Supplemental Material

Appendix 1 Host Model

Here we describe the model of hosts, including the different resources in the model and how they were measured. Given the application context described in Section 3, we consider hosts to have 6 key resources:

- **Processing Cores**: the number of primary processing cores. This does not include GPU cores or other special purpose secondary processors. For Windows machines this was measured by the `GetSystemInfo` function, for Apple/Linux/Unix machines by the `sysconf` or `sysctl` functions.
- **Integer computing speed**: the speed of a processing core as measured by the Dhrystone [34] 2.1 benchmark in C.
- **Floating point computing speed**: the speed of a core as measured by the 1997 Whetstone benchmark in C [35].
- **Processor Cache**: the amount of processor cache on the host. For Windows machines this was measured by the `get_cpuid` function, for Linux/Unix machines by reading the /proc/cpuinfo file. Cache size was not measured on machines running OS X.
- **Volatile Memory**: the amount of random access memory in the host. For Windows machines this was measured by the `GlobalMemoryStatusEx` function, for Apple/Linux/Unix machines by the `Gestalt`, `sysconf` and `getsysinfo` functions.
- **Non-volatile storage**: unused space in long term storage including hard disk drives. This does not necessarily include all storage devices attached to a host, only those accessible to the BOINC client. For Windows machines this was measured by the `GetDiskFreeSpaceEx` function, for Apple/Linux/Unix machines by the `statfs` or `statvfs` functions.

One issue with measuring main memory is that some graphics chips use main memory for graphics processing purposes, making that part unavailable to the OS and thus not measured. The BOINC system does not record whether this type of graphics card is present or how much main memory it uses. To compensate we round up to the nearest 256MB for hosts with 1GB or less of measured memory, and up to the nearest 512MB for hosts with more than 1GB of memory. Doing so increases host memory in our data by a mean of 70MB (median of 8MB) so it does not significantly skew our results.

Although Whetstone and Dhrystone have various shortcomings, we feel their use is acceptable as an approximate measure of host computational ability. In the official BOINC distribution these benchmarks were compiled using the -O2 flag for the UNIX version, the -Os flag for the Mac version using XCode and the /O2 /Ob1 flags for Windows version using Visual Studio.

For the purposes of measuring host characteristics, a host is considered to be active (and thus included in the measurements) at a time T if the host first connected to the server on or before time T and the most recent connection to the server is on or after time T. Because we care about the aggregate statistics of hosts, we did not consider host availability at a detailed level. For more fine-grained analysis of host availability see [32], [33].

TABLE 3
Parameters for model resources of the form $Log_2(at + b)$ where $t = (year - 2006)$. r is the correlation coefficient.

		All	S@H	E@H	WCG	R@H	CP
Cores	a	0.211	0.191	0.342	0.143	0.210	0.197
	b	0.369	0.360	0.303	0.502	0.358	0.430
	r	0.998	0.999	0.974	0.996	0.999	0.999
Memory MB	a	0.395	0.359	0.608	0.286	0.376	0.354
	b	9.86	9.83	9.77	10.0	9.86	9.98
	r	0.996	0.997	0.967	0.999	0.999	0.997
Cache KB	a	0.395	0.350	0.473	0.432	0.388	0.341
	b	9.643	9.639	9.642	9.474	9.515	9.746
	r	0.945	0.966	0.899	0.968	0.971	0.963
Integer MIPS	a	0.228	0.230	0.249	0.221	0.217	0.233
	b	11.13	11.09	11.11	11.19	11.21	11.18
	r	0.996	0.995	0.993	0.997	0.988	0.995
Flop MIPS	a	0.147	0.147	0.160	0.155	0.138	0.136
	b	10.31	10.29	10.30	10.27	10.36	10.40
	r	0.997	0.995	0.992	0.992	0.993	0.995
Avail. Disk GB	a	0.410	0.364	0.715	0.306	0.364	0.351
	b	4.98	5.00	4.83	4.95	5.01	5.12
	r	0.992	0.996	0.938	0.987	0.995	0.998

