Near Optimal Number of Replicas for Continuous Media in Ad-hoc Networks of Wireless Devices

Ashkan Aazami, Shahram Ghandeharizadeh, Tooraj Helmi
Department of Computer Science
University of Southern California
Los Angeles, CA 90089, USA

ABSTRACT

This study investigates replication of data in a novel streaming architecture consisting of ad-hoc networks of wireless devices. One application of these devices is home-to-home (H2O) entertainment systems where a device collaborates with others to provide each household with on-demand access to a large selection of audio and video clips. These devices are configured with a substantial amount of storage and may cache several clips for future use. A contribution of this study is a technique to compute the number of replicas for a clip based on the square-root of the product of bandwidth required to display clips (β_i) and their frequency of access (f_i), i.e., $(\beta_i \times f_i)^{\alpha}$ where $\alpha = 0.5$. We provide a proof to show this strategy is near optimal when the objective is to maximize the number of simultaneous displays in the system with string and grid (both symmetric and asymmetric) topologies. We say "near optimal" because values of α less than 0.5 may be more optimum. In addition, we use analytical and simulation studies to demonstrate its superiority when compared with other alternatives. A second contribution is an analytical model to estimate the theoretical upper bound on the number of simultaneous displays supported by an arbitrary grid topology of H2O devices. This analytical model is useful during capacity planning because it estimates the capabilities of a H2O configuration by considering: the size of an underlying repository, the number of nodes in a H2O cloud, the representative grid topology for this cloud, and the expected available network bandwidth and storage capacity of each device. It shows that one may control the ratio of repository size to the storage capacity of participating nodes in order to enhance system performance. We validate this analytical model with a simulation study and quantify its tradeoffs.

1. INTRODUCTION

Recent advances in communication and processing have made ad-hoc networks of wireless devices a reality. Intel, for example, offers a small device that consists of a 500 MHz processor and a wireless component that operates in the 5 GHz spectrum, offering transmission rates in the order of tens of Megabits per second, Mbps. This device costs approximately $85 and can be extended with a mass storage device. One application of these devices is in support of home entertainment systems where multiple home-to-home (H2O) devices [10] collaborate to provide on-demand programming to a household. A cellular base station might serve as its interface to a wired infrastructure such as Internet (for services such as billing and permanent storage of data). Wireless H2O clouds complement the world-wide-web and peer-to-peer networks, e.g., Kazaa, Gnutella [6], etc., by increasing availability of data at households. Moreover, a household may store its personal video library on a H2O cloud for retrieval anywhere, e.g., at a friend's home. The system might encrypt the content to either protect it from un-authorized access, i.e., authentication, or implement a business model for generating revenues. This flexible access to data is a building component of complex systems such as Memex [3] and MyLifeBits [8].

One challenge of a H2O cloud is to display continuous media, audio and video clips. Continuous media consists of a sequence of quanta, either audio samples or video frames, that convey meaning when presented at a pre-specified rate [9, 13]. Once the display is initiated, if the data is delivered below this rate then the display might suffer from frequent disruptions and delays. Each H2O device may store clips in anticipation of future reference either by its household or a neighboring household. Ideally, when a household references a clip using a H2O device, this device would find the clip in its local storage. Otherwise, this H2O device (termed $H2O_d$) must first locate a H2O device containing the referenced clip (termed $H2O_p$). Next, it must admit itself into the system through a process that reserves a path from $H2O_p$ to $H2O_d$ in order to stream the referenced clip at a pre-specified bandwidth. This process is typically termed admission control. It involves collaboration of other H2O devices serving as intermediate application-based routers by requiring them to reserve their networks bandwidth on behalf of this stream. By reserving bandwidth along a path, $H2O_d$ may overlap its display with delivery of data from $H2O_p$, minimizing the latency observed by a user. Thus, a H2O device might serve either as a producer of data, a data router, an active display, or a combination of these simultaneously.

Assuming the mass storage bandwidth exceeds the wireless network bandwidth, overall performance is enhanced when the length of the path that offers sufficient bandwidth to stream data from $H2O_p$ to $H2O_d$ is minimized. When the length of this path is zero, $H2O_d$ finds its referenced clip resident in its local storage. In order to approximate this ideal scenario, the system must address two critical topics. First, how many replicas of a clip should be constructed in a cloud of $\mathcal{N}$ H2O devices? Second, how should these replicas be placed across

.

the H2O nodes? The focus of this study is on the first topic. We make the following simplifying assumptions in order to address this topic. First, a H2O device displays one stream. Second, it is configured with hundreds of Gigabytes (GB) of mass storage and may cache tens of clips on its local storage. (A 2 hour DVD quality video is typically 3.5 GB in size.) Third, we assume a centralized strategy which periodically gathers all the relevant profiles such as frequency of access and decides the number of replicas for each clip. This replication strategy might be deployed at a base station responsible for a fixed geographical area containing $H2O$ devices.

The primary contribution of this paper is a near optimal technique to compute the number of replicas for a clip i proportional to the square root of the product of its bandwidth (β_i) and frequency of access (f_i): $\sqrt{\beta_i.f_i}$. Using analytical and simulation studies, we demonstrate this technique maximizes the number of simultaneous displays supported by a grid topology. The analytical models are a secondary contribution of this study. They are flexible enough to approximate the bandwidth required to display clip k with r_k replicas for both symmetric (square-shaped) and asymmetric (rectangular-shaped) grids. In the worst case scenario, an asymmetric grid might represent a string topology. The analytical models show the maximum number of simultaneous displays supported by a H2O cloud is dependent on the size of the repository and available storage capacity of the system. These models can be used during capacity planning in order to meet a pre-specified performance objective.

Our proposed replication technique complements web cache servers and their employed techniques to manage static content, i.e., audio and video clips. Similar to our proposed replication strategy, cache servers strive to bring a copy of data closer to a potential consuming client. One may conceptualize each H2O device as both a potential consuming client and a cache server for its neighboring households. Web cache replacement policies such as GreedyDual [15], and others [2, 1, 4] capitalize on the short-term temporal correlations in web request patterns [18, 16]. These might be extended to manage storage of H2O devices in a decentralized manner. Our replication strategy is centralized with advanced knowledge of the available storage and the popularity of continuous media clips to be replicated across devices. Design of decentralized techniques is a future research direction. The proposed centralized technique is near optimal and can be used as a yard stick to evaluate a decentralized technique.

In an early study, we evaluated alternative replication techniques for $\mathcal{N}$ H2O devices organized in a string topology, see [12]. Its primary contribution was to show that a technique based on Webster's Monotone[1] Divisor Method [19] is inferior to a technique employing the square-root of the frequency of access. i.e., a technique employing $\beta_i.f_i$ is inferior to $\beta_i.\sqrt{f_i}$. We extend this early study to consider a technique based on $(\beta_i.f_i)^\alpha$. We show this technique to provide either comparable performance or outperforms a technique based on $\beta_i.\sqrt{f_i}$ when $\alpha = 0.5$, see Section 4. Moreover, the analytical models presented here are a generalization of those described in [12] because they model both an arbitrary shaped grid and a string topologies.

The rest of this paper is organized as follows. In Section 2, we provide a formal statement of the replication problem. Section 3 provides an analytical model to estimate the bandwidth required to stream a clip with r replicas. Section 4 uses these models to compare the proposed replication technique with other alternatives and a random strategy that decides the number of replicas for each clip using a random number generator. Obtained results demonstrate that some of the alternatives to our proposed replication strategy are as blind as using a random number generator. In Section 5, we validate our analytical models using a simulation study. Brief conclusions and future research directions are contained in Section 7.

