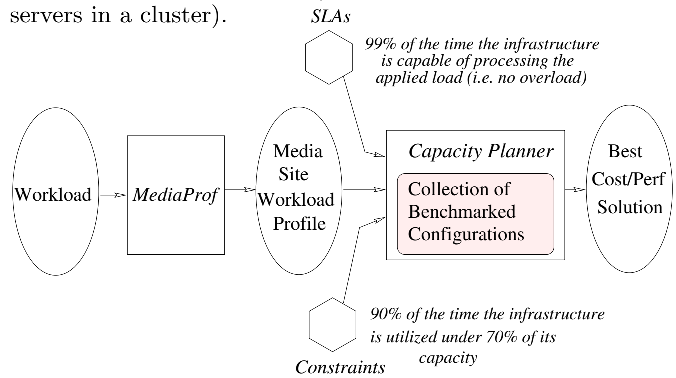

Figure 1: Capacity Planning Process.

Based on the past media workload, the *Capacity Planner* can find the best *cost/performance* solution from the existing set of

[1] These results are sensitive to a disk/file subsystem used in the media server configuration.

MM'03, November 2–8, 2003, Berkeley, California, USA.

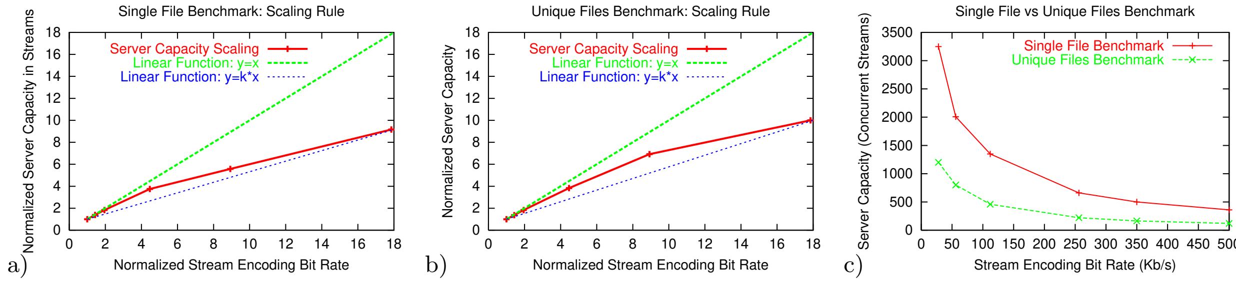

Figure 2: a) Server Capacity Scaling Rules under *Single File Benchmark*, **b) Server Capacity Scaling Rules under** *Unique File Benchmark*, **c) Comparison of Server Capacity:** *Single File vs Unique Files Benchmark*.

benchmarked configurations. Since workload measurements of existing media services indicate that client demands are highly variable (the "peak-to-mean" ratio may be the order of magnitude), it might not be economical to overprovision the future system using the past "peak" demand. Thus one can specify the service level requirements (denoted as *SLAs* in Figure 1), which would mean: "Based on the past workload history, find the best cost/performance solution that 99% of the time is capable of processing the applied (past) load". Additionally, a service provider may wish to get a configuration with planned "spare" capacity for future growth (denoted as *Constraints* in Figure 1), which would mean: "Based on the past workload history, find the best cost/performance solution that 90% of the time is utilized under 70% of its available capacity."

The proposed framework provides a flexible and convenient mapping of a service demand (client requests) into the corresponding system resource requirements necessary for accurate capacity planning.

2. BENCHMARKING MEDIA SERVER CAPACITY AND CAPACITY EQUATIONS

Commercial media servers are characterized by the number of concurrent streams a server can support [9] without loosing a stream quality, i.e. until the real-time constraint of each stream can be met. Typically, vendors measure a maximum number of concurrent streams delivered by the server when all the clients are accessing the same file encoded at a certain bit rate, e.g. 500 Kb/s. However, a multimedia content is typically encoded at different bit rates depending on a type of content and a targeted population of clients and their connection bandwidth to the Internet. In paper [3], we introduce two basic benchmarks that can establish the scaling rules for server capacity when multiple media streams are encoded at different bit rates:

- *Single File Benchmark* measuring a media server capacity when all the clients in the test are accessing the same file;
- *Unique Files Benchmark* measuring a media server capacity when each client in the test is accessing a different file.