Appendix 2 Host Resource Overview

From 2006 to 2010, for the combined projects the mean values rose by varying degrees - the number of cores in a host rose from 1.32 to 2.33 (77% increase), memory from 913 MB to 2738 MB (200% increase), cache from 707 KB to 2238 KB (217% increase), the integer performance from 2313 MIPS to 4207 MIPS (82% increase), floating point performance from 1281 MIPS to 1900 MIPS (48% increase) and mean available disk space rose from 33.9 GB to 101.8 GB (200% increase). For Einstein@home the increases were greater - from 1.31 to 3.24 cores (147% increase), 910 MB to 4666 MB memory (413% increase), 773 KB to 2605 KB cache (237% increase), 2310 to 4396 integer MIPS (90% increase), 1275 to 1957 floating point MIPS (54% increase) and 32.5 GB to 212.2 GB available disk space (553% increase). Conversely, the increases for World Community Grid were less impressive - from 1.40 to 2.10 cores (50% increase), 1035 to 2314 MB memory (124% increase), 685 KB to 2296 KB cache (235% increase), 2168 to 4028 integer MIPS (86% increase), 1195 to 1871 floating point MIPS (57% increase) and 31.1 GB to 72.3 GB available disk space (132% increase). These are shown in graphical form in Figure 9.

Figure 9 shows the mean, 5% and 95% quantiles of the base-2 logarithm value of resources (cores, memory, cache, computing speed and storage) over a 4-year period for two sample projects (Einstein@home and World Community Grid) as well as the combined total of all projects. The mean of resource values is indicated by black dots, the 5% and 95% quantiles by (red) dotted and (blue) dashed lines. The thin black line indicates the linear best fit of the form $y = a(year - 2006) + b$. Table 3 shows the parameters (a and b) of the best fit

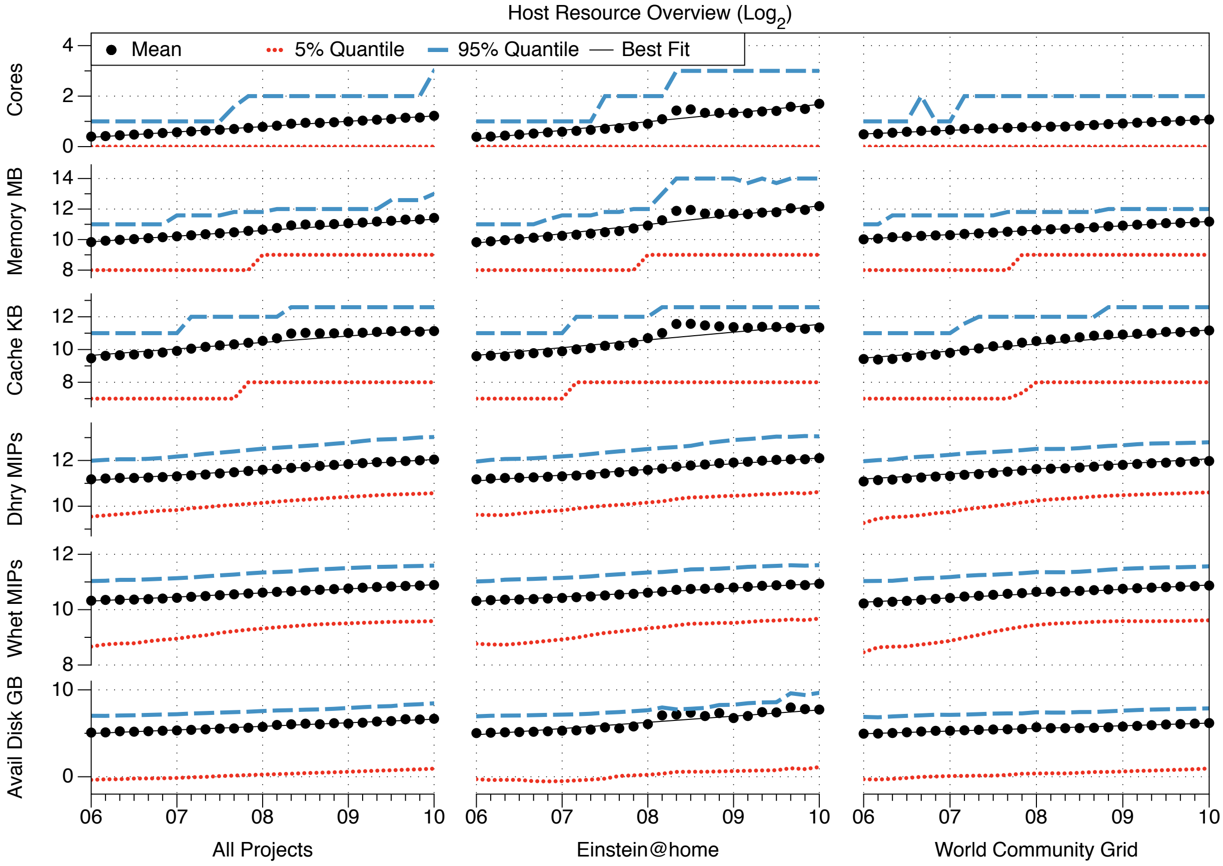


Fig. 9. Overview of host resources including mean and quantiles.

along with the correlation coefficient r of the fit for hosts across all projects and for each project. The correlation coefficients indicate the combined host resources are well-modeled by a linear fit with greater than a 0.99 correlation coefficient for all resources except cache.