2. CLIP REPLICATION

Database Parameters	
m	Number of clips
s_k	Size of clip k
d_k	Display time of clip k
β_k	Bandwidth required to display clip k
f_k	Frequency of access to clip k
r_k	Number of replicas for clip k
System Parameters	
$\mathcal{N}$	Number of nodes in $M+1$ by $N+1$ grid, $\mathcal{N} = (M+1)(N+1)$
M	A grid has M+1 columns numbered from 0 to M
N	A grid has N+1 rows numbered from 0 to N
$c_{(i,j)}$	Storage capacity of $H2O_{(i,j)}$
C	Total storage capacity in the grid topology, $\sum_{a=0}^{M}\sum_{b=0}^{N} c_{(a,b)}$
Analytical Models	
$n_l(i,j)$	Number of neighbor with distance $\leq l$ from $H2O_{(i,j)}$
$\widetilde{n}_l(i,j)$	Approximation formula for number of neighbor with distance $\leq l$ from $H2O_{(i,j)}$
$D_k(i,j)$	Distance to the nearest copy of clip k from $H2O_{(i,j)}$
$\widetilde{D}_k$	Average distance to the nearest copy of clip k

Table 1: Terms and their definitions

Assume a $M+1$ by $N+1$ grid, i.e. a grid of $M+1$ columns and $N+1$ rows, consisting of $\mathcal{N} = (M+1).(N+1)$ H2O devices. There is a H2O device at the intersection of each row and column. We identify each device with its X and Y coordinates, H2O$_{(i,j)}$ where $0 \leq i \leq M$ and $0 \leq j \leq N$. Two devices, say $H2O_{(i,j)}$ and $H2O_{(r,s)}$, may communicate if their Manhattan distance equals one, i.e. $|i-r| + |j-s| = 1$, in other words, they are in the same radio range when $\sqrt{(i-r)^2+(j-s)^2} < \sqrt{2}$. $H2O_{(i,j)}$ has storage capacity $c_{(i,j)}$ bytes. The total storage capacity of the system is C=$\sum_{i=0}^{M}\sum_{j=0}^{N} c_{(i,j)}$. There are m clips in the database, each with an average bandwidth requirement of β_k and size s_k. These two parameters specify the display time of a clip, d_k=$\frac{s_k}{\beta_k}$. The frequency of access to clip k is denoted as f_k with $\sum_{k=1}^{m} f_k = 1$. The number of replicas for clip k is denoted as r_k where $1 \leq r_k \leq \mathcal{N}$ (r_k includes the original copy of clip k). A replication strategy determines a r_k value for each clip k with the objective to ensure $\mathcal{N}$ H2O devices can display a clip simultaneously[2]. We assume the total size of the database ex-

[1]Webster's Method and its alternatives due to Hamilton, Adams and Jefferson allocate storage to each replica (assign seats to a party) as a function of its popularity (ratio of votes casted for a party to the total vote). See [14] for details.

[2]A placement strategy assigns a replica of clip k to a specific

ceeds the storage capacity of a H2O device, $\sum_{k=1}^{m} s_k > c_{(i,j)}$ for $0 \leq i \leq N, 0 \leq j \leq M$. Otherwise, the problem is trivial and the database should be replicated on each device in its entirety. Similarly, we assume there is at least one copy of a clip in the system and that the database size is smaller than the total storage capacity of the system, $\sum_{k=1}^{m} s_k \leq C$. Otherwise, clips cannot be replicated due to insufficient storage capacity. In sum, a replication strategy must construct at least one copy of each clip in the system and no more than $\mathcal{N}$ replicas, $1 \leq r_k \leq \mathcal{N}$. Also the total database size after replication may not exceed the total capacity of the network:

$$\sum_{k=1}^{m} r_k.s_k \leq C \tag{1}$$

2.1 Near Optimal Number of Replicas

The goal is to compute the number of replicas for each clip such that the system can simultaneously support the maximum number of displays. In other words we want to minimize the expected total bandwidth required to support simultaneous displays. The last statement is formulated as follows:

$$\min_{\sum r_j s_j = C} \left\{ \sum_{i=1}^{m} f_i \beta_i \widetilde{D}_i \right\} \tag{2}$$

where the size and bandwidth requirement of clip i, are denoted by s_i and β_i, respectively, and f_i is the frequency of access to the clips. $\widetilde{D}_i$ is the average distance to the nearest copy of clip i in the grid (see the discussions of Equation 7). Minimization of Equation 2 is subject to the constraint of Equation 1. In the following we prove that replication of clips proportional to $\sqrt{\beta_i.f_i}$ is near optimal because it minimizes Equation 2. A strategy to replicate clips accordingly is as follows. First, it employs the available storage capacity of the system to approximate the number of replicas for each clip k:

$$r_k = \frac{C}{\sum_{i=1}^{m} s_i (\beta_i.f_i)^{\alpha}} \times (\beta_k.f_k)^{\alpha}, \ where\ \alpha = 0.5 \tag{3}$$

If r_k is less than 1 then it is reset to one. If it exceeds $\mathcal{N}$ then it is reset to $\mathcal{N}$. Once r_k is computed for all clips, the total storage requirement of the database might be either less than or greater than C. Consider each case in turn. If the final storage requirement is less than C then it computes the remaining idle storage C_{idle}. Next, it removes those clips with $\mathcal{N}$ replicas (changes the value of m) and re-applies the replication strategy by using C_{idle} instead of C. If the final storage requirement exceeds C, it decrements the value of r_k starting with the clip that has the maximum r_k value. When several clips have identical r_k values, it chooses the clip with the largest size. If several clips qualify then one is chosen randomly.

In order to simplify discussions and without loss of generality, we first prove this for a string topology. Subsequently, we extend this to a symmetric grid topology. Each proof assumes a homographic function to model $\widetilde{D}_i$. This assumption is not realistic; However, it enables us to prove the optimality of our claim for $\alpha = 0.5$. Equation 7 derives the correct value of $\widetilde{D}_i$ and has no closed form. Sections 4 to 7 present several instances based on analytical and simulation models to substantiate our claim of near-optimality with $\alpha = 0.5$. Of course other values of α, say $\alpha = 0.3\overline{3}$, might provide superior performance.

node. An investigation of these strategies is beyond the focus of this study.

THEOREM 1. *With a $\mathcal{N}$ by 1 grid, i.e., a string topology, a technique that computes the number of replicas for each clip i proportional to its $\sqrt{\beta_i.f_i}$ is near optimal.*

PROOF. First step is to compute the average distance to the nearest copy of clip i, $\widetilde{D}_i$, for each $1 \leq i \leq m$, see Equation 7. We provide an approximation which can be represented in a closed form. To motivate this approximation, consider different r_i values: 1,0,$\mathcal{N}$,and $\mathcal{N}-1$. With a string topology and just one replica of clip i, an optimal placement strategy places this replica in the middle of the string. In this case, there are 2 paths of each length l, where $0 < l \leq \frac{\mathcal{N}-1}{2}$ and just one path of length 0. So the average length of the path between the client and the replica will be:

$$\widetilde{D}_i = \frac{\sum_{l=0}^{\frac{\mathcal{N}-1}{2}} 2l}{2 \times \frac{\mathcal{N}-1}{2} + 1} = \frac{\mathcal{N}-1}{4} \ when\ r_i = 1$$

When there are no replicas of a clip in the network (r_i=0), $\widetilde{D}_i$ is infinite. At the other extreme, with $\mathcal{N}$ replicas for clip i then $\widetilde{D}_i = 0$. With $\mathcal{N}-1$ replicas for clip i, $\widetilde{D}_i = \frac{1}{\mathcal{N}}$ because there are $\mathcal{N}-1$ paths of length 0 and one path of length 1. Thus, we use a homographic function to approximate the average distance between a client and a replica of its referenced clip, $\widetilde{D}_i = \frac{\mathcal{N}-r_i}{4r_i}$. By using this function in our minimization problem, we obtain:

$$\min_{\sum r_j s_j = C} \left\{ F = \sum_{i=1}^{m} f_i \beta_i \frac{\mathcal{N}-r_i}{4r_i} \right\}$$

