Capturing the Spatio-Temporal Behavior of Real Traffic Data [⋆]

Mengzhi Wang, Anastassia Ailamaki, and Christos Faloutsos

Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213
{mzwang,natassa,christos}@cs.cmu.edu

Abstract

Traffic data, such as disk and memory accesses, typically exhibits burstiness, temporal locality, and spatial locality. However, except for qualitative speculations, it is not even known how to measure the spatio-temporal correlation, let alone how to re-produce it realistically. In this paper, we propose the "entropy plots" to quantify the correlation and develop a new statistical model, the "PQRS" model, to capture the burstiness and correlation of the real spatio-temporal traffic. Moreover, the model requires very few parameters and offers linear scalability. Experiments with multiple real data sets show that our model can mimic real traces very well.

Key words: traffic modeling, space-time correlation, spatio-temporal burstiness, locality, entropy

1 Introduction

System design is typically evaluated through trace-driven simulations in which real traces are fed into the system [2,19]. Collecting traces involves system instrumentation or system

[⋆] This material is based upon work supported by the National Science Foundation under Grants No. IIS-9988876, IIS-0083148, IIS-0113089, IIS-0209107, IIS-0133686, CCR-0205544, and by the Defense Advanced Research Projects Agency under Contract N66001-00-1-8936. Mengzhi Wang is partially supported by IBM CAS fellowship. Additional funding was provided by donations from Intel. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, DARPA, or other funding parties.

simulation [17], which is expensive in terms of time and human resources. Modeling traffic data with a simple and accurate statistical model, on the other hand, has several advantages. Firstly, we can run 'what if' scenarios, by generating as long or as short a trace as we want; or by varying the load, burstiness and other parameters of our statistical model. Secondly, we need much less storage space: a statistical model typically requires only a handful of parameters. Finally, we can do analytical performance studies. If we know, for instance, that our traffic is Poisson, we can estimate analytically queue length distributions at a server with a given service time distribution.

There have been significant efforts on modeling the temporal burstiness and spatial locality of traffic data. However, none of them explicitly measures the spatio-temporal correlation or provides statistical models to incorporate the correlation. When designing a model to approximate traffic data, we need to address the following concerns:

- Capture the temporal behavior. Various types of traffic data have been shown to exhibit self-similarity and burstiness.
- Capture the spatial behavior. Are all the disk blocks equi-probable (e.g., random accesses in credit card applications), piece-wise uniform, or Zipf-like?
- Capture the spatio-temporal correlation. Should we worry about the issue? How close to reality is the (convenient) independence assumption?

Given these requirements, the problem is formalized as follows:

Problem 1 *Given a two-dimensional trace, $Y = \{(t, s)\}$, (i.e. (t, s) defines a request of arrival time t on address s.), develop a mathematical model that can generate a synthetic trace, $Y' = \{(t', s')\}$, that has "similar" spatio-temporal behavior as Y.*

Evaluation of such a model includes comparing the synthetic trace to the real trace in terms of both statistical measures (i.e., mean, burstiness, and correlation) and performance behavior (i.e., response time distributions for disk traces). The latter, in our opinion, is more important for practical reasons. In addition, the model should be compact and efficient. A naive model can simply remember the given trace and reproduce it as a synthetic trace, but this hardly saves any space or effort, nor allows for generation of longer traces.

This paper instroduces the PQRS model that addresses all of the above requirements. The PQRS model (1) requires only 4 parameters, (2) captures both the temporal and spatial behavior of the trace, and (3) preserves the spatio-temporal correlation. In addition, the PRRS model is extremely accurate as our experiments on real data sets show. Finally, the model is efficient; both the trace generation and parameter fitting algorithms scale linearly to the number of requests in the traces.

The paper is organized as follows. Section 2 reviews the related work. Section 3 studies the behavior of the real world traffic and Section 4 provides a measure for both the burstiness and the correlation. Section 5 introduces the PQRS model. Section 6 evaluates the model using real traffic data. Section 7 concludes the paper and provides some future directions.

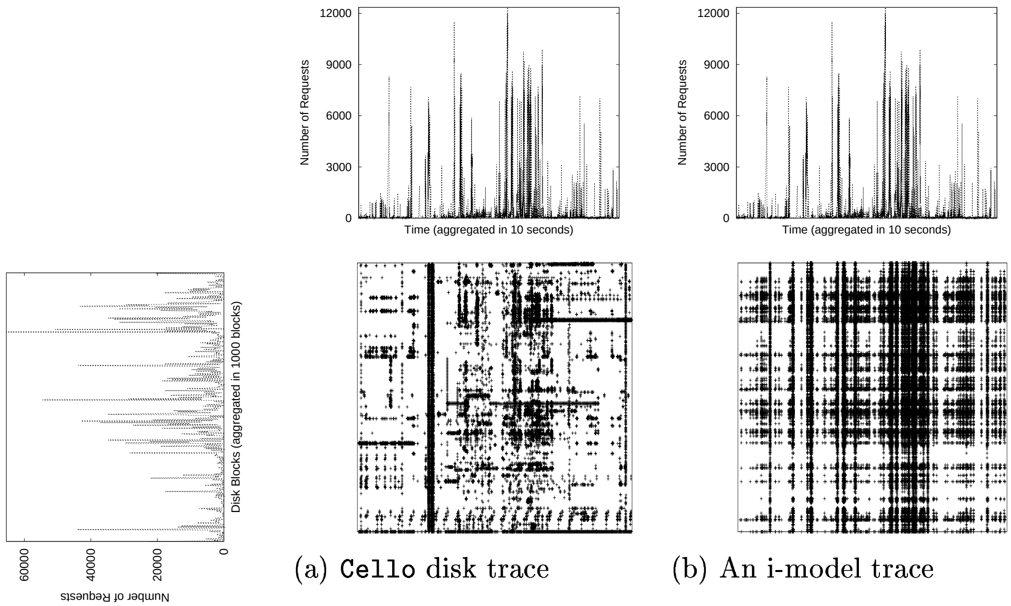
2 Related Work

Traffic modeling has recently attracted much attention, arguably thanks to the discovery of self-similarity in multiple forms of computer generated traffic [15,8,12,14], as well as thanks to the emergence of new applications like web caching. This section outlines past and current research efforts toward traffic modeling.