Appendix 3 Processor and OS Composition

Here we briefly examine the composition of processors and operating systems among the hosts and how it has changed over time. Because availability and performance of new processor models and OSs cannot be predicted far in the future, we do not include processor model or OS type in our model. There is also a wide range of speeds and capabilities even within a single processor family, making it difficult to predict the effect on a particular application.

TABLE 4
Host processors over time (% of total).

	2006	2007	2008	2009	2010	2011
PowerPC G3/4/5	3.56	4.06	2.67	1.84	1.28	0.85
Athlon XP	12.96	9.17	5.89	3.51	2.17	1.26
Athlon 64	6.96	10.09	11.64	10.70	8.93	7.16
Other AMD	8.20	8.58	7.93	7.65	9.37	11.52
Pentium 4	37.23	32.59	26.21	18.90	14.15	9.40
Pentium M	5.72	5.64	4.89	4.21	2.56	1.48
Pentium D	0.77	3.01	4.31	3.74	2.93	2.31
Other Pentium	3.75	2.40	1.92	3.07	5.03	5.80
Intel Core 2	0.97	3.69	15.11	26.62	35.44	37.67
Intel Celeron	5.46	6.18	5.72	5.36	4.57	3.49
Intel Xeon	2.20	2.96	3.71	5.01	4.91	5.31
Other x86	10.05	8.79	8.42	7.43	7.93	13.15
Other	2.14	2.85	1.56	0.96	0.72	0.59

Table 4 shows the change in processor composition as a percent of total over the full data period. The Pentium 4 and similar Pentium processors were dominant in 2006 comprising over 45% of processors, but by 2011 fell significantly to comprise only 19% of processors. Pentium 4 processors stopped shipping in 2008, so we expect the numbers to fall further as the processors are replaced over time. In place of the Pentium 4, the Intel Core 2 started shipping in 2006 and went from zero to over a third of available processors. The Intel Core 2 will likely stop shipping by 2011 so we expect the share to fall in the near future. The share of Athlons dropped slightly from about 30% in 2006 to 20% in 2011. Other architectures also slowly fell in share over the data period. The jump in "Other x86" at 2011 is due to the release of the Intel Core i3, i5, and i7 processors in 2010.

Table 5 shows the change in host operating system over the data period. During this time, hosts using Windows XP dropped from 71% to 43% while Windows Vista and Windows 7 increase from 0% to 34%. Other Windows OSs dropped from 18% to 5%. The remainder

TABLE 5
Host OS over time (% of total).

	2006	2007	2008	2009	2010	2011
Windows XP	71.17	71.87	68.92	64.25	55.92	43.17
Windows Vista	0	0	6.99	13.44	15.03	11.44
Windows 7	0	0	0	0	7.96	22.15
Windows 2000	12.26	8.22	5.13	3.08	1.74	0.95
Other Windows	5.49	5.20	4.38	3.82	1.74	4.60
Mac OS X	5.14	7.58	7.00	6.98	7.15	7.85
Linux	5.76	6.91	7.39	8.29	9.15	9.73
Other	0.19	0.23	0.18	0.15	0.13	0.13

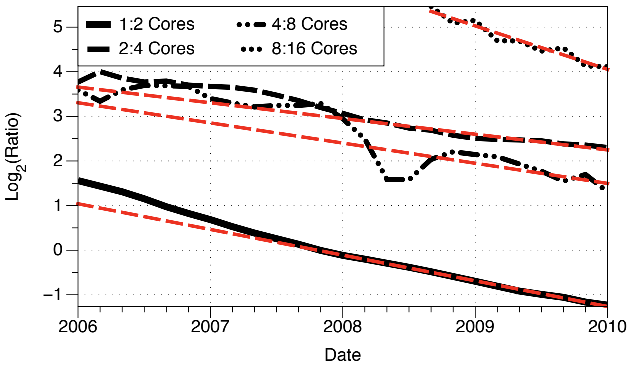


Fig. 10. Ratios of host cores fit by the function $Log_2(a(year - 2006) + b)$ (shown in red). Table 2 has the a and b values.