In order to solve the above minimization problem we take the derivative of B with respect to r_i's: $\frac{\partial F}{\partial r_1} = \frac{\partial F}{\partial r_2} = ... = \frac{\partial F}{\partial r_m} = 0$. By considering the first equality we obtain:

$$\begin{aligned} 0 &= \frac{\partial F}{\partial r_1} \\ &= \frac{\partial}{\partial r_1} \left(f_1 \beta_1 \frac{\mathcal{N}-r_1}{4r_1} + \sum_{i=2}^{m-1} f_i \beta_i \frac{\mathcal{N}-r_i}{4r_i} \right. \\ &\quad \left. + f_m \beta_m \frac{\mathcal{N}.s_m}{S - \sum_{i=1}^{m-1} f_i \beta_i \frac{\mathcal{N}-r_i}{4r_i}} - \frac{1}{4} \right) \\ &= -f_1 \beta_1 \frac{\mathcal{N}}{4{r_1}^2} + \frac{\mathcal{N}.s_m}{\left(S - \sum_{i=1}^{m-1} f_i \beta_i \frac{\mathcal{N}-r_i}{4r_i} \right)^2} \\ \Rightarrow \quad & f_1 \beta_1 \frac{\mathcal{N}}{4{r_1}^2} = \frac{\mathcal{N}.s_m}{\left(S - \sum_{i=1}^{m-1} f_i \beta_i \frac{\mathcal{N}-r_i}{4r_i} \right)^2} \end{aligned} \tag{4}$$

Similarly for the second equality we have:

$$f_2 \beta_2 \frac{\mathcal{N}}{4{r_2}^2} = \frac{\mathcal{N}.s_m}{\left(S - \sum_{i=1}^{m-1} f_i \beta_i \frac{\mathcal{N}-r_i}{4r_i} \right)^2} \tag{5}$$

By combining Equations 4 and 5, we obtain: $\frac{r_1}{r_2} = \sqrt{\frac{f_1 \beta_1}{f_2 \beta_2}}$. Similarly for any i and j it can be shown that: $\frac{r_i}{r_j} = \sqrt{\frac{f_i \beta_i}{f_j \beta_j}}$ □

THEOREM 2. *In a $N+1$ by $N+1$ grid network, $\sqrt{\beta_i.f_i}$ computes the near optimal number of replicas for each clip.*

PROOF. Once again, consider the value of $\widetilde{D}_i$ with four different r_i values: 0, 1, $\mathcal{N}-1$, and $\mathcal{N}$. The values and discussion for r_i values 0, $\mathcal{N}-1$ and $\mathcal{N}$ are the same as the proof

for Theorem 1 (and not repeated here). It is different for 1 because of the diamond shown in Figure 1.a. With 1 replica of a clip placed in the center of a symmetric grid topology, i.e., a square, the number of paths with length l, where $l \leq \frac{\sqrt{\mathcal{N}}}{2}$, is: $\frac{\sum_{l=0}^{\frac{\sqrt{\mathcal{N}}}{2}} 4l \times l}{\sum_{l=0}^{\frac{\sqrt{\mathcal{N}}}{2}} l} = \frac{4(\sqrt{\mathcal{N}}+1)}{3}$. The following homographic function estimates the average distance between a client and a replica of a clip: $\widetilde{D}_i = \frac{4(\sqrt{\mathcal{N}}+1)(\mathcal{N}-r_i)}{3(\mathcal{N}-1)r_i}$. By repeating the approach of Theorem 1 using this expression, the near optimality of the square root technique is proven. □

3. ANALYTICAL MODEL

This section computes the amount of bandwidth required from an arbitrary grid network topology, either square or rectangular, to display clip k. This analysis consists of two steps: First, we compute the expected distance from each H2O$_{(i,j)}$ node to a replica of clip k with r_k copies, $r_k \geq 1$. This is because each H2O node might be a potential client for clip k. This estimation is based on an average and ignores the placement of k's replicas. Second, using this expected distance, we compute the expected amount of bandwidth required to display clip k. Section 4 presents an experimental model that utilizes these models to estimate the number of simultaneous displays supported by a grid consisting of $\mathcal{N}$ nodes.

To compute the expected distance from $H2O_{(i,j)}$ to the nearest replica of clip k, we count the number of neighbors for $H2O_{(i,j)}$, see the discussions of Equations 6 and 8. This is based on the number of neighbors for $H2O_{(i,j)}$ computed as follows. First, we place a diamond with $H2O_{(i,j)}$ located as its center. For example, in Figure 1a, a diamond is placed on $H2O_{(3,3)}$. This diamond may exceed the dimensions of the grid in sixteen possible scenarios. One scenario is shown in Figure 1b with $H2O_{(1,0)}$ as the center of diamond. With this and the other 15 scenarios, we employ the principal of exclusion and inclusion to accurately count the number of neighbors that are at most l hops away. Using exclusion, we count those phantom nodes that are outside of the grid, see the discussions of Equation 10. Using inclusion, we re-include those phantom nodes which are excluded twice, see the discussions of Equation 11. The closed form formula for the number of neighbors is shown in Equation 12. Section 4.3 compares this exact and somewhat complex equation with an approximation shown in Equation 9. It makes the key observation that an approximation makes an inferior technique to appear better than its true capabilities. This is specially true with asymmetric topologies, e.g., rectangular grids. The following details our analytical model.

Assume r_k replicas of the k^{th} clip and let $D_k(i,j)$ be the distance from node $H2O_{(i,j)}$ to the nearest replica of clip k in the network. Let $n_l(i,j)$ be the total number of nodes with a distance less than or equal to l from $H2O_{(i,j)}$. (We derive $n_l(i,j)$ in the following paragraphs.) The expected value of $D_k(i,j)$ is:

$$\mathbb{E}(D_k(i,j)) = \sum_{l \geq 0} l \times \mathbb{P}(D_k(i,j) = l) = \sum_{l \geq 0} \mathbb{P}(D_k(i,j) > l) \tag{6}$$

where $\mathbb{P}(D_k(i,j) > l)$ is the probability of finding the nearest replica of clip k in a device that is more than l hops away from $H2O_{(i,j)}$. Define $\widetilde{D_k}$ be the average of $D_k(i,j)$ across all $H2O$ devices on the network:

$$\widetilde{D_k} = \frac{1}{\mathcal{N}} \sum_{i=0}^{M} \sum_{j=0}^{N} \mathbb{E}(D_k(i,j)) \tag{7}$$

Placement of replicas is based on a random assignment of r_k replicas across $\mathcal{N}$ nodes. A placement that prevents $H2O_{(i,j)}$ from locating a replica within l hops is identical to placing the r_k replicas across $\mathcal{N} - n_l(i,j)$ nodes, i.e. excluding nodes whose distance is either less than or equal to l from $H2O_{(i,j)}$. Thus:

$$\mathbb{P}(D_k(i,j) > l) = \begin{cases} \dfrac{\binom{\mathcal{N} - n_l(i,j)}{r_k}}{\binom{\mathcal{N}}{r_k}} & r_k \leq \mathcal{N} - n_l(i,j) \\ 0 & otherwise \end{cases} \tag{8}$$