Focusing only on temporal modeling, one approach is to employ some mathematical models to capture the self-similarity in the request arrival time exhibited in the real traffic data. Successful such models include the fractional Brownian motion [15], fractal ARIMA [11], the ON/OFF model used in the SURGE web trace generator [3], Multifractal Wavelets [16], and the b-model [20]. Related efforts on memory and web proxy trace characterization ultimately aim to estimate cache hit-ratios, for memory or web caches. Several groups identified Zipf-like distributions in document popularity and used the Independent Reference Model or segregated IRM to generate web proxy traces [1,4,5].

Contrary to the modeling of web workloads, modeling I/O workloads can not afford to ignore the location of each request, because both the arrival time and the location are needed to determine its service time. Some of the few exceptions that pay attention to both time and location [19,13] use a mixture of sequential and random access patterns. Although a step to the correct direction, these models require a large number of parameters and they propose no way to measure the spatio-temporal locality, nor do they guarantee that their synthesized traces will exhibit similar locality properties.

The PQRS model introduced in this paper is the first statistical model that captures not only the temporal and spatial behavior, but also their correlation. It achieves the goal by a surprisingly small number of parameters.



(a) `Cello` disk trace (b) An i-model trace

Fig. 1. Time-space plot of `cello` disk trace and an i-model trace. Both have the same marginals.

3 Observing Data: Naive Model

Understanding the real traffic data is crucial before we can start to build the model. This section studies the behavior of the real world traffic data.

3.1 Burstiness

We use the time-space plot to get a basic feeling about the characteristics of real traffic data. The time-space plot $C_{T,S}$ is similar to the memory reference map: The value of $C_{T,S}(t, s)$ is the number of requests of arrival time t and address s. Figure 1 shows the time-space plot for `cello` disk trace [17]. Further projection of the trace onto time and space gives the "marginal" one-dimensional traces C_T and C_S, in which $C_T(t)$ ($C_S(s)$) tells the number of requests of arrival time t (on disk block s).

We observe "bursty" behavior (i.e., non-uniformity) in both marginals.

- **Temporal burstiness.** The temporal burstiness is expected: various traffic traces, such as disk I/O traffic [12] and network traffic [15,8], have all been shown to be bursty.
- **Spatial burstiness.** Similarly, C_S is bursty, too, as noticed before [7]. Some blocks are more popular than others.

It is tempting to conjecture that the marginal distributions are Zipf. To settle the issue,

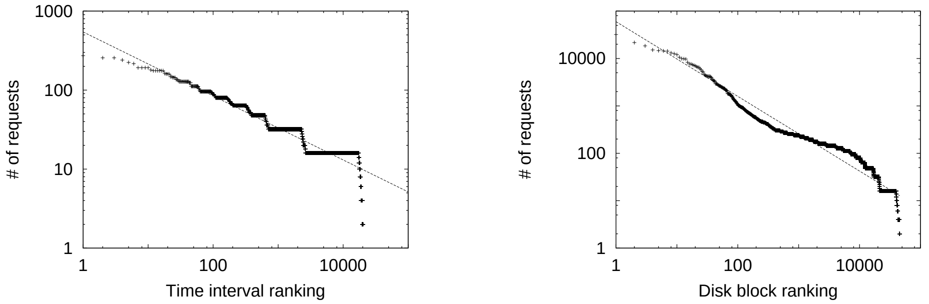


Fig. 2. Rank-frequency plots (in log-log scales) for time and location of `cello` trace. Left: Rank of temporal interval (10 seconds) versus number of requests in that interval. Right: Rank of disk blocks (aggregated in 1000 disk blocks) versus number of requests on these disk blocks. The slopes of the fitting lines are 0.404 and 0.788 respectively.

we give Figure 2, which presents the "rank-frequency" plots for the temporal and spatial marginals in log-log scale. Specifically, Figure 2(a) gives the number of disk requests per time interval (10 seconds), versus the interval-ids sorted in decreasing popularity. Figure 2(b) gives the number of disk requests per 1000 disk blocks, versus the block-ids sorted in decreasing popularity. All axis are logarithmic. The plots are close to lines, leading to the conjecture that they follow generalized Zipf distributions, with slopes 0.40 and 0.79 respectively. However, closer inspection shows a flattening at the left top. This phenomenon had been observed in several data sets even by Zipf himself, who coined the term "top concavity" [21], and who went into great lengths to try to explain it. On the contrary, models that use multifractals, like the upcoming PQRS model, naturally lead to curving rank-frequency distributions [9] and thus provide even better fits.

As mentioned earlier, there are many successful models to capture the temporal behavior of the traffic data. Similarly, they should be able to capture the spatial behavior since the temporal and spatial burstiness looks similar. That is, we can synthesize the marginal traces using existing models. However, a combining algorithm is necessary to combine the marginals into two-dimensional traces. The straight-forward combining algorithm is the i-model.

