A Large-Scale Computing Infrastructure for Design Education

CECCATO Cristiano, FISCHER Thomas, LI CHUN-MAN Gary, FRAZER John

The Hong Kong Polytechnic University School of Design
http://www.sd.polyu.edu.hk

Most departmental computing infrastructure reflects the state of networking technology and available funds at the time of construction, which converge in a preconceived notion of homogeneity of network architecture and usage patterns. The DMAN (Digital Media Access Network) project, a large-scale server and network foundation for The Hong Kong Polytechnic University's School of Design, was created as a platform that would support a highly complex academic environment while giving maximum freedom to students, faculty and researchers through simplicity and ease of use. As a centralized multi-user computation backbone, DMAN faces an extremely heterogeneous user and application profile, exceeding implementation and maintenance challenges of typical enterprise, and even most academic server set-ups. This paper summarizes the specification, implementation and application of the system while describing its significance for design education in a computational context.

***Keywords**: Design Computation; Design Teaching; Design Research; Network; Infrastructure; Distributed; Collaboration*

Introduction

The School of Design of The Hong Kong Polytechnic University is one of the most diversified academic departments in Hong Kong and in the South China region. The department encompasses all academic design disciplines including Industrial, Environmental, Fashion, Visual Communication and Interactive Systems Design. More than forty full-time staff members and an average of twenty-five part-time staff support some 700 students at the undergraduate and graduate levels including doctoral research. An increasingly large portion of student work, as well as educational material prepared by the faculty, is accomplished through the computer, mostly in an online/web setting as reported on various occasions, e. g. Falk et. al (2000) or Ceccato et. al (2000).

The ongoing development of academic curricula means that particularly in the undergraduate courses, discipline areas of concentration are becoming intertwined, with students choosing major and minor concentrations in separate areas of study, and often questioning the very nature of the subject by mixing and blurring preconceived discipline boundaries. This is coupled with ever-increasing amounts of high-end, user-level computing infrastructure that is deployed throughout the department. These result in a formidable level of complexity within the academic process that requires a unifying platform to maintain stability and provide consistent performance, security and flexibility for all users. The DMAN system is a large-scale infrastructure that was developed and implemented at the School of design to address these issues. With this paper we do not only give a report on our design considerations and the successful system implementation but also provide planning information and give suggestions for similar future projects in other Design and Architecture schools.

A Platform for Post-Digital Design Education

User-Centred Infrastructure

From the start, DMAN was conceived to address all significant elements of the School of Design's academic fabric. First and foremost, the central leitmotif of the project was guided by the vision to achieve a system that would be part of a post-digital academic environment, meaning that computational resources would be subliminal yet ubiquitous, unassuming but powerful. Above all, a new system would have to support almost every conceivable usage pattern, user-end computer hardware and design software with a single, platform-independent and highly user-friendly interface.

Teaching and Learning

Many of the School of Design's courses are now offered as online "courseware" modules where students interact with one another through virtual portals that function as project platforms. Beginning in the spring of 2000, teaching topics and creative assignments have been increasingly integrated in web-based interfaces that permit students to create individual and group projects that make use of the web as both a presentation and a project research medium.

Prior to DMAN, this work was achieved by implementing each course on its own custom-programmed Linux servers, see Ceccato et al. (2000). While a fast-track pioneering effort, this method was inefficient, partly due to the lack of computing power capable of handling over 100 concurrent users and low system availability and reliability (system crashes, limited backup facilities). While still pursuing low-cost, Linux-based experiments in our research and teaching, DMAN is a response to our decision to introduce a professional production (as opposed to development) platform for the more mission-critical elements of our teaching work. In the following, we will discuss our system specification and implementation approach.

Reconciling Computing in Design

To explain how we identified task types in our domain requires a quick analysis of the general characteristics of design computation. Today, design and its education naturally embrace computers to a point which makes it unnecessary to discuss the requirement for computation in design. Here, we thus axiomatically presume a general need for computational infrastructures in design-related schools and do not further discuss this requirement as such. With respect for this requirement for computers, design does not differ from most other fields and disciplines. However, design computation differs from other computational tasks, such as communication, accounting, engineering or commerce in that it is aimed at solving open-ended problems. At the beginning of design processes, the final destinations of these processes are unknown. This is not the case in office environments such as insurance companies, banks or travel agencies where the nature of electronic transactions and the criteria of their validity are well-defined, known (closed-ended) and mostly static.

Computational facilities can quite easily be specified to support the distinctive set of operations required by these transactions and are often commonly available off-the-shelf. But in design, where no such problem-definitions can be given, computation and networking tools must be designed in open ways to support as many types of transactions as possible.