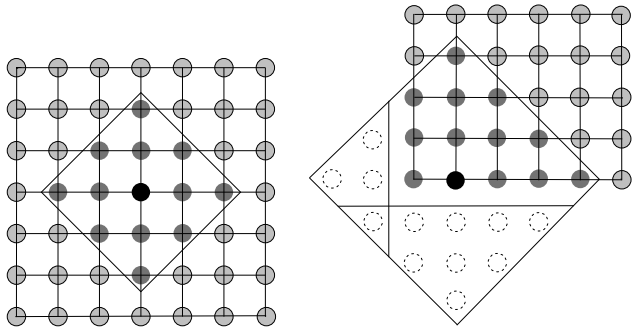
In the following we drive $n_l(i,j)$. We begin by providing an approximation $\widetilde{n}_l(i,j)$ by ignoring the edges of the grid, i.e. consider the grid as infinite in all directions. This means that $\widetilde{n}_l(i,j)$ depends only on l:

$$\widetilde{n}_l(i,j) = 2l^2 + 2l + 1 \tag{9}$$

To illustrate, in Figure 1a, $\widetilde{n}_2(3,3)$ equals 13, i.e. number of devices in the diamond. In those scenarios where the center of diamond, $H2O_{(i,j)}$ is close to the grid's edges, Equation 9 overestimates the number of neighbors. Figure 1b is an example where $\widetilde{n}_3(1,0)$ computes 25 instead of 13. In order to fix this discrepancy, 12 phantom nodes depicted with dotted lines in Figure 1b must be excluded from Equation 9. It is trivial to see that the number of phantom nodes attributed to an edge of the grid depends on the distance from the center to that edge. If this distance for an edge is larger than l then that edge does not suffer from phantom nodes. Otherwise, let $b(d,l)$ be the number of phantom nodes attributed to an edge, and d be the distance from the center to that edge. The following *boundary function* counts the number of phantom nodes:

$$b(d,l) = \begin{cases} 1 + 3 + \ldots + 2(l-d) - 1 = (l-d)^2 & d < l \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

For example, in Figure 1b, the number of phantom nodes attributed by the bottom edge is $b(0,3) = (3-0)^2 = 9$.



1a. Number of neighbors for $H2O_{(3,3)}$ with distance ≤ 2

1b.Number of neighbors for $H2O_{(1,0)}$ with distance ≤ 3

Figure 1: Number of neighbors for a specific H2O client requesting a clip

(Bottom edge is the one with Y value equal zero, top edge is the one with maximum Y value, left edge is the one with $X = 0$, and right edge is the one with maximum X value.) Assume the center of the diamond is at $H2O_{(i,j)}$ in a $M+1$ by $N+1$ grid. To compute the number of nodes with a maximum distance of l from $H2O_{(i,j)}$ we subtract the number of phantom nodes attributed to the four edges of the grid, i.e. $\widetilde{n}_l(i,j) - b(i,l) - b(j,l) - b(M-i,l) - b(N-j,l)$. Note that this formula is still incorrect because phantom nodes might be counted twice. For example, in Figure 1b, one node is excluded twice because it falls at the intersection of the phantom nodes identified for both the bottom and left edges. In order to fix this limitation, the number of such nodes must be reincluded once. The number of such nodes depends on the relative position of $H2O_{(i,j)}$ to the corners. Function $c(d_1, d_2, l)$, *corner function*, counts these nodes:

$$\begin{cases} 1 + 2 + \ldots + (l - d_1 - d_2 - 1) = \binom{l - d_1 - d_2}{2} & d_1 + d_2 < l \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where d_1, d_2 are distances to the edges which intersect at that corner. For example in Figure 1b, the number of phantom nodes counted twice is $c(1,0,3) = 1$. By combining the boundary and corner functions, we obtain the following formula, which counts the exact number of nodes at a distance less than or equal to l from $H2O_{(i,j)}$:

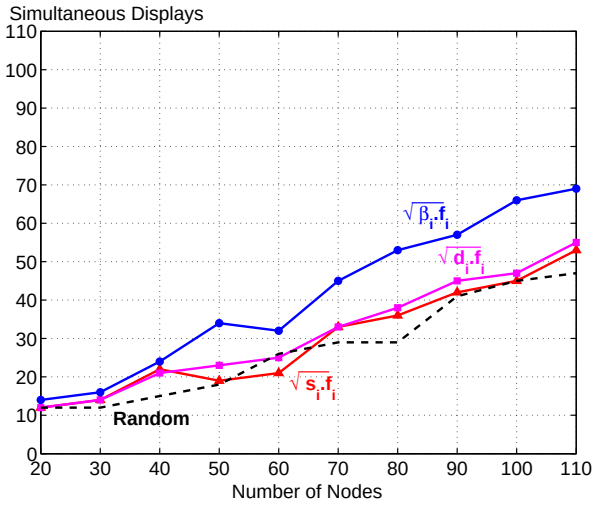
$$\begin{aligned} n_l(i,j) = \widetilde{n}_l(i,j) \quad & -b(i,l) - b(j,l) - b(M-i,l) - b(N-j,l) \\ & +c(i,j,l) + c(i,N-j,l) + c(M-i,j,l) \\ & +c(M-i,N-j,l) \end{aligned} \tag{12}$$

Equation 6 computes the expected distance to the nearest replica. To compute the expected bandwidth consumed by the network to deliver the referenced clip, we multiply the expected distance to the nearest copy from $H2O_{(i,j)}$ by the bandwidth required to display clip k: $\beta_k \times \mathbb{E}(D_k(i,j))$.

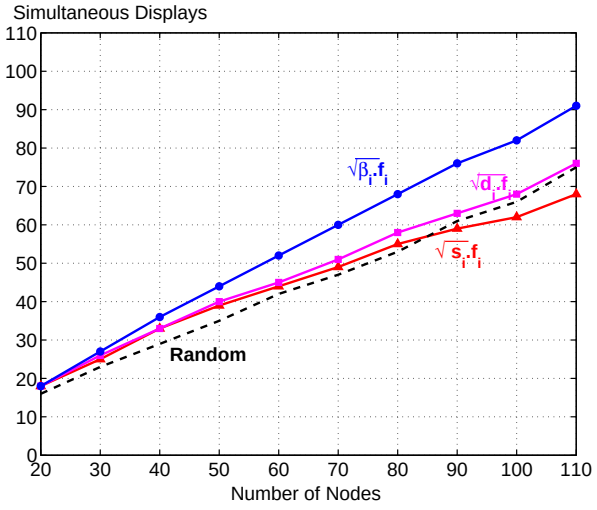
4. AN EVALUATION USING THE ANALYTICAL MODEL

In our experiments we considered two different grid types consisting of $\mathcal{N}$ nodes: 1) *Symmetric*, $N = M = \sqrt{\mathcal{N}} - 1$, and 2) *Asymmetric*, $N = 1, M = \frac{\mathcal{N}}{2} - 1$. (Recall that we assume a grid consisting of $N+1$ row and $M+1$ columns.) Each node is configured with 100 GB of storage. The wireless bandwidth between two nodes with a manhattan distance of 1 is 4 Mbps. We simulated a skewed distribution of access to the clips using a *Zipfian* distribution with a mean of 0.27. This distribution is shown to correspond to sale of movie theater tickets in the United States [7].