3.2 I-Model

The i-model generates a two-dimensional trace by "multiplying" two marginal traces. For example, if 10% of the total requests arrive at time t and 5% of the total requests occur on address s, $10\% \times 5\%$ of the total requests have arrival time t and address s. Formally, the i-model specifies that given C_T and C_S,

$$C_{T,S}(t,s) = C_T(t) \times C_S(s)/M, \qquad t = 1, 2, \ldots, \quad s = 1, 2, \ldots, \tag{1}$$

where M is the total number of requests in the trace.

The i-model preserves the temporal and spatial burstiness because the marginals of $C_{T,S}$ are exactly C_T and C_S. In addition, it requires no parameters.

Despite of these advantages, the i-model ignores a very important property of the traffic: *strong spatio-temporal* correlation. The strong correlation suggests that requests coming close in time tend to access nearby objects. Figure 1 (b) shows a two-dimensional i-model trace. Although the i-model trace has the same marginals as the real trace, we observe significant differences between the two traces. The differences are attribute to the existence of strong spatio-temporal correlation in the real trace. In fact, the i-model leads to grossly pessimistic results, as we will show in Section 6.

4 Proposed Method to Quantify Correlation

The i-model ignores the strong spatio-temporal correlation of the real traffic data. Performance behavior of the traffic data heavily depends on the degree of spatio-temporal correlation because requests close in address space take less time to finish. This section introduces the entropy plot to measure the correlation.

4.1 Definitions

Mutual information [18] measures the correlation between two events. We give a brief description of the related concepts. (Please refer to Table 1 for the symbols used in the paper.)

Entropy measures the uniformity of a discrete probability function. **Entropy** on a random variable E, (e.g., disk block id of a random request), is defined as

$$H(E) = -\sum_{i=1}^{N} p_i \log_2 p_i, \tag{2}$$

where p_i is the probability that event E_i will happen (e.g., the i-th block will be hit) and N is the total number of possible outcomes (e.g., total number of disk blocks). H is close

$P_{T,S}(t,s)$	Probability that a request on location s arrives at time t.
$P_T(t)$	Probability that a request arrives at time t.
$P_S(s)$	Probability that a request is on location s
$H(E)$	Entropy of a random variable E
$H_T^{(n)}$	Temporal entropy at aggregation level n
R_T	Slope of the temporal entropy plot
$H_S^{(n)}$	Spatial entropy at aggregation level n
R_S	Slope of the spatial entropy plot
$H_{T,S}^{(n)}$	Joint entropy on time and space at aggregation level n
$R_{T,S}$	Slope of the joint entropy plot
(p,q,r,s)	Parameters to the PQRS model
M	Total number of requests in a trace

Table 1
Symbols table.

to 0 if the distribution is highly skewed and H reaches its maximum value of $\log_2 N$ when the distribution is uniform. In another word, H measures the burstiness of the probability function.

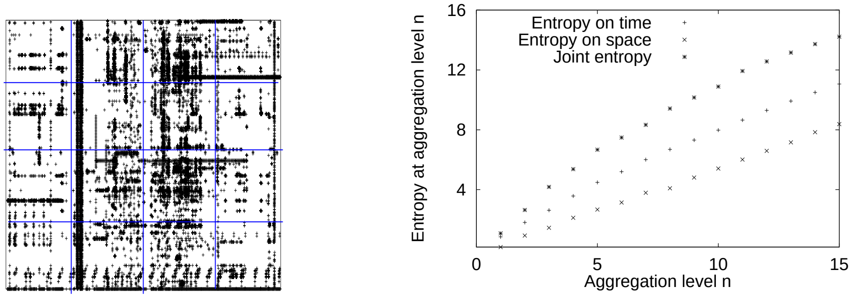
The joint entropy on two random variables is defined similarly: for a given probability function $P = \{p_{i,j}\}$ on two random variables $\{E\}$ and $\{F\}$, (e.g., arrival time and disk block id of a random request), where $p_{i,j}$ gives the probability that both event E_i and event F_j will happen, (e.g., a disk request at block id j arrives at i), the **joint entropy** on E and F is

$$H(E,F) = -\sum_{i,j} p_{i,j} \log_2 p_{i,j}. \tag{3}$$

Definition 2 *The* **mutual information** $I(E;F)$ *on two random variables* E *and* F *is*

$$I(E;F) = H(E) + H(F) - H(E,F). \tag{4}$$

The mutual information $I(E;F)$ indicates the degree of correlation between E and F. It becomes zero if E and F are independent.



(a) Aggregation level 2 ($2^2 \times 2^2$ grids) (b) Entropy plot

Fig. 3. Entropy plot for `cello` disk trace.

4.2 Entropy Plots

We can apply the above definitions to traffic data to measure the burstiness and spatio-temporal correlation. The question is, then, at what the granularity. If we calculate the entropy on the finest resolution, the mutual information on time and space will be very close to zero because no correlation will be observed. Our answer is to calculate the entropy values at all "aggregation" levels.

To find the entropy values at aggregation level n, the time-space plot of the trace is divided into $2^n \times 2^n$ grids. For example, Figure 3 (a) shows the 4×4 grids at aggregation level 2. $P_{T,S}^{(n)}$ describes the probability that a request falls into each grid; that is, the request has arrival time t, $(t_1 \leq t < t_2)$, and address s, $(s_1 \leq s < s_2)$. $P_T^{(n)}$ and $P_S^{(n)}$ are the projections of $P_{T,S}^{(n)}$ on time and space. All the Ps can be easily derived from the given trace.