Each of these benchmarks consists of a set of sub-benchmarks with media content encoded at a different bit rate (in our performance study, we used six bit rates representing the typical Internet audience: 28 Kb/s, 56 Kb/s, 112 Kb/s, 256 Kb/s, 350 Kb/s, and 500 Kb/s. Clearly, the set of benchmarked encoding bit rates can be customized according to targeted workload profile). Using an experimental testbed and a proposed set of basic benchmarks, we measured capacity and scaling rules of a media server running RealServer 8.0 from RealNetworks. The configuration and the system parameters of our experimental setup are specially chosen to avoid some trivial bottlenecks when delivering multimedia applications such as limiting I/O bandwidth between the server and the storage system, or limiting network bandwidth between the server and the clients.

Figure 2 a) shows the normalized graph reflecting the scaling rules for the media server capacity under the *Single File Benchmark* and different encoding bit rates. In this figure, point (1,1) presents the maximum capacity achievable by a server when all the clients in the test are accessing the same file encoded at a 500 Kb/s bit rate. Each absolute value for the other encoding bit rates is normalized with respect to it. Figure 2 b) shows the similar normalized graph for the *Unique File Benchmark*.

The measurement results show that the scaling rules for server capacity when multiple media streams are encoded at different bit rates are non-linear. For example, the difference between the highest and lowest bit rates of media streams used in our experiments is 18 times. However, the difference in the maximum number of concurrent streams a server is capable of supporting for corresponding bit rates is only around 9 times for the *Single File Benchmark* as shown in Figure 2 a), and 10 times for the *Unique Files Benchmark* as shown in Figure 2 b). The media server performance is 2.5-3 times higher under the *Single File Benchmark* than under the *Unique Files Benchmark* as shown in Figure 2 c) . This quantifies the performance benefits for multimedia applications when media streams are delivered from memory.

Using a set of basic benchmark measurements, we derive a cost function which defines a *fraction* of system resources needed to support a particular media stream depending on the stream bit rate and type of access (memory file access or disk file access):

- $cost^{disk}_{X_i}$ - a value of cost function for a stream with disk access to a file encoded at X_i Kb/s. If we define the media server capacity being equal to 1, the cost function is computed as $cost^{disk}_{X_i} = 1/N^{Unique}_{X_i}$, where $N^{unique}_{X_i}$ - the maximum measured server capacity in concurrent streams under the *Unique File Benchmark* for a file encoded at X_i Kb/s,
- $cost^{memory}_{X_i}$ - a value of cost function for a stream with memory access to a file encoded at X_i Kb/s. Let $N^{single}_{X_i}$ - the maximum measured server capacity in concurrent streams under the *Single File Benchmark* for a file encoded at X_i Kb/s. Then the cost function is computed as $cost^{memory}_{X_i} = (N^{unique}_{X_i} - 1)/(N^{unique}_{X_i} \times (N^{single}_{X_i} - 1))$. For more details, we refer the reader to [4].

Let W be the current workload processed by a media server, where

- $X_w = X_1, ...X_{k_w}$ - a set of distinct encoding bit rates of the files appearing in W ($X_w \subseteq X$),
- $N^{memory}_{X_{w_i}}$- a number of streams having a memory access type for a subset of files encoded at X_{w_i} Kb/s,
- $N^{disk}_{X_{w_i}}$ - a number of streams having a disk access type for a subset of files encoded at X_{w_i} Kb/s.