Tool openness, however, typically comes at high costs in form of user-unfriendliness and harsh learning curves (this is easily understood by comparing any email-client with a C-compiler or a jukebox with a piano). The design of a server and networking architecture cannot change the fact that designers need to learn and know how to use the tools they want to use. This difference between open and closed problem orientation is an essential difference between corporate computer networks and academic design computation. Applying their standard thinking and recipes, commercial system integrators which are most likely not aware of this difference, might easily struggle and need early briefing accordingly.

A psychological dimension of this implementation challenge is dealing with the great expectations a high-priced unified design computational foundation triggers even amongst faculty staff. What is difficult (such as thinking or learning to use a particular tool) remains difficult. What is inconvenient (such as data backup or secure file-sharing) however, can be made very convenient, efficient and safe. We used exactly this line between individual designer activity and the general requirement for (closed-ended) convenience and security to find the lowest common denominator of needs and to put different weights on the role of workstations and servers in our system.

Fat Workstations or. Thin Clients?

Since the early days of digital computation, we have observed a continuous oscillation between centralized (mainframe-based) and decentralized (desktop-based) system designs. In the old days, very few highly expensive computers had to serve great numbers of users. The advent of time-sharing did not change this paradigm as terminals were 'dumb' and computational power still resided in central mainframes. When terminals eventually grew "smarter", the centralized organization of CPU power was enhanced by modest decentralized functionality. This situation has dramatically changed with the introduction of the Personal Computer, which has brought strong CPU's to everybody's desks. But when PCs were networked, centralized support from servers, working in the background, became available again. The Internet and technologies evoked by the Internet such as Java have inspired a partial return to "thin clients". This de-centralization was re-emphasized recently by the advent of new types of mobile computing hardware. Generally, none of both paradigms is better and to be preferred over the other as both have a number of specific advantages and disadvantages. However, given a certain usage profile, they can make more or less sense and serve more or less well.

As there is little of a "uniform usage pattern" to be expected to emerge in a school of design, as mentioned above, we have decided to develop a hybrid solution. This solution partly capitalizes on resource centralization and partly decentralizes resources on users' desktops, maximizing the benefits and minimizing the disadvantages of both paradigms.

In this system (DMAN) storage, high-speed network connectivity, backup facilities, print server, Web server, FTP server, file system server and video streaming server are all provided centrally. This particularly involves massive data storage. Individual design tools (software packages, special-purpose hardware) and the CPU power they require are distributed over the school network and desktops according to individual needs. Workstations are generally fast and run problem-centred software tools but do not provide standard services and extensive, fast-growing storage space. The back-end server farm provides scalable storage space, standard services but no specialized functionality for individual design requirements.

User File System Design

The philosophy outlined above is reflected in the user home directory structure which represents the primary user interface to the system. Connecting a (Unix, Windows or Macintosh) computer to the school's network (even from other campus locations) automatically mounts the file system shown in figure 1.

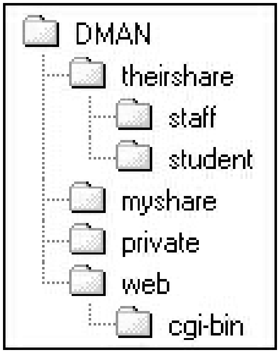

Figure 1. The personal file system on DMAN.

The file system (looking for example like an extra hard disk on Windows) provides easy-to-use and safe means for storing personal data, share data and publish data on the web. A directory named ‘private’ provides personal read/write access but is accessible only to the owner. Contained data is thus protected from unauthorised access. Data stored in folder ‘myshare’ can be read by all other users in the department through the directory ‘theirshare’, which – divided into staff and student subdirectories – link to all personal ‘myshare folders. The folder ‘web’ is connected to DMAN’s Web server and everything stored here is accessible via http://people.sd. polyu. edu. hk/~username. ‘cgi-bin’ stores Web-executables for dynamic HTTP content (Java servlets, Perl, Python, PHP etc.). This personal directory structure can be accessed from anywhere through the Internet, using the FTP protocol. It was the primary starting point we took from the user’s perspective for the entire system design in order not to compromise our goal of maximum ease of use. The planning and implementation descriptions below ought to be understood as a strategy to accomplish this top-down goal.