Definition 3 *The* **entropy plot** *for a given trace is defined by plotting the entropy values against the aggregation level n, where the entropy on time, space and the joint entropy at aggregation level n are*

$$\begin{cases} H_T^{(n)} &= H(P_T^{(n)}); \\ H_S^{(n)} &= H(P_S^{(n)}); \\ H_{T,S}^{(n)} &= H(P_{T,S}^{(n)}). \end{cases} \tag{5}$$

The entropy plot provides an insight on how the burstiness and correlation change across different resolution levels. The points form a line if the burstiness and correlation are stable at all granularities. Interestingly, real traffic has stable burstiness and correlation as the linear entropy plot of the sample disk trace suggests (Figure 3 (b)). This further confirms the self-similarity of the disk I/O workloads.

Lemma 4 *For a trace of stable temporal and spatial burstiness and spatio-temporal correlation, all the entropy plots are linear:*

$$\begin{cases} H_T^{(n)} = nR_T; \\ H_S^{(n)} = nR_S; \\ H_{T,S}^{(n)} = nR_{T,S}. \end{cases} \tag{6}$$

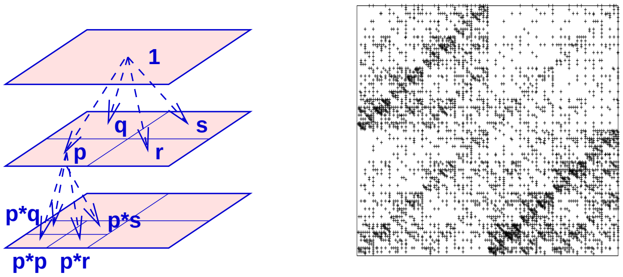
All the proofs are omitted from the paper for brevity. R_T and R_S measure the temporal and spatial burstiness respectively. The intuition behind R_T is the rate of information contained in one more bit of time-stamp. For example, when all the requests come in a burst, all the time-stamps will be the same and all the bits are useless, which leads to $R_T = 0$. When the requests are uniformly distributed along time, all the bits in the time-stamps are useful and R_T, in this case, is 1. Similarly, R_S gives the rate of information in the address bit.

R_I, defined as $R_T + R_S - R_{T,S}$, tells the mutual information per bit (e.g., how much information the time-stamp bit tells about the location of the request). When R_I equals to 0, the time-stamps and the addresses of the requests is independent.

The real traffic data shows strong spatio-temporal correlation. The estimated value of R_T, R_S, and $R_{T,S}$ (Figure 3 (b)) are 0.722, 0.573, and 0.881, leading to $R_I \approx 0.414$. The large value of R_I indicates strong spatio-temporal correlation in real traffic data. R_I for the i-model trace, on the other hand, is 0.001, suggesting independence between time and space. (Hence the name i-model.)

5 Proposed Model: PQRS Model

The independence assumption in the i-model leads to totally different performance behavior even when the synthetic traces and the real traces have the same temporal and spatial characteristics as we will show later in Section 6. Therefore, it's important to incorporate the correlation in the model. This section presents a new statistical model, called the PQRS model, to capture the stable burstiness and correlation of the real traffic data.



(a) Generation algorithm (b) A Sample PQRS trace

Fig. 5. Recursive trace generation process for the PQRS model.

5.1 Generation: PQRS Model

The PQRS model generates a two-dimensional trace using four parameters, namely, p, q, r, s, where $p+q+r+s=1$. The recursive construction is the reverse process of the aggregation in entropy plot calculation (Figure 5 (a)). At the starting point, the probability that a request falls into the whole time-space plot is 1. In step 1, the time-space plot is divided into 2×2 grids and the probability that a request falls in each grid is p, q, r, s respectively. In step 2, each grid is further divided into 4 smaller grids and the requests are distributed to the 4 smaller grids with the same probabilities, p, q, r, s. The process goes on recursively until the required resolution on time and space is achieved (e.g., the size of the grids is smaller than a microsecond or a disk block).

```
initialize the stack;
push the whole trace onto the stack;
while (stack is not empty) do
  pop a grid from the stack;
  if grid is small enough then
    output the requests of the grid;
  else
    divide the grid into $2 \times 2$ grids;
    distribute the requests to the grids;
    push the four grids onto the stack;
```

Fig. 4: PQRS trace generation

The above algorithm assumes the same order of p, q, r, s in all the levels. We can generate a random PQRS trace by imposing a different order at each step. Figure 5 (b) gives a sample trace generated by the PQRS model with p, q, r, s of $0.2, 0.3, 0.4, 0.1$. More requests cluster at bottom right corner since r has the greatest value among the four parameters.

5.2 Parameter Fitting

The recursive construction algorithm guarantees that PQRS traces have stable burstiness and spatio-temporal correlation because all the steps use the same parameters to distribute the requests.

Lemma 5 *Traces generated by the PQRS model have stable burstiness and correlation as they have linear entropy plots.*

$$\begin{cases} H_T^{(n)} &= nH_T^{(1)} = nR_T; \\ H_S^{(n)} &= nH_S^{(1)} = nR_S; \\ H_{T,S}^{(n)} &= nH_{T,S}^{(1)} = nR_{T,S}; \end{cases} \tag{7}$$

Lemma 6 *For a PQRS trace generated with parameter* p, q, r, s, $p+q+r+s=1$, *the entropy rates are*

$$\begin{cases} R_T &= -(p+q)\log_2(p+q) - (r+s)\log_2(r+s); \\ R_S &= -(p+r)\log_2(p+r) - (q+s)\log_2(q+s); \\ R_{T,S} &= -p\log_2 p - q\log_2 q - r\log_2 r - s\log_2 s. \end{cases} \tag{8}$$

Equation 8 suggests that the difference between $(p+q)$ and $(r+s)$ determines the temporal burstiness of the synthetic traces and the difference between $(p+r)$ and $(q+s)$ determines the spatial burstiness. Varying the value of p changes the degree of the spatio-temporal correlation when temporal and spatial burstiness is fixed.