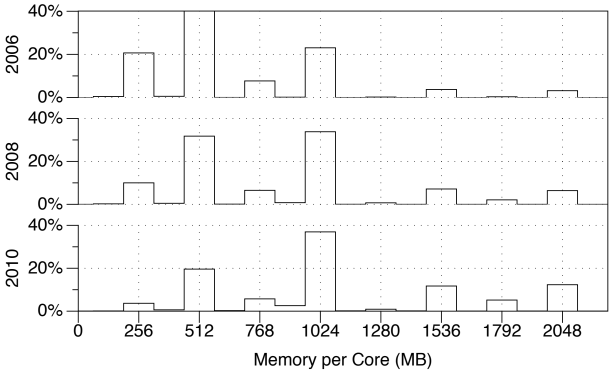
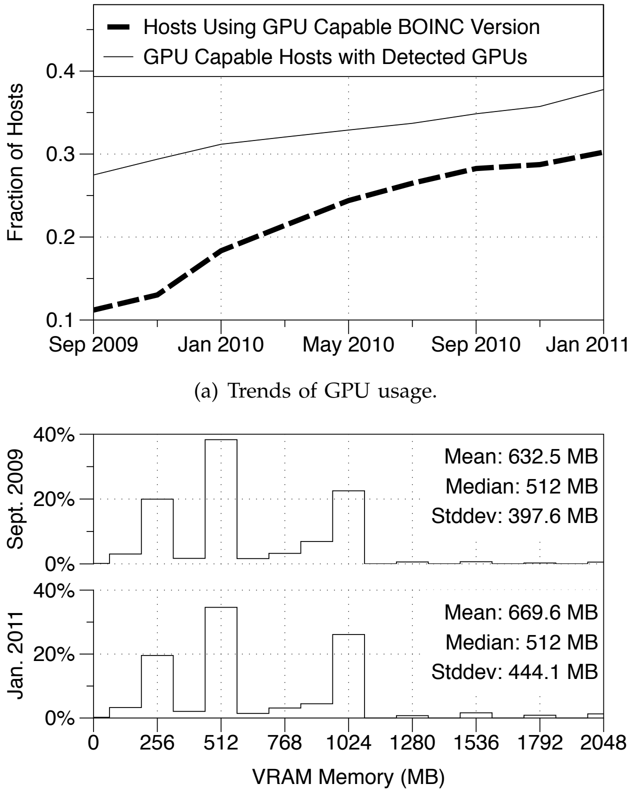


Fig. 11. Percent of hosts with varying per-core-memory in different years.

of hosts use Mac OS X (steady around 7%) or Linux (growing from 6-10%). These results indicate that although Windows is still the most common operating system, the share of Linux is slowly growing.

Appendix 4 GPU Analysis

In recent years, GPGPU (general purpose computation on graphics processing unit) has become popular and many computers include one or more GPUs which can be used for GPGPU. BOINC did not start recording GPU resource information until September 2009, so we feel there is insufficient data to include GPU resources in our model. However, for completeness we include a brief analysis of GPU resources in this section.



(a) Trends of GPU usage.

(b) GPU memory distributions.

Fig. 13. Trends of GPU usage.

Even though BOINC started recording GPU characteristics in September 2009, it could only do so on hosts with an updated client. Figure 13(a) shows the trends of GPU availability among hosts. We see that the fraction of clients which were able to detect GPUs grew from 11% in September 2009 to 30% by January 2011. Among the hosts with the updated version of BOINC, not all detected a GPU available for GPU based processing. The fraction of hosts detecting a GPGPU capable GPU grew from 27% in September 2009 to 38% in January 2011. Assuming a linear rate of increase, GPGPU capable hosts will not become a majority until late 2012.

Figure 13(b) shows the distribution of memory in GPUs from September 2009 and January 2011. Between these dates, mean GPU memory increased by only 6% from 632.5 MB to 669.6 MB. There was a slight jump of GPUs with 1GB or more of memory from 23% to 26% of total. However, these rises are far slower than the rate of increase in total host memory. In addition, hosts with more than 1GB of GPU memory still comprise less than 6% of GPGPU capable hosts (1.5% of all hosts), indicating memory bound applications may not be suitable for Internet end host GPUs in the near future.