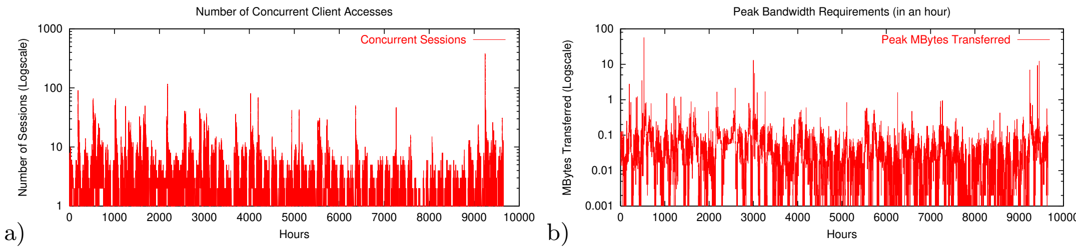

Figure 3: a) Number of Concurrent Client Sessions over Time, b) Peak MBytes Transferred over Time.

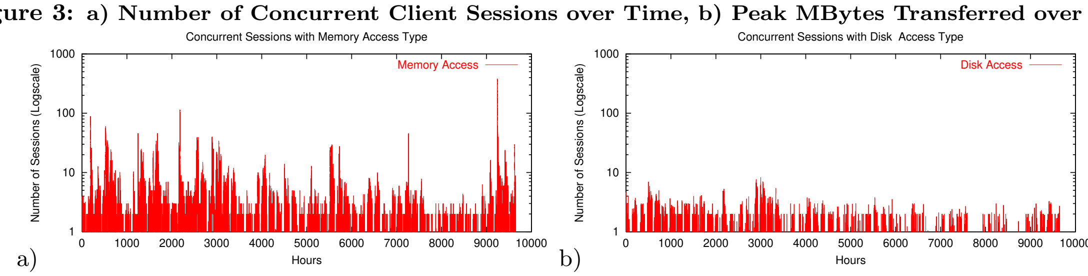

Figure 4: Concurrent Client Sessions from Memory vs Disk over Time (Bit Rate Group 112-256 Kb/s, $Size^{mem} = 1\ GB$.)

Then the service demand $Demand$ to a media server under workload W can be computed by the following capacity equation

$$Demand = \sum_{i=1}^{k_w} N^{memory}_{X_{w_i}} \times cost^{memory}_{X_{w_i}} + \sum_{i=1}^{k_w} N^{disk}_{X_{w_i}} \times cost^{disk}_{X_{w_i}} \quad (1)$$

If $Demand \leq 1$ then the media server operates within its capacity, and the difference $1 - Demand$ defines the amount of available server capacity.

The same reasoning applies to the whole media site: using a media site traffic profile, we can compute $Demand$ the site needs to support using the cost functions of different media server configurations, and then compare the computed results. For example, for a server configuration of *type1* and the corresponding cost functions, the computed service demand is $Demand = 4.5$, i.e. considered media traffic requires 5 nodes of *type1* for its support, and for another server configuration of *type2* and its corresponding cost functions the computed service demand is $Demand = 6.3$, i.e. it would require 7 nodes of *type2* to support the media site traffic.

Introduced *cost* function uses a single value to reflect the combined resource requirement such as CPU, bandwidth and memory necessary to support a particular media stream depending on the stream bit rate and type of the file access (memory or disk access). The proposed framework provides a convenient mapping of a service demand (client requests) into the corresponding system resource requirements.

3. WORKLOAD PROFILER MEDIAPROF

Media site profile, based on the past workload history, is a critical component in decision making about the future supporting infrastructure.

In the past, we have developed a tool, called *MediaMetrics* [2], for detailed media workload analysis, which is based on the corresponding media server logs. While it provides many useful insights into the corresponding client access patterns, this tool does not address the capacity planning issues. Currently, we designed and implemented a different version of the media workload analysis tool, called *MediaProf*, which reflects the access traffic profile for capacity planning goals.

While it is always useful to understand how much traffic is serviced by the site in a particular time interval (e.g per hour), this knowledge does not translate into requirements for a necessary supporting infrastructure. For capacity planing goals, the knowledge about the number of *simultaneous (concurrent)* connections and the corresponding *peak bandwidth requirements* is important. To demonstrate some of the introduced concepts, we use access log from one of the media servers supporting HP Corporate Media Site. The log covers almost one year duration. Figures 3 a),b) show the number of concurrent client sessions and the maximum bandwidth requirements to the site over the considered workload duration. (Note that Y axes use a logscale for both figures). These numbers are typically significantly lower than the corresponding numbers of aggregate traffic per hour.