The parameter fitting algorithm for the PQRS model is simple. For a given trace, we estimate the value for p, q, r, s by plugging the slopes of the entropy plot in Equation 8.

The following two lemmas give some additional features of the PQRS model.

Lemma 7 *The Poisson model is a special case of the PQRS model where* $p = q = r = s = 0.25$.

Lemma 8 *The i-model is a special case the PQRS model where* $\frac{p}{q} = \frac{r}{s}$.

5.3 Complexity

The computational complexity of the algorithm is an important property of the model. One would rather choose to collect traces if the trace generation is too slow. Our analysis shows that both the trace generation and parameter fitting algorithms for the PQRS model offer linear scalability to the number of requests in traces.

Lemma 9 *The computational complexity in generating a PQRS trace is $O(M \times N)$, where M is the total number of requests and N is the number of steps in the generation algorithm.*

We outline the proof here. We upper-bound the trace generation through a naive implementation of the algorithm. The recursive generation conceptually forms a quad tree of height N. (See Figure 5 (a).) The 4^n grids in step n form the 4^n nodes at level n in the quad tree. We decide the arrival time t and address s of each request by walking down the quad tree from the root. At each level of the tree, we choose one of the four subnodes with probability p, q, r, s until we reach a leaf node. Enumerating all the requests gives the final trace. Therefore, the complexity of the trace generation is $O(M \times N)$. In reality, N is usually logarithm to the length of the trace in time (or space).

Lemma 10 *The computation complexity for the parameter fitting algorithm of the PQRS model is $O(M \times N)$.*

We sketch the proof here. The number of non-zero grids in each aggregation level is at most M; therefore, it takes $O(M)$ computations to generate a point in the entropy plot. Given that there are N points in the entropy plot, the total computational complexity is $O(M \times N)$.

In summary, the strength of the PQRS model lies in its power as well as in its simplicity. The model generates traces with stable burstiness and correlation as the real traffic data exhibits. In addition, the model offers linear scalability.

6 Experiments

We evaluate the PQRS model using two types of traffic data: disk and memory reference traces. The experiments examine the validity of the PQRS model and compare the performance behavior of the PQRS model traces to the real ones.

We make two main observations. First, the real traffic data have reasonably linear entropy plots which verifies the assumption we made in developing the PQRS model. Second, strong spatio-temporal correlation plays an important role in performance behavior and invalidates the i-model. The PQRS model, on the other hand, leads to performance measures that match the reality.

6.1 Experiment Setup

Two types of traffic data are in use. Table 2 summarizes the data sets.

Trace	Total disk requests	$\hat{R}_T$	$\hat{R}_S$	$\hat{R}_{T,S}$	$I_{T,S}$	$(\hat{p},\hat{q},\hat{r},\hat{s})$
Disk-a	4,575,798	0.641	0.819	1.058	0.402	(0.092,0.745,0.162,0.001)
Disk-r	1,822,781	0.847	0.833	0.984	0.696	(0.016,0.258,0.720,0.006)
Disk-w	3,300,628	0.641	0.728	0.992	0.377	(0.150.0.013,0.053,0.784)
Disk-0	1,101,416	0.814	0.690	0.941	0.563	(0.043,0.184,0.772,0.001)
Disk-2	1,396,649	0.790	0.723	0.904	0.609	(0.200,0.027,0.001,0.772)
Disk-7	371,320	0.722	0.573	0.881	0.414	(0.056,0.135,0.808,0.001)

(a) `Cello` disk trace summary

Trace	Length	Total requests	$\hat{R}_T$	$\hat{R}_S$	$\hat{R}_{T,S}$	$I_{T,S}$	$(\hat{p},\hat{q},\hat{r},\hat{s})$
New Order	14,990,636	4,000,000	0.962	0.200	0.996	0.166	(0.030,0.255,0.001,0.714)
Payment	17,242,172	4,573,044	0.963	0.281	1.042	0.202	(0.239,0.047,0.713,0.001)
Order Status	1,355,168	268,943	0.950	0.456	0.989	0.417	(0.095,0.185,0.001,0.722)
Delivery	525,100	129,388	0.957	0.439	0.987	0.409	(0.090,0.192,0.001,0.717)
Stock Level	14,453,440	3,613,360	0.974	0.349	1.052	0.271	(0.231,0.064,0.704,0.001)
Mix	12,268,876	4,000,000	0.983	0.309	0.990	0.302	(0.248,0.054,0.697,0.001)

(b) TPC-C memory reference trace summary (Trace length in CPU cycles)