Appendix 5 Model Usage and Validation

Here we use the model developed in Section 4 to generate hosts at a specified point in time. We use standard

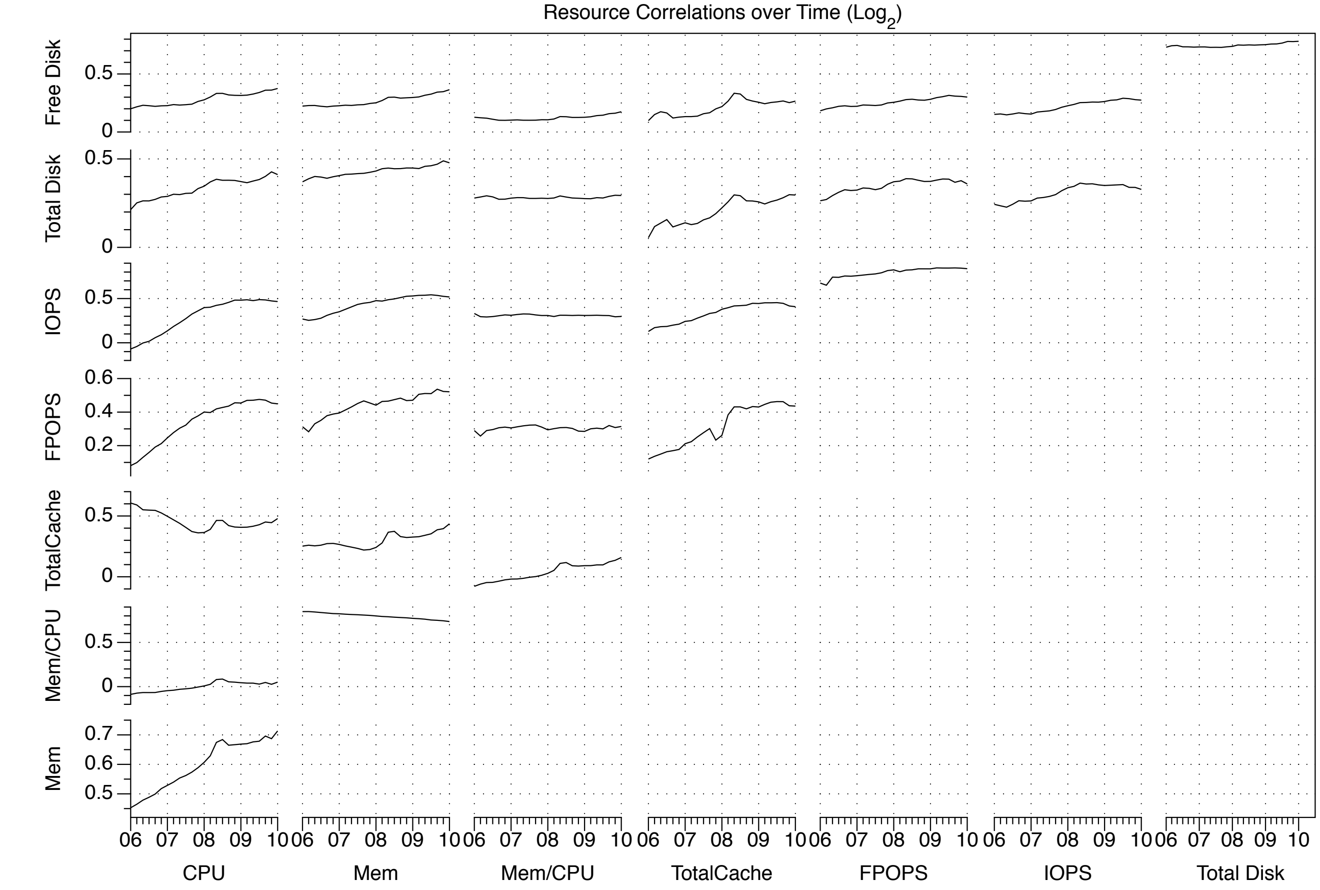


Fig. 12. Resource correlations over time.

statistical methods to validate the generated hosts and compare them to the actual host data.

5.1 Model Based Host Generation

Figure 14 shows the flowchart of host creation using our model. First the user selects the date for the generated host. This date is used to generate a core count by sampling a discrete probability distribution created based on core ratios calculated with the values from Table 2.

Using the method described in Section 4.8, correlated values are generated from sampled distributions for per-core-memory, cache, processor benchmark speeds and disk space. Similar to core count, per-core-memory is sampled from a probability distribution generated using the ratio equations from Table 2. Total memory is calculated by multiplying per-core-memory by the number of cores. We generate correlated values for cache, benchmark values and disk space and re-normalize them to the mean and variance predicted from Table 2.