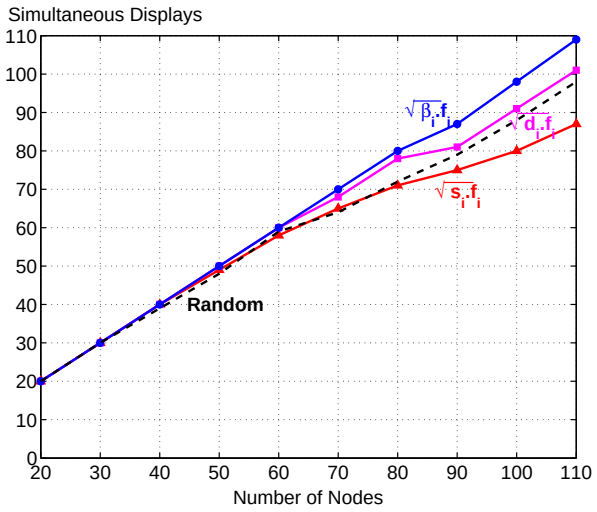
We studied the behavior of grid topologies with both a *Homogeneous* and a *Heterogeneous* repository. A homogeneous repository consists of m equi-sized clips, each requiring 4 Mbps for its continuous display and a display time of 120 minutes. A heterogeneous database consists of $\frac{m}{2}$ video clips and $\frac{m}{2}$ audio clips. While video clips continue to require 4 Mbps for their display, audio clips require 340 Kbps. The audio clips are evenly divided among three sizes: those with a display time of 2, 4, and 8 minutes. The video clips are evenly divided among four sizes: those with a display time of 30, 60, 90, and 120 minutes. We continue to use the Zipfian distribution with a mean of 0.27 across all clips. Thus, if the m clips are sorted based on their frequency of access then: 1) The video



2a. Worst of 100



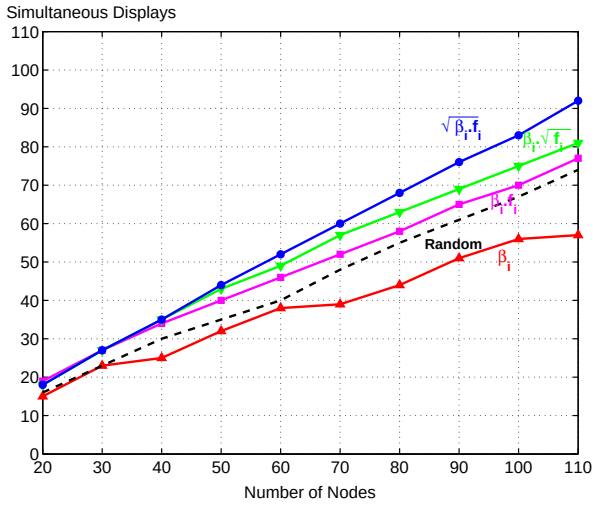
2b. Average across 100



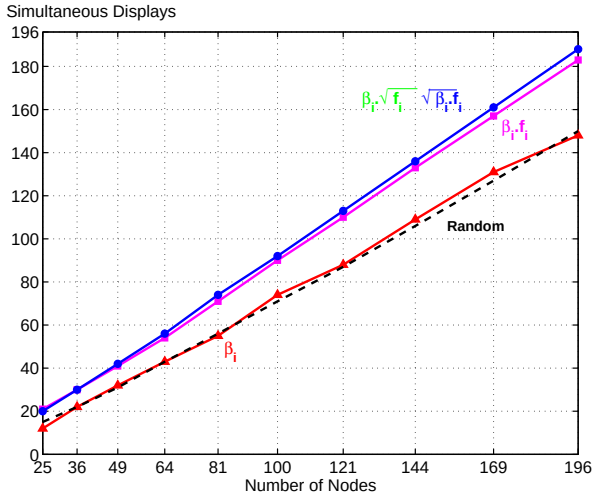
2c. Best of 100

Figure 2: Alternative replication strategies using an asymmetric topology with a heterogeneous database consisting of 1300 two hour video clips, m=1300, size = 1.4 TB.

clips are assigned to the odd positions in a round-robin manner starting with the shortest clip first: clip 1 is 30 minutes long, clip 3 is 60 minutes long, clip 5 is 90 minutes long, clip 7 is 120 minutes long, etc., and 2) The audio clips are assigned to the even positions in a round-robin manner starting with the shortest clip first: clip 2 has a display time of 2 minutes, clip 4 has a display time of 4 minutes, clip 6 has a display time of 8 minutes, etc.



3b. Homogeneous-Symmetric, $m = 400$ (1.37 TB)



3a. Heterogeneous-Asymmetric, $m = 1300$ (1.4 TB)

Figure 3: A comparison of alternative replication techniques based on bandwidth.

An experiment starts with the computation of the number of replicas for each clip using one of the strategies with the available storage of $\mathcal{N}$ H2O devices. Next, each node requests a clip using a random number generator conditioned based on the Zipfian distribution of access. We employ the analytical model of Section 3 to estimate the expected distance to find a requested clip for each node in the network. By multiplying this distance to the display bandwidth requirement of a clip we obtain the total bandwidth which is consumed in the network in order to stream and display the requested clip. The simulator schedules requests starting with a randomly chosen node. It continues this process until either all nodes have an active display or the available bandwidth is exhausted. The number of scheduled nodes is the number of simultaneous displays supported by a replication strategy. In the reported experiments, we invoked the above procedure 100 times and observed its average number of displays. Each iteration is with a different seed for the random number generator which changes the identity of a clip referenced by a node (conditioned using a Zipfian distribution).

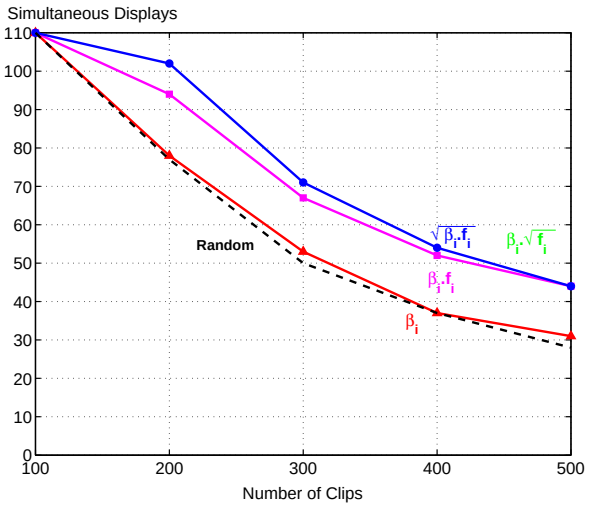
As a yardstick, we also analyzed a *Random* replication strategy that determines the number of replica for a clip randomly. Its details are as follows. This technique randomly chooses a clip and assigns it a random number of replicas. If the total number of replicas for a clip equals $\mathcal{N}$, the number of nodes, then this clip is removed from further consideration. Note that the same clip might be chosen repeatedly due to random chance. The termination condition of random is reached when either the total storage capacity is exhausted, $\mathcal{N}$ replicas are created for all clips, or fragmentation of disk space prevents construction of additional replicas because the size of clips exceeds the remaining storage of a node.

We conducted many experiments. We organize the following subsections based on the lessons learned from these experiments. These lessons are as follows. First, replication technique of Section 2.1 is superior to its alternatives based on either display time or size with a heterogeneous databases. This is demonstrated in Section 6 using a closed simulation study. Second, replication technique of Section 2.1 is superior to other alternatives that employ β_i and f_i in different combinations. Third, the exact computation of Equation 12 is worth its complexity. An approximation would cause an inferior technique to appear better than its true capabilities.