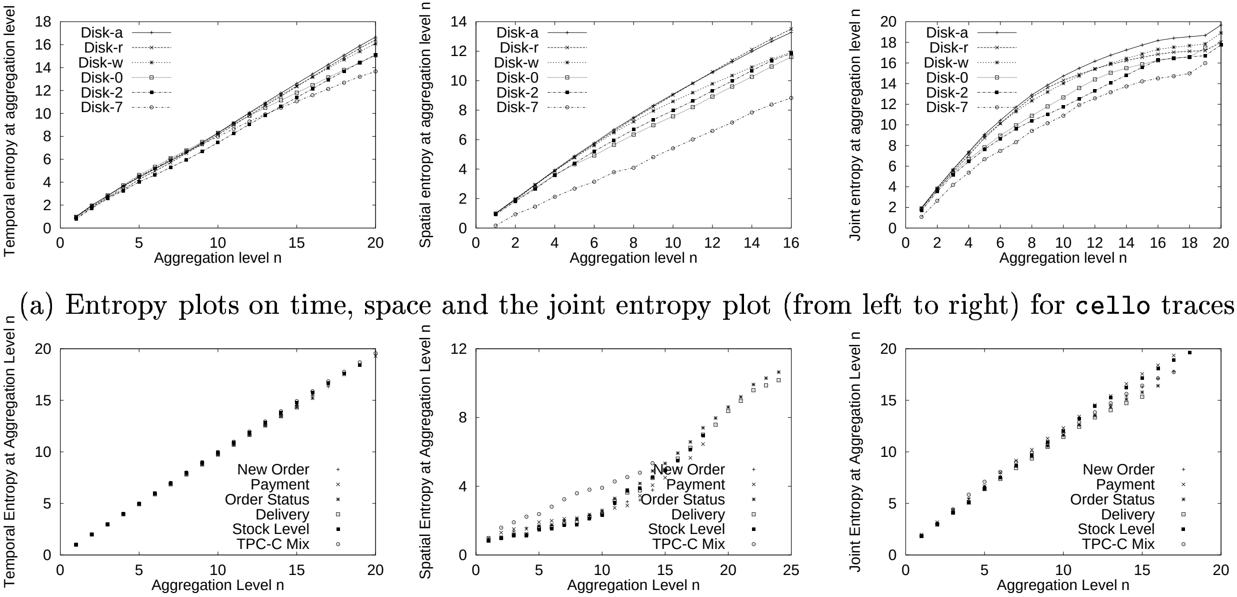
Table 2
Trace Summary

`Cello` disk traces. The disk traces were collected on a UNIX file server in HP on June 12, 1992 [17]. The server has 8 disks attached to it. Total of six traces are in use: `Disk-a` for the aggregation of all the disk requests, `Disk-r` for all the read requests, `Disk-w` for all the write requests, and `Disk-0`, `Disk-2`, `Disk-7` for individual disk 0, 2, 7. All the traces are one day long. The other five disks are not studied because of the small volume of disk requests. The arrival time is accurate to microseconds. The disk block number ranges from 0 to more than 5,000,000.

TPC-C memory reference traces. The TPC-C memory traces were collected on a realistic processor simulator running TPC-C workloads on Shore [6]. There are total of six traces: five for five types of transactions and one for a mixture of different types of transactions. Only references to the heap area are studied here.

Evaluation tools. The ultimate goal of traffic modeling is to facilitate system designs. Therefore, we focus on the performance behavior of the traces. We use the response time and queue length distributions for disk traces and the cache miss ratio for memory reference traces as our performance metrics.

Methodology. We want to answer the following questions. First, does real traffic have stable burstiness and correlation over aggregation? Second, how does the PQRS model perform? The synthetic traces should have the same performance behavior as the real ones if the synthetic traces accurately capture the characteristics of the real traffic data.



(a) Entropy plots on time, space and the joint entropy plot (from left to right) for `cello` traces.

(b) Entropy plots on time, space and the joint entropy plot (from left to right) for TPC-C traces.

Fig. 6. Entropy plots for the two data sets.

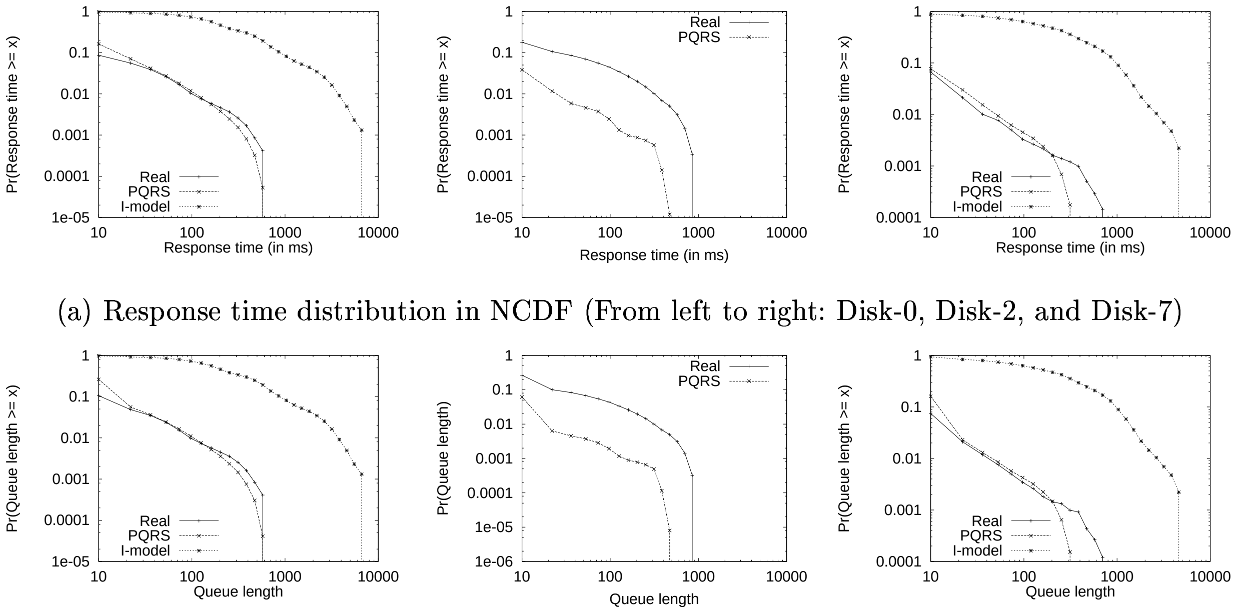
6.2 Model Checking

The PQRS model is designed for the traffic data with stable burstiness and spatio-temporal correlation. Therefore, the traffic data should have linear entropy plots for the PQRS model to work. In addition, we rely on linear entropy plots for estimation of parameter p, q, r, s.