5.2 Model Validation

Using our model we generate a set of sample hosts for January 1, 2011. Figure 15 shows a comparison of the generated and actual data. The lower half of the figure shows a comparison of cumulative distribution functions (CDFs), the upper half shows a comparison of cumulative mass functions (CMFs). The generated values are close to the actual data, with means ranging from a difference of 0.2% for Whetstone MIPS up to 11.1% for host cache. We also generated QQ-plots (not included for space reasons) to visually confirm the fit.

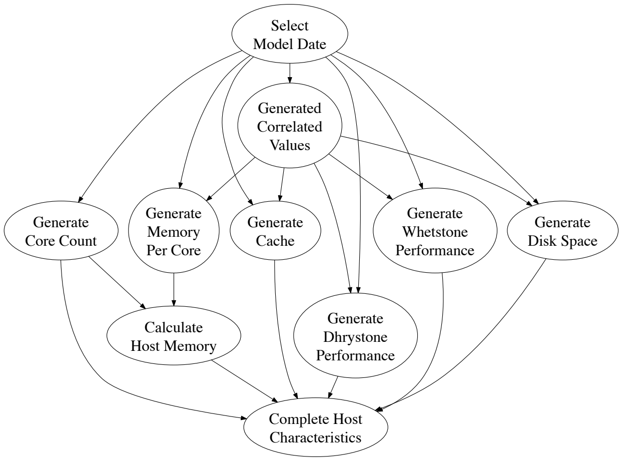

Select Model Date
Generated Correlated Values
Generate Core Count
Generate Memory Per Core
Generate Cache
Generate Whetstone Performance
Generate Disk Space
Calculate Host Memory
Generate Dhrystone Performance
Complete Host Characteristics

Fig. 14. Flowchart of host creation.

Table 6 shows the difference in correlation coefficients

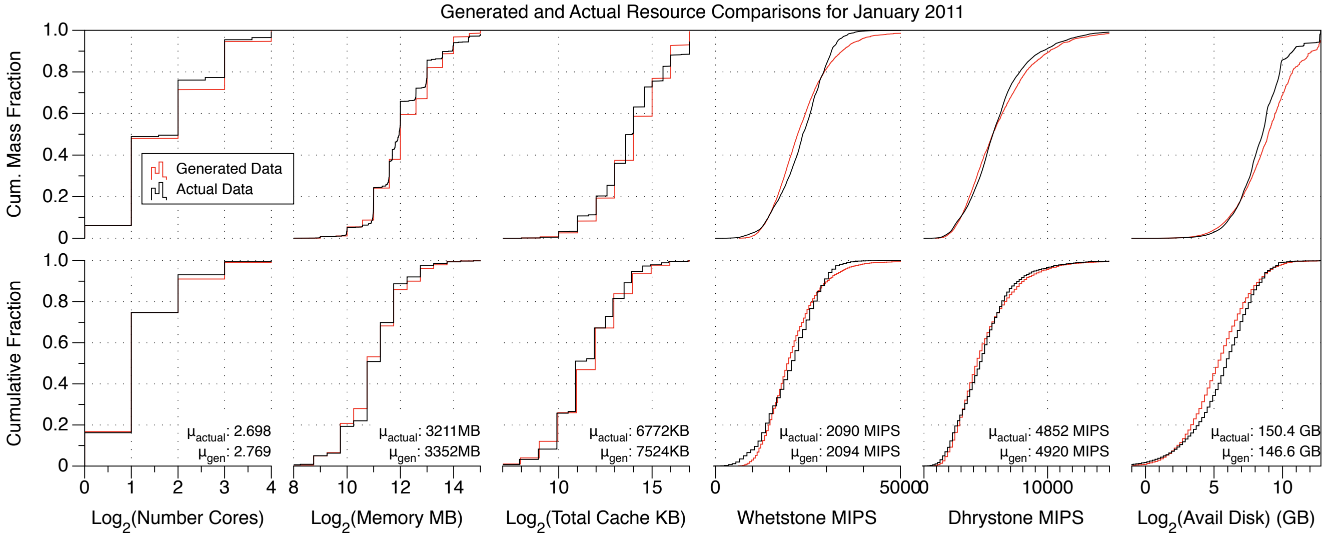


Fig. 15. Comparison of generated and actual data.

TABLE 6
Difference in correlation coefficients between generated and actual data.