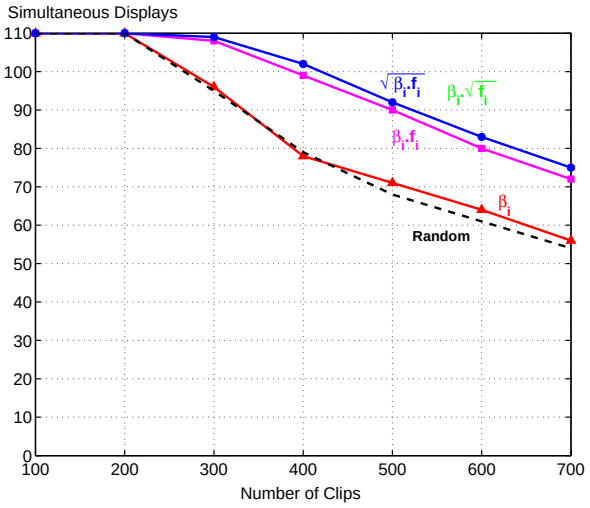
4.1 A Comparison of Alternative Replication Strategies

Instead of using bandwidth, one may employ either display time or size of a clip to determine its number of replicas, e.g., with size, we use $\sqrt{s_i.f_i}$ instead of $\sqrt{\beta_i.f_i}$ in Equation 3. In all our experiments, the replication technique of Section 2.1 either outperforms or provides the same performance as these alternatives and Random. With a homogeneous database, replication using either size or display time performs the same as bandwidth [12] because bandwidth is a function of size and display time, i.e. $\beta_i = \frac{s_i}{d_i}$. Moreover, all techniques provide a similar performance when either bandwidth is abundant or the repository is so small that it can be replicated aggressively. Thus, we present results from the heterogeneous database where bandwidth is scarce. Figure 2 shows the number of simultaneous displays supported by $\sqrt{\beta_i.f_i}$, $\sqrt{d_i.f_i}$ and $\sqrt{s_i.f_i}$ as a function of the number of nodes for a heterogeneous database consisting of 1300 clips. This repository is 1.4 Terabytes (TB) in size. Each node is configured with 100 GB of storage. Based on the 100 iterations with a different random seed for generation of request, Figure 2 shows the worst, average, and best of these 100 experiments. This is an asymmetric configuration and we vary the system size from 2×10 to 2×55 H2O devices. This increases the storage capacity of the system from 1.9 to 10.9 TB. Ideally, there should be a linear increase in the number of simultaneous displays such that 110 displays are supported with the 2×55 configuration.

This can be achieved if the requests issued by the H2O devices closely match the expected frequency of access to the clips. This is shown in Figure 2c where the best sequence of requests with $\sqrt{\beta_i.f_i}$ increased almost linearly. When the distribution of requests does not match due to the use of a random number generator, a higher bandwidth is consumed by those requests referencing the less popular titles. This exhausts the available bandwidth, resulting in a sub-linear increase in the number of simultaneous displays as a function of the number of nodes, see Figures 2a and 2b. The $\sqrt{\beta_i.f_i}$ replication constructs the near optimal number of replicas for each clip and maximizes the utilization of available bandwidth to outperform other strategies. Beyond 70 nodes, both $\sqrt{d_i.f_i}$ and $\sqrt{s_i f_i}$ perform the same as Random.



4a. Homogeneous-Asymmetric, 2×55



4b. Homogeneous-Symmetric, 10×11

Figure 4: Number of simultaneous displays as a function of the number of clips.

The main observation from these results is that the replication class of techniques based on bandwidth (β_i) is superior to those based on either size (s_i), display time (d_i), or random.

4.2 A comparison of alternative strategies using bandwidth

We compared the replication technique of Section 2.1 with the following simpler alternatives:

1. Bandwidth: replicates clips proportional to their bandwidth by using β_j instead of $(\beta_j.f_j)^\alpha$ in Equation 3.

2. Bandwidth and frequency of access: Is similar to Webster's Divisor method and replicates clips proportional to their bandwidth and frequency of access by using $\beta_j.f_j$ (employs $\alpha = 1$ instead of 0.5 in Equation 3).

3. Bandwidth and square-root of the frequency of access: employs $\beta_j.\sqrt{f_j}$ instead of $\sqrt{\beta_j.f_j}$ in Equation 3.

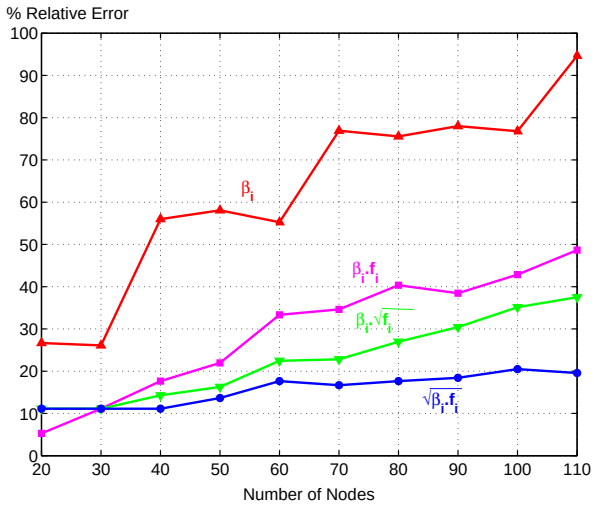
Figure 3 shows the average number of simultaneous displays supported by each technique as a function of the number of nodes across 100 iterations. The technique based on bandwidth alone (β_i) is inferior to the other alternatives because it assigns a higher number of replicas to the least frequently accessed video clip instead of the most frequently accessed audio clip. For example, with 110 nodes, it assigns 7 replicas to the least frequently accessed video clip and 1 replica to the most frequently accessed audio clip[3]. This replication is contradictory to the objective of Equation 2, enabling a random replication technique to either provide similar performance or outperform it. The other two alternatives perform between β_i and $\sqrt{\beta_i.f_i}$ as they incorporate frequency of access in a variety of ways. It is important to note that a technique based on $\beta_i.\sqrt{f_i}$ is competitive and is identical to $\sqrt{\beta_i.f_i}$ with a symmetric topology.

As one increases the size of a repository (by increasing its number of unique clips) while maintaining the same amount of storage per node, the number of simultaneous displays supported by a H2O cloud decreases. Figure 4 shows this for two different configurations (asymmetric and symmetric) for a homogeneous repository. Both repositories consist of 110 nodes with 10.9 TB of storage capacity. As we increase the number of 2 hour video clips from 100 to 700, the repository size increases from 351 GB to 2.4 TB. The number of simultaneous displays with a symmetric topology is higher than that with asymmetric topology because it offers more network paths (and a higher amounts of network bandwidth). With 500 clips and $\sqrt{\beta_i.f_i}$, the symmetric topology supports twice as many display when compared with the asymmetric topology. The decrease in simultaneous displays with $\sqrt{\beta_i.f_i}$ is less dramatic than the other alternatives because it approximates a nearly optimal number of replicas for each clip. Once again, a technique based on $\beta_i.\sqrt{f_i}$ provides a performance almost identical to $\sqrt{\beta_i.f_i}$.

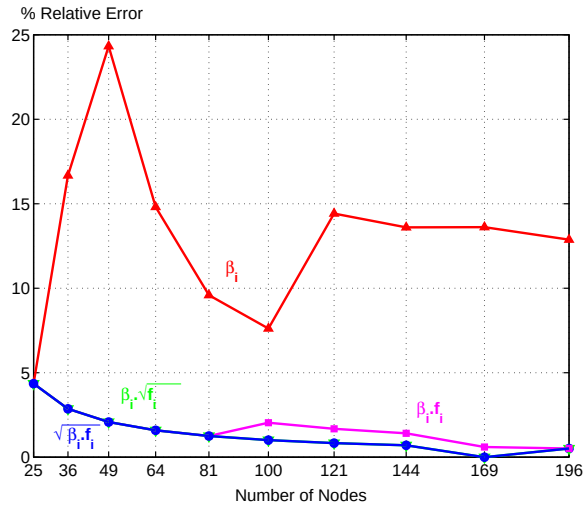
4.3 A comparison of exact and approximation models

We wanted to quantify the tradeoff associated with using an approximation to estimate the number of nodes with a distance less than or equal to l from $H2O_{(i,j)}$. This is because Equation 9 (approximation) is simpler than Equation 12 (exact). We computed the percentage difference between the number of simultaneous displays supported using each alternative. The results obtained form two different topologies and a hetero-

[3]With $\sqrt{\beta_i.f_i}$, these numbers are 4 and 15, respectively.