Figure 6 shows the entropy plots for the disk and memory traces. We have made the following observations.

- The entropy plots are reasonably linear, suggesting stable burstiness and correlation in the traces. This ensures that the traces are well within the capability of the PQRS model.
- Strong spatio-temporal correlation exists: R_I ranges from 0.313 to 0.696 for disk traces and from 0.166 to 0.417 for memory traces.
- The PQRS model is able to model uniform traces as well. R_T for the memory traces is close to 1, suggesting a uniform distribution of the memory accesses on time. This is because the program is consistently accessing data during its course of execution.

In summary, real traffic data has stable burstiness and correlation over aggregation and is within the capability of the PQRS model. Strong correlation exists, invalidating the inde-



(a) Response time distribution in NCDF (From left to right: Disk-0, Disk-2, and Disk-7)

(b) Queue length distribution in NCDF (From left to right: Disk-0, Disk-2, and Disk-7)

Fig. 7. Disk Trace Performance Evaluation. (I-model results for disk-2 are missing due to queue saturation.)

pendence assumption of the i-model.

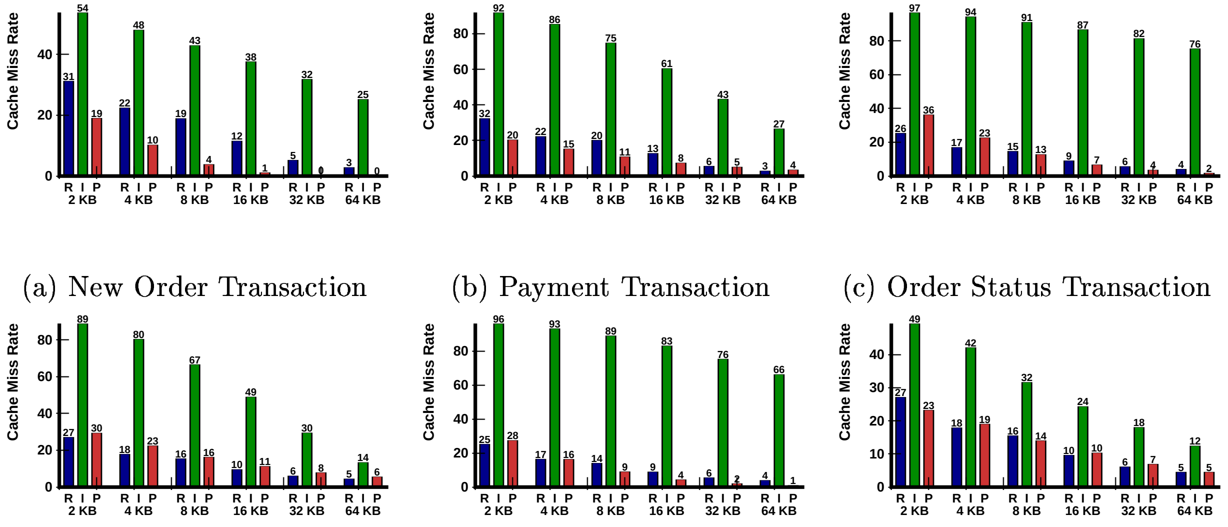
Table 2 lists the estimated p, q, r, s value for the data sets. The following sections compare the performance behavior of the real traces and the PQRS traces generated using these values.

6.3 Disk Trace Evaluation

Figure 7 compares the response time and queue length distributions of the real disk traces and the PQRS traces on a realistic disk simulator [10]. Both distributions are in negative cumulative form (NCDF) and in log-scale. That is, point $(10, 0.01)$ in the response time distribution plot tells that 1% of the disk requests have response time greater than 10 milliseconds. Traces with strong spatio-temporal correlation should have short tails in these distributions as requests close in address can be served quickly.

The comparison shows that the PQRS traces simulates the performance behavior of the real traces very well by accurately capturing the burstiness and the correlation.

- Strong spatio-temporal correlation plays an important role in performance behavior. The i-model traces produce extremely large response time because of the independence assumption although they have exactly the same burstiness along time and space as the real



(a) New Order Transaction (b) Payment Transaction (c) Order Status Transaction

(d) Delivery Transaction (e) Stock Level Transaction (f) TPC-C Transaction Mix

Fig. 8. Comparison of the performance behavior of the memory reference traces: "R" stands for the real traces; "I" for the i-model, and "P" for the PQRS model.

traces. The i-model results for `Disk-2` are missing because the queue becomes long enough to saturate the system.

- The PQRS model works amazingly well in simulating the real traffic by accurately capturing both the burstiness as well as the strong correlation at all aggregation levels.

The above comparison has shown that the PQRS model mimic the real disk I/O traffic very well in performance behavior.

6.4 Memory Trace Evaluation

Memory trace evaluation involves comparing the cache miss rates of the real traces to the PQRS traces. The miss rate is an important performance metric in computer architecture research and it reflects the temporal and spatial locality. Memory references on nearby locations have a better chance to be cache hits if they are close to each other in arrival time. Therefore, strong spatio-temporal correlation leads to low cache miss rates.

Figure 8 compares the cache miss rates for three sets of traces: R for the real traces, I for the i-model traces of the same marginals as the real ones, and P for the PQRS traces with parameters listed in Table 2. Six groups of bars show the cache miss rates on six different cache sizes in each graph.

We observe that the traces with high degree of spatio-temporal correlation, such as the R and P traces, suffer low cache miss rates as we have expected. The relative error of the PQRS traces is within 30%. On the other hand, the I traces, assuming independence between time and space, experience extremely high miss rates and have relative error as high as 1800%.