Since the amount of system resources needed to support a particular client request depends on the file encoding bit rate as well the access type of the corresponding request, (i.e. different requests have a different "cost" as described in Section 2), *MediaProf* provides a corresponding classification of simultaneous connections.

In particular, for a given memory size, [2] *MediaProf* classifies whether the client request will be streaming data from memory or will be accessing data from disk. Note, that memory access does not assume or require that the whole file resides in memory: if there is a sequence of accesses to the same file, issued closely to each other on a time scale, then the first access may read a file from disk, while the subsequent requests may be accessing the corresponding file prefix from memory.

The basic idea of computing the request access type is as follows. Let $Size^{mem}$ be the size of memory in bytes. For each request r in the media server access log, we have the information about the media file requested by r, the duration of r in seconds, the encoding bit rate of the media file requested by r, the time t when a stream corresponding to request r is started (we use $r(t)$ to reflect it), and the time when a stream initiated by request r is terminated.

Let $r_1(t_1), r_2(t_2), ..., r_k(t_k)$ be a recorded sequence of requests to a media server. Given the current time T and request $r(T)$ to media file f, we compute some past time T^{mem} such that the sum of the bytes stored in memory between T^{mem} and T is equal to $Size^{mem}$. This way, the files' segments streamed by the media server between times T^{mem} and T will be in memory. In such a way, we can identify whether request r will stream file f (or some portion of it) from memory.

[2] Here, a memory size means an estimate of what the system may use for a file buffer cache.

Figure 4 shows the classification of client requests into memory accesses and disk accesses for client requests with encoding bit rates of 112-256 Kb/s, and a memory size of 1 GB. The results show that very large fraction of requests in this bit rate group can be served from memory. In particular, practically all the traffic bursts (spikes) can be served from memory as Figure 4 a) shows. Since a media server capacity is 3-5 times higher when media streams are delivered from memory versus from disk, such a qualitative media traffic classification and analysis will directly translate in significant configuration savings.

4. CAPACITY PLANNING TOOL

The overall capacity planning process is shown in Figure 1. Based on the known media site workload, *MediaProf* computes a media site workload profile. A media site workload profile reflects the number of concurrent sessions over time. These concurrent sessions are classified in the groups with defined encoding bit rates. Inside the groups, they are further partitioned according to the file access type: from memory versus from disk. Since the classification according to the access type depends on memory size, the resulting workload profile is a set of media site profiles computed for a particular memory size.

The next module, the *Capacity Planner*, has a collection of benchmarked configurations with the corresponding cost functions for different type of requests. The *Capacity Planner* takes the media site workload profile (for a particular memory size) and using the cost functions of a particular media server computes the corresponding service demand profile over time according to formula (1) from Section 2. The service demand profile is computed for different memory sizes and different benchmarked configurations. If a service provider is interested to find the sizing recommendations for a subset of media servers in the database with particular memory sizes, he/she can specify them via a tool configuration file. This will narrow the search space.

Computed *service demand profile* is the list of pairs. The first element of a pair represents a time duration, e.g. 300 sec. The second element reflects the service demand over this time duration in , e.g. 4.5 means that the considered media site workload requires 4.5 servers during 300 sec, or for example, 0.6 means that the workload requires one server for 300 sec, and the server will be utilized during this time at 60%.

The service demand profile is ordered by the second element. Pairs with the same second component are aggregated. Thus, the top element in the service demand profile represents the peak load demands for the considered media site, as well as the corresponding time duration for these peak loads over time. Since workload measurements of existing media services indicate that client demands are highly variable (the "peak-to-mean" ratio may be the order of magnitude), it might not be economical to overprovision the future system using the past "peak" demand. In this case, one can specify the service level requirements (denoted as *SLAs* in Figure 1), which would mean: "Based on the past workload history, find the best cost/performance solution that 99% of the time is capable of processing the applied (past) load". Using the computed service demand profile, the *Capacity Planner* finds the maximum load requirements corresponding to the 99-th percentile of all the media site service demands over time. Let us denote this service demand as $Demand_{SLA}$.