	Cores	Mem.	MPC	Cache	Whet	Dhry	Disk
Cores	0.00	0.06	0.08	0.41	0.45	0.50	0.33
Mem.	-	0.00	0.00	0.21	0.31	0.39	0.30
MPC	-	-	0.00	0.19	0.07	0.00	0.06
Cache	-	-	-	0.00	0.18	0.06	0.11
Whet	-	-	-	-	0.00	0.19	0.13
Dhry	-	-	-	-	-	0.00	0.18
Disk	-	-	-	-	-	-	0.00

TABLE 7
Simulation parameters for sample applications.

Application	SETI@home	Folding@home	Climate Pred.	P2P
Cores (α)	0.01	0.30	0.10	0.01
Memory (β)	0.10	0.10	0.20	0.03
Cache (γ)	0.20	0.20	0.20	0.05
Dhrystone (δ)	0.20	0.10	0.10	0.15
Whetstone (ϵ)	0.40	0.25	0.25	0.01
Disk (ζ)	0.09	0.05	0.15	0.75

between generated and actual host data for January 2011 calculated the same way as in Figure 12. Since the matrix is symmetric, we replace identical entries with a dash for clarity. We see that most of the resource values we explicitly correlate have excellent agreement between the generated and actual data. The difference in correlation coefficients for the actual and generated data ranges from nearly 0 (e.g., Dhrystone with per-core-memory and cache) up to roughly 0.2 at the worst (e.g., Whetstone with cache and Dhrystone). Resources that were not explicitly correlated (e.g., cores and total memory with all other resources) have worse differences in correlation ranging from roughly 0.2 to 0.5. However, given the questionable usefulness of core correlation with other resources described in Section 4.2, this should not overly affect model quality.

Appendix 6 Simulation Based Model Validation

Here we perform simulations to demonstrate the value of our model compared to other host resource representations. Currently, most Internet-based computing applications have focused on exclusively utilizing the CPU and most scheduling algorithms aim to optimize the application makespan. However, recent work has investigated using other resources, such as disk space, to perform a wider range of services. Certain applications may benefit disproportionally from hosts with increased memory, greater processor speed or more disk space.

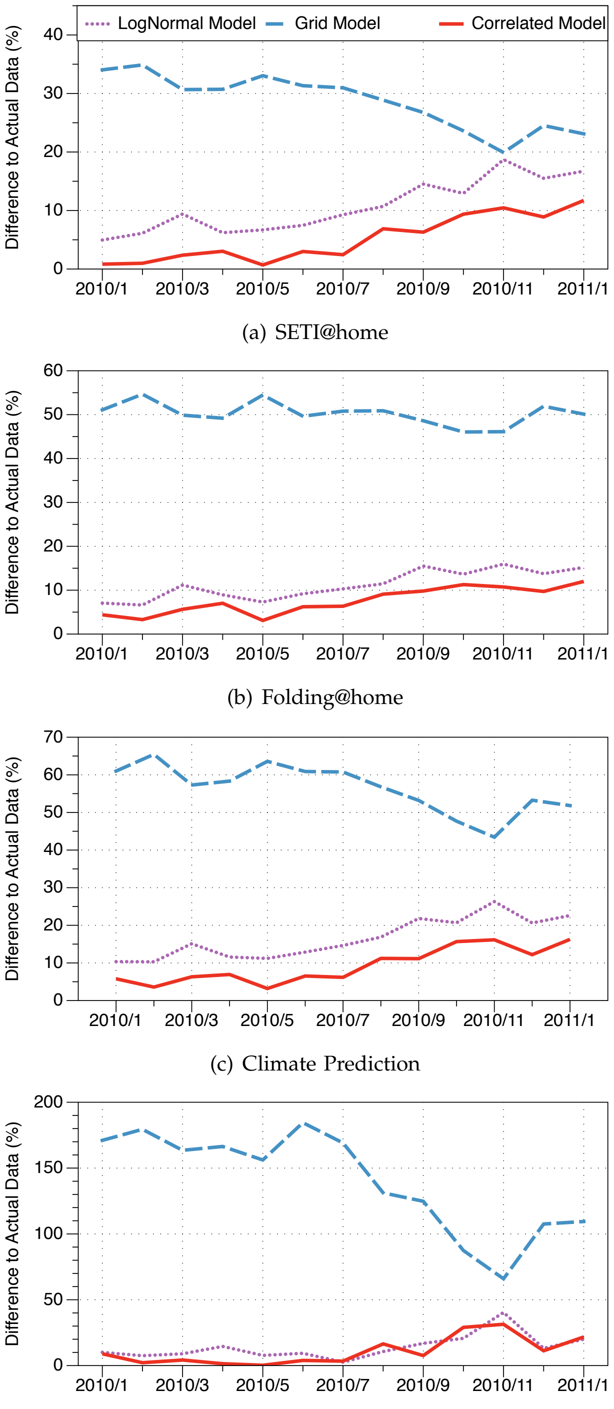
Because of this, in these simulations we attempt to maximize total application utility of host resources rather than minimizing execution time. Host utility can be thought of as how much benefit an application gets from running on a certain host. We feel this is a better fit for analyzing our model since it includes all resource types and is able to represent a generalized application that uses a mix of resources or prefers certain resources over others. To represent the utility of resources for a given application we use a variation on the well known Cobb-Douglas [36] utility function from economics. This function models the output of an economy based on the labor and capital inputs.