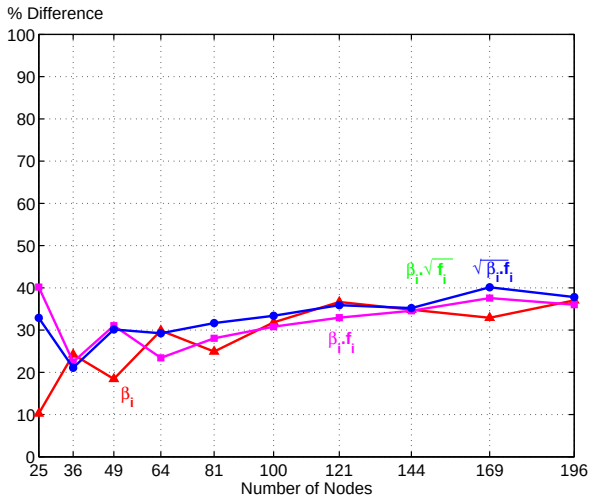


5a. Heterogeneous-Asymmetric, m=1300

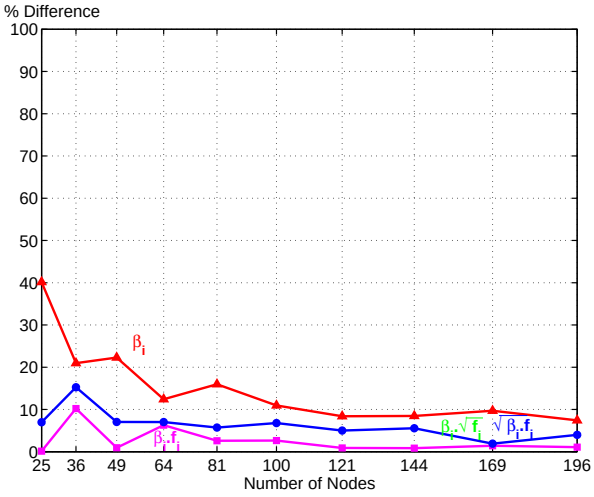


5b. Heterogeneous-Symmetric, m=2250

Figure 5: Percentage error attributed to use of an approximation model.



6a. Before adjusting for bandwidth fragmentation



6b. After adjusting for bandwidth fragmentation

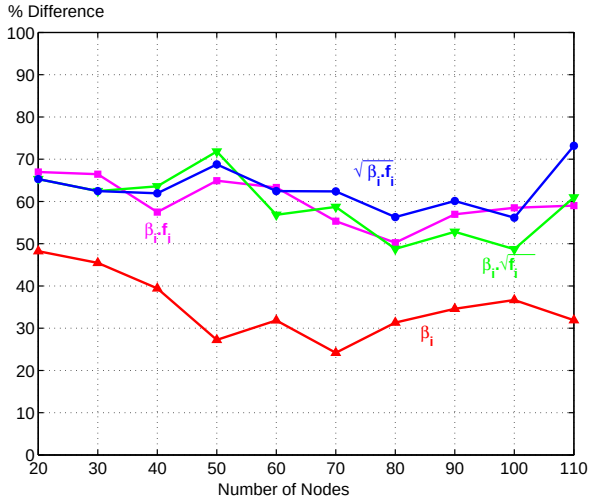
Figure 6: Validation of the analytical models for a Homogeneous-Symmetric with m=400 clips

geneous databases are presented in Figure 5. These figures show approximation makes a naive technique based on bandwidth alone (β_i) appear better than its true capabilities. This is specially true for asymmetric topologies. With 110 nodes, Figure 5a shows the β_i technique that replicates clips proportional to their bandwidth is 90% better than its true potential. At the same time, other techniques do not observe such a percentage improvement. As expected, the approximation of Equation 9 has a less dramatic impact with a symmetric topology. This is because the number of phantom nodes is minimized with this topology. These results justify the complexity of Equation 12 to accurately estimate the number of nodes within a fixed distance of $H2O_{(i,j)}$.

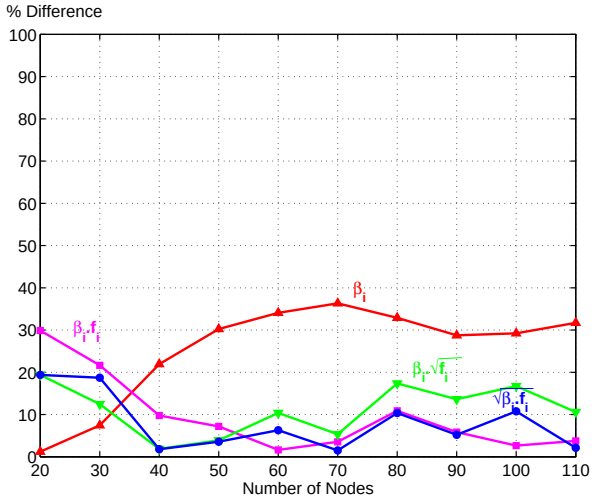
5. VALIDATION

In order to validate our analytical models, we used a simulator for both static and mobile networks. This simulator models a different number of nodes organized in alternative topologies. Each node was configured with 100 GB of storage. For each technique, we computed the number of replicas using the discussion of Section 2.1. Next, we placed these replicas across the nodes of a grid topology randomly. We analyzed both symmetric and asymmetric grid topologies as follows. Each node chooses a random clip based on the Zipfian distribution. Next, it invokes a centralized admission control technique that either admits or rejects this request. In order to admit the request, the admission control must reserve a path from a producing H2O device to the requesting client. (Admission control is detailed in Section 6.) We assume a 4 MbpS bandwidth for each wireless connection between two nodes with a manhattan distance of one.

Figures 6a and 7a show the percentage difference between the analytical and simulation results for different topologies and databases, % diff = $100 \times \frac{|analytical - simulation|}{simulation}$. Both topologies deviate from the analytical expectation due to fragmentation of bandwidth: network bandwidth of some H2O



7a. Before adjusting for bandwidth fragmentation



7b. After adjusting for bandwidth fragmentation

Figure 7: Validation of the analytical models for a Heterogeneous-Asymmetric with m=1300 clips

devices remains idle because a path cannot be established between H2O$_p$ and H2O$_d$ referencing that clip. This is due to bottlenecks attributed to random generation of requests with the simulator. The analytical models of Section 3 do not model these realities of a system and deviate.

In general, the amount of fragmented bandwidth is higher with an asymmetric configuration. It is 30% for symmetric and 43% for asymmetric. We subtracted this fraction from the total bandwidth assumed by the analytical models, and re-invoked the analytical model to obtain the number of simultaneous displays. Next, we computed the percentage difference between this number and those reported by the simulator. The obtained results are presented in Figures 6b and 7b. These results show the analytical models of Section 3 are fairly accurate once the impact of bandwidth fragmentation is incorporated.

6. A CLOSED SIMULATION STUDY

The discussion of Section 4 focuses on the number of simultaneous displays supported by a replication strategy. In this section, we use the simulation model of Section 5 to investigate the start-up latency of the alternative replication strategies. These results show the near optimal technique to almost always provide a better start-up latency when compared with other alternatives. The details of our simulator and its obtained results are presented below.