Additionally, a service provider may wish to define a configuration with planned "spare" capacity for future growth (denoted as *Constraints* in Figure 1), which would mean: "Based on the past workload history, find the best cost/performance solution that 90% of the time is utilized under 70% of its available capacity." Thus using the computed service demand profile, the *Capacity Planner* finds the maximum load requirements corresponding to the 90-th percentile of all the media site service demands over time. For example, if the service demand corresponding to the 90-th percentile is 3.5 then the requirements to configuration utilized under 70% of its available capacity will be 3.5/0.7=5. Let us denote this service demand as $Demand_{Constraints}$.

The requirement for a desirable configuration is:

$$Demand_{overall} = max(Demand_{SLA}, Demand_{Constraints})$$

rounded up to the closest integer.

There could be multiple configurations satisfying the specified performance requirements. Taking into consideration the price information of the corresponding configurations, the best cost/performance solution can be chosen.

5. CONCLUSION AND FUTURE WORK

In this paper, we outlined a new capacity planning approach for a streaming media service using its past workload. The proposed approach provides a flexible and convenient mapping of a service demand (client requests) into the corresponding system resource requirements necessary for accurate capacity planning.

In summary, our capacity planning tool is promoting a new unified framework for:

- measuring media server capacity via a set of basic benchmarks;
- deriving the resource requirements (a *cost*) of a particular stream from the basic benchmark measurements;
- estimating the service capacity demands from the proposed media site workload profile and the corresponding cost functions of the benchmarked configurations.

In the future, we plan to incorporate in our tool a set of load balancing and content placement strategies for media clusters.

6. REFERENCES

[1] E. Biersack, F. Thiesse. Statistical Admission Control in Video Servers with Constant Data Length Retrieval of VBR Streams. Proc. of the 3d Intl. Conference on Multimedia Modeling, France, 1996.

[2] L. Cherkasova, M. Gupta. Characterizing Locality, Evolution, and Life Span of Accesses in Enterprise Media Server Workloads. Proc. of ACM NOSSDAV, 2002.

[3] L. Cherkasova, L. Staley. Measuring the Capacity of a Streaming Media Server in a Utility Data Center Environment. Proc. of 10th ACM Multimedia, 2002.

[4] L. Cherkasova, L. Staley. Building a Performance Model of Streaming Media Applications in Utility Data Center Environment. Proc. of ACM/IEEE Conference on Cluster Computing and the Grid (CCGrid), May, 2003.

[5] J. Dengler, C. Bernhardt, E. Biersack. Deterministic Admission Control Strategies in Video Servers with Variable Bit Rate. Proc. of European Workshop on Interactive Distributed Multimedia Systems and Services (IDMS), Germany, 1996.

[6] A. Dan, D. Dias R. Mukherjee, D. Sitaram, R. Tewari. Buffering and Caching in Large-Scale Video Servers. Proc. of COMPCON, 1995.

[7] Y. Fu, A. Vahdat. SLA-Based Distributed Resource Allocation for Streaming Hosting Systems. Proc. of the 7th Intl Workshop on Web Content Caching and Distribution (WCW-7), 2002.

[8] M. Kamath, K. Ramamritham, D. Towsley. Continuous Media Sharing in Multimedia Database Systems. Proc. of the 4th Intl. Conference on Database Systems for Advanced Applications, 1995.

[9] Realsystem Server Product Line Comparison. http://www.realnetworks.com/products/servers/comparison.html

[10] H. Vin, P. Goyal, A. Goyal, A. Goyal. A Statistical Admission Control Algorithm for Multimedia Servers. In Proc. of ACM Multimedia, San Francisco, 1994.