6.5 Summary

Both disk traces and memory references traces have shown reasonably stable burstiness and spatio-temporal correlation over aggregation as suggested by the linear entropy plots. Strong spatio-temporal correlation exists in both types of traffic data and the correlation has a significant impact on the performance behavior of the traces. Therefore, traffic modeling should take the correlation into consideration.

The PQRS model, carefully designed for such traffic data, is able to replicate the behavior of real traces very well. The i-model, on the other hand, fails to do so by ignoring the correlation.

7 Conclusions

Modeling disk traffic is a hard problem [10], especially when we need to capture both the temporal as well as spatial correlations. In this paper, we propose the entropy plot to measure the spatio-temporal correlation and we discover that the burstiness and correlation remain stable for many scales for real traffic data, which is another evidence of self-similarity. We develop a simple statistical model, the PQRS model, to capture the characteristics of such traffic: it can be bursty or uniform in time, bursty or uniform in space, and it can give zero to 100% correlation between space and time.

Additional contributions include

- We are the first to quantify the popular, but vague intuition that memory and disk accesses exhibit locality, not only in space or time, but in space-time as well.
- We give fast, scalable algorithms to run our model: both the parameter fitting and trace generation algorithms require linear time on the number of requests.
- Experiments on multiple real data sets show that the simple PQRS model can mimic them very well, leading to good performance predictions (cache-hit-ratios, queue length distributions). In contrast, the independence model (i-model), fails miserably.

Several promising research directions stem from this work. For example, we plan to apply the PQRS model to other spatio-temporal settings (e.g., earthquakes over space and time). Alternatively, given the p, q, r, s values of a trace, we plan to analytically derive the performance measures of interest (like the cache-hit ratio, or disk queue length distributions).

Acknowledgements

We wish to thank John Wilkes from HP Storage System Lab for providing us with `cello` disk trace and constant feedback on our work. We are grateful to Jiri Schindler for helping on the disk simulator and Prof. Tara Madhyastha and her student, Bo Hong, for their quick responses to our need. In particular, we appreciate Prof. Anthony Brockwell for his insight on classical statistical models. We would also like to thank IBM and Intel for their support.

References

[1] Virgílio Almeida, Azer Bestarvos, Mark Crovella, and Adriana de Oliveira. Characterizing reference locality in the WWW. In *PDIS'96*, 1996.

[2] Guillermo Alvarez, Kim Keeton, Erik Riedel, and Mustafa Uysal. Charaterizing data-intensive workloads on modern disk arrays. In *Workshop on Computer Architecture Evaluation using Commercial Workloads*, 2001.

[3] Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *SIGMETRICS'98*, pages 151–160, 1998.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: evidence and implications. In *Proceedings of the IEEE Infocom Conference*, 1999.

[5] Mudashiru Busari and Carey Williamson. On the sensitivity of web proxy cache performance to workload characteristics. In *Proceedings of the IEEE Infocom Conference*, 2001.

[6] M. Carey, D. J. DeWitt, M. Franklin, N. Hall, M. McAuliffe, J. Naughton, D. Schuh, M. Solomon, C. Tan, O. Tsatalos, S. White, and M. Zwilling. Shoring up persistent applications. In *SIGMOD*, 1994.

[7] Stavros Christodoulakis. Implications of certain assumptions in database performance evaluation. *ACM Transactions on Database Systems*, 9(2):163–186, 1984.

[8] Mark E. Crovella and Azer Bestavros. Self-similarity in world wide web traffic evidence and possible causes. In *Proc. of the 1996 ACM SIGMETRICS Intl. Conf. on Measurement and Modeling of Computer Systems*, May 1996.

[9] Christos Faloutsos, Yossi Matias, and Avi Silberschatz. Modeling skewed distributions using multifractals and the '80-20 law'. In *VLDB*, 1996.

[10] Gregory R. Ganger. Generating representative synthetic workloads: An unsolved problem. In *Proceedings of the Computer Measurement Group (CMG) Conference*, 1995.

[11] Mark W. Garrett and Walter Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *SIGCOMM'94*, 1994.

[12] Marı́a E. Gómez and Vicente Santonja. Self-similiarity in I/O workload: Analysis and modeling. In *Workshop on Workload Characterization*, 1998.

[13] Marı́a E. Gómez and Vicente Santonja. A new approach in the modeling and generation of synthetic disk workload. In *International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2000.

[14] Steven D. Gribble, Gurmeet Singh Manku, Drew Roselli, Eric A. Brewer, Timothy J. Gibson, and Ethan L. Miller. Self-similarity in file systems. In *SIGMETRICS'98*, 1998.

[15] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE Transactions on Networking*, 2:1–15, 1994.

[16] Ruldolf H. Riedi, Matthew S. Crouse, Vinay J. Ribeiro, and Richard G. Baraniuk. A multifractal wavelet model with application to network traffic. *IEEE Transactions on Information Theory*, 45(3), April 1999.

[17] Chris Ruemmler and John Wilkes. Unix disk access patterns. In *Proceedings of the Winter'93 USENIX Conference*, pages 405–420, 1993.

[18] Claude E. Shannon and Warren Weaver. *Mathematical Theory of Communication.* University of Illinois Press, 1963.

[19] Elizabeth Shriver, Arif Merchant, and John Wilkes. An analytic behavior model for disk drives with readahead caches and request reordering. In *SIGMETRICS'98*, 1998.

[20] Mengzhi Wang, Tara Madhyastha, Ngai Hang Chan, Spiros Papadimitriou, and Christos Faloutsos. Data mining meets performance evaluation: Fast algorithm for modeling bursty traffic. In *ICDE'02*, 2002.

[21] G.K. Zipf. *Human Behavior and Principle of Least Effort: an Introduction to Human Ecoloby.* Addison Wesley, 1949.