Rather than the normal inputs of labor and capital, we use the resources for a host H: core count (P_H), total memory (M_H), cache (C_H), integer/floating point speed (I_H and F_H) and disk space (D_H). Then the utility Y of running an application A on host H can be written as:

$$Y_A(H) = P_H^{\alpha} M_H^{\beta} C_H^{\gamma} I_H^{\delta} F_H^{\epsilon} D_H^{\zeta} \tag{1}$$

where α, β, γ, δ, ϵ, ζ represent the utility returns to scale on each resource to the application.

Table 7 shows the parameters we use for some sample applications in our simulation. We chose these applications as a representative set of possible applications



(a) SETI@home

(b) Folding@home

(c) Climate Prediction

(d) P2P

Fig. 16. Utility simulation results.

requiring Internet end hosts. SETI@home represents an application doing radio signal analysis, which benefits from fast processing but does not require significant memory or disk space and does not utilize multiple cores. Folding@home represents a parallel molecular dynamics simulation, which can use multiple cores and requires a medium amount of memory, but little disk. Climate prediction requires a mix of all resources, with some emphasis on floating point speed. P2P uses Internet end hosts to perform distributed file sharing and benefits greatly from large disks.

The simulation calculates the utility of each application running on each resource using Equation 1, then assigns resources to applications in a greedy round-robin fashion where each application takes turns reserving a host which will give it the most utility. In the simulations we compare our correlated host synthesis model with two others. The first is a simple model which uses extrapolation of the values in Figure 9 and samples resource values from uncorrelated log-normal distributions. The second is based on the Grid resource model by Kee et al. [22]. This model uses a log-normal distribution for processors, a time and processor dependent model of memory and an exponential growth model for disk space. We assign processor speed using the same method as the log-normal model, and we use the same estimated mean/variance as our correlated model for the Grid resource model parameters where appropriate. To make the comparison fair, we also update this model with more recent values from our analysis and generate a mix of old/new hosts based on the host lifetime distribution. All the models are of similar complexity in terms of parameters, though our correlated model has more parameters with regards the core and memory resources.

The simulation calculates the total utility for each application with the resources created by each model. Figure 16 shows the results for the simulation, comparing the normal distribution model, Grid resource model and correlated resource model described in this paper. The simulations were run with data from January 2010 to January 2011. The figure shows the percent difference between the total utility calculated using the specified model and the utility using the actual host data. Multiple simulation runs showed little variance in results due to the large numbers of hosts involved.

The figure shows that the correlated model generally has an smaller difference to the actual data than the other models. For the SETI@home application, the correlated model ranges between 0-12% difference from the actual data, the Grid model between 20-35% and the normal distribution model between 5-19% difference. The Folding@home application shows a similar range of model accuracy, with the correlated model between 3-12% difference, the Grid model between 46-55% and the normal model around 7-16% difference. This is likely since the correlated model accurately captures the correlations between benchmark, memory and core count, which are all key components to the application. The correlated model also appears to more accurately model core count, which is a significant resource in the application.

The Climate Prediction application has similar results, with 3-16% difference for the correlated model, 43-65% difference for the Grid model and 10-26% difference for the normal distribution model. The P2P application shows a major difference between the models, with a 0-31% difference for the correlated model, 66-184% for the Grid model and 3-40% difference for the normal distribution model. This large error in the Grid model is likely because it uses an exponential growth rule for disk space, which overestimates the available space.

Based on these results, we have shown that our model more closely reflects actual host resources, resource correlations and time dependent behavior. Our model matches actual data quite well and is equally or more accurate than simpler uncorrelated models or other Grid models for modeling host resources.