We used a closed simulation model to measure the start-up latency observed with our proposed technique. As a comparison, we also analyzed techniques using size and display time where s_j and d_j are used in Equation 3 instead of β_j, respectively. We analyzed two system loads: 20% and 90%. At the beginning of the simulation study, for a given system load (say 90%), 90% of H2O devices were chosen randomly to request a clip. The requested clip was chosen using a random number generator conditioned with the Zipfian distribution. Once a H2O device completes its display of a clip, the simulator chooses one of the idle H2O devices randomly to issue a request for another randomly chosen clip. In essence, the think time between subsequent requests is zero and a 90% load is maintained on the system. We employ a centralized admission control to either admit or reject a new request. Upon the arrival of a new request, this component identifies the closest server containing a copy of the referenced clip. Next, it invokes the ford-fulkerson algorithm and finds the path with the maximum available bandwidth connecting the client to the server. If no paths are identified then admission control queues the new request and waits for an active request to finish its display. Once this event is encountered, admission control re-invokes the ford-fulkerson algorithm to detect a potential path on behalf of each queued request. All those requests with a candidate path are scheduled for display. Note that the ford-fulkerson algorithm may produce multiple candidate paths. The admission control chooses one randomly.

The admission control may allocate bandwidth which is a fraction of that required to display a clip. In this case, it requires the displaying H2O to prefetch enough data to prevent data starvation. Assuming S_C denotes the number of blocks in a clip, the number of prefetched blocks is [11]: $S_P = S_C - [\frac{B_{Assigned}}{B_{Display}} \times S_C]$, where $B_{Assigned}$ is the bandwidth allocated by the admission control, $B_{Display}$ is the bandwidth required to display the clip, and S_C is the number of blocks that constitute the referenced clip. The time required to stage S_P on the displaying H2O device dictates the start-up latency encountered by this display.

Table 2 shows the average start-up latency observed with three alternative replication techniques: based on $\sqrt{\beta_i.f_i}$, $\sqrt{d_i.f_i}$, and $\sqrt{s_i.f_i}$. Start-up latency is defined as the elapsed time from when a request for a clip is issued to the onset of its display. With a replication technique and a configuration, we analyzed 100 random placements of the replicas proposed by a technique across its available nodes. For a placement, we measured the number of simultaneous displays using the experimental methodology of Section 4 where each node issues a request and the admission control component either schedules or rejects this request. This was repeated 20 times to compute the average number of maximum simultaneous displays for a placement. Of the 100 placements for a technique, we recorded the placement that results in the best and worst average number of simultaneous displays. (We did not consider the average because a placement out of 100 might not be rep-

Placement	Replication Technique	Symmetric: 7×7, m=1300		Asymmetric: 2×25, m=2250	
		20% Load	90% Load	20% Load	90% Load
Best	$\sqrt{\beta_i.f_i}$	35.41	227.37	169.98	764.38
	$\sqrt{d_i.f_i}$	371.92	282.57	606.82	924.51
	$\sqrt{s_i.f_i}$	320.13	583.62	1270.25	2143.93
Worst	$\sqrt{\beta_i.f_i}$	475.14	788.71	736.29	1304.01
	$\sqrt{d_i.f_i}$	423.75	708.94	1094.65	3376.21
	$\sqrt{s_i.f_i}$	634.89	1254.32	2054.84	4289.77

Table 2: Startup latency of alternative replication techniques with the best and worst placement for a heterogeneous database. All reported times are in Seconds.

resentative of the true average.) Next, we configured our simulation model with each placement and measured its average response time for 20% and 90% workloads. For each, the same sequence of requests is issued with all replication strategies.

Table 2 shows $\sqrt{\beta_i.f_i}$ provide either comparable or superior performance when compared with its alternatives. Moreover, it highlights the importance of data placement. With all replication strategies, there is a significant difference between observed latencies with the best and worst placements for all configurations. The best placement provides substantial savings in the observed latencies. Typically one expects a higher start-up latency with a higher system load. This is violated by replication using $\sqrt{d_i.f_i}$ with its best replica placement across the symmetric topology (start-up latencies of 371 with 20% load and 282 with 90% load). This is because a certain combination of requests with a 20% load results in substantial fragmentation of bandwidth, causing other requests to wait for a long time. This combination is not repeated with a 90% load.

7. CONCLUSIONS

The primary contribution of this study is to motivate a family of techniques that employs $(\beta_i.f_i)^\alpha$ to determine the number of replicas for m continuous media clips that constitute a repository. We analyze two members of this family where $\alpha = 0.5$ and 1. Our analytical and simulation studies demonstrate the superiority of $\alpha = 0.5$. We are extending this study in two directions. First, we are investigating a decentralized implementation of our proposed replication strategy. Second, we are investigating placement techniques, which control the assignment of replicas to nodes in a manner that maximizes the utilization of a replica. Both extensions must consider heterogeneous H2O devices offering different amounts of network bandwidth and storage capacities. Finally, we are investigating extensions of dynamic replication techniques [17, 5] to our environment.

8. ACKNOWLEDGMENTS

This research was supported in part by NSF grant IIS-0307908, National Library of Medicine LM07061-01, and an unrestricted cash gift from Microsoft Research.

9. REFERENCES

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. Caching on the World Wide Web. *Knowledge and Data Engineering*, 11(1):95–107, 1999.

[2] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *INFOCOM (1)*, pages 126–134, 1999.

[3] V. Bush. As We May Think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[4] P. Cao and S. Irani. Cost-Aware WWW Proxy Caching Algorithms. In *Proceedings of the 1997 Usenix Symposium on Internet Technologies and Systems (USITS-97)*, Monterey, CA, 1997.

[5] C. Chou, L. Golubchik, J. C. S. Lui, and I. Chung. Design of Scalable Continuous Media Servers. *Kluwer Multimedia Tools and Applications*, 17(2/3):181–212, July/August 2002.

[6] E. Cohen and S. Shenker. Replication Strategies in Unstructured Peer-to-Peer Networks. In *Proceedings of the ACM SIGCOMM*, August 2002.

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

[8] J. Gemmell, B. Gordon, R. Lueder, S. Drucker, and C. Wong. MyLifeBits: Fullfilling the Memex Vision. In *ACM Multimedia*, December 2002.

[9] J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe. Multimedia Storage Servers: A Tutorial. *IEEE Computer*, 28(5):40–49, 1995.

[10] S. Ghandeharizadeh. H2O Clouds: Issues, Challenges and Solutions. In *Fourth IEEE Pacific-Rim Conference on Multimedia*, December 2003.

[11] S. Ghandeharizadeh, A. Dashti, and C. Shahabi. Pipelining Mechanism to Minimize the Latency Time in Hierarchical Multimedia Storage Managers. *Computer Communications*, 18(3):38–45, March 1995.

[12] S. Ghandeharizadeh and T. Helmi. An Evaluation of Alternative Continuous Media Replication Techniques in Wireless Peer-to-Peer Networks. In *Third International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE, in conjunction with MobiCom'03)*, September 2003.

[13] S. Ghandeharizadeh and R. Muntz. Design and Implementation of Scalable Continuous Media Servers. *Parallel Computing*, 24(1):91–122, May 1998.

[14] T. Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic Approaches*. MIT Press, 1988.

[15] S. Jin and A. Bestavros. GreedyDual* Web Caching Algorithm: Exploiting the Two Sources of Temporal Locality in Web Request Streams. In *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, May 2000.

[16] S. Jin and A. Bestavros. Temporal Locality in Web Request Streams: Sources, Characteristics, and Caching Implications (Extended Abstract). In *Proceedings of the ACM-SIGMETRICS*, June 2000.

[17] P. W. K. Lie, J. C. S. Lui, and L. Golubchik. Threshold-Based Dynamic Replication in Large-Scale Video-on-Demand Systems. *Kluwer Multimedia Tools and Applications*, 11(1):35–63, May 2000.

[18] A. Mahanti, D. L. Eager, and C. L. Williamson. Temporal Locality and its Impact on Web Proxy Cache Performance. *Performance Evaluation*, 42(2–3):187–203, 2000.

[19] J. Wolf, P. Yu, and H. Shachnai. DASD Dancing: A Disk Load Blanacing Optimization Scheme for Video-on-Demand Computer. In *Proceedings of ACM SIGMETRICS*, pages 157–166, May 1995.