System Architecture: An Overview

Our network and server system DMAN was installed from scratch in a new office area our department moved into in mid-2001. The project comprises a dedicated climate-controlled server room, an integration of a Storage Area Network (SAN) and a Gigabit Ethernet Network as well as application services and custom-engineered networking- data- and user- administration tools. DMAN is based on dynamic fail-over mechanisms with redundant components taking up operations in case of system failures. A tape library allows a full user data backup every night. Moreover, a comprehensive warning and alarm system (system operations, climate control, water leakage, fire warning etc.) has been installed. The core of the system comprises a multi-platform server cluster providing the centralized part of the system functionality discussed above. It is connected to the decentralised part, a highly heterogeneous client landscape, spread throughout the department through a network which was also installed from scratch on a dedicated new range of IP addresses covering four class C subnets.

A Multi-Platform Server Cluster

The core of the server farm consists of multiple computing nodes working with legacy Windows operating system and a commercial 64-bit ready Unix kernel. Equipped with a dedicated power-backup and LAN-free data backup, this setup can guarantee the high availability our departmental users demand.

Figure 2. DMAN server room at the School of Design.

Figure 3 gives an architectural overview of DMAN. The system room, shown in figure 2, was planned and constructed as part of the design and construction of a new departmental office space. Apart from the actual computational equipment, the server room is equipped with a raised floor, climate control, a security and alarm system with access control and closed circuit video which were all planned and installed as integrated parts of the system.

System Planning and Implementation

A total of 25 university staff members were involved in the system planning phase from various departments including Information Technology Services, Estates Office and Finance Office. Five staff constituted the core team on the university side and a professional integrator which designated eight staff to the project plus sub-contractors implemented the system. The total of DMAN's development time, including the initial proposal, planning, specification, implementation and testing phases was 1.5 years.

Continuous Specification Adjustments

This time span (proposal, planning, specification, implementation and testing) exceeds the hardware and software industries' innovation cycles by far. A risk projects of this magnitude therefore face is potentially to end up with outdated technology before commissioning. In order to minimize this risk, we have monitored the market and carefully adjusted the DMAN specification throughout the entire development and implementation process wherever appropriate. As a result, DMAN components have been upgraded to later and more powerful components throughout the implementation period and agreements have been made based on according agreements with the integrator.

One typical example of a late adjustment was the installation of a so-called storage area networking software (SAN). For system availability and failover performance, the two SUN servers were set up to access one single storage file system simultaneously via a SAN. At the time of implementation, the respective piece of software did however not support a user storage quota system – a feature that is normally taken for granted in all UNIX environments. As later versions of the software support a quota system, an agreement with the vendor has been made for a free advance license assertion and free upgrade support in return for providing a promotional case study.

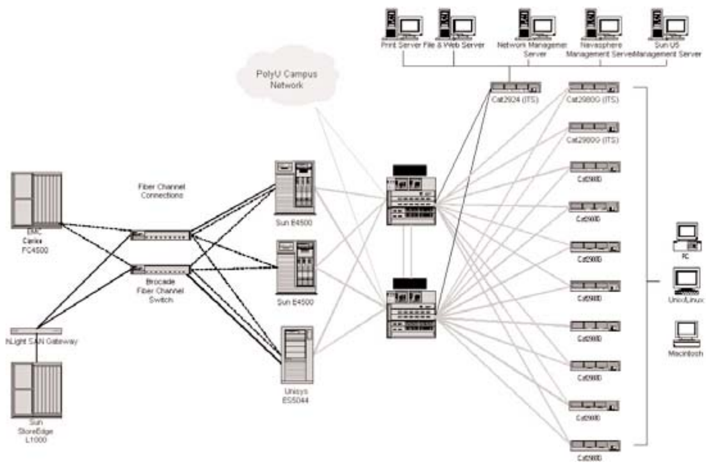

Figure 3. DMAN system architecture.

Soft Storage Quota

A particular challenge of (not only student) design computation is that the potential amount of data involved is barely predictable. Sound, video and animation data volumes to be handled during usage peaks (e. g. before student project submission deadlines) can be immense. Being aware of this particularity, and given the temporary lack of a kernel-based quota system support described above, we have decided to implement a custom-engineered *intelligent soft quota system*. This system allows individuals to momentarily exceed their allocated storage space if required. The development of this feature, a joint effort of the school of design and the system integrator, contributes significantly to temporary storage space availability and thus to student design quality and general departmental computation flexibility.

This system monitors storage usage in a defined frequency (e. g. every 2 hours) and maintains a simple report in form of a text file in the user's home directory. It notifies users via email if personal storage space has been exceeded. If this is the case for an individual for too long, further write access to the directory will be withdrawn automatically while data deletion is still possible. Figure 4 gives a statistic overview of the storage space usage. During end-of-term, student storage space overflow increases significantly. During this period, an average of 50 students use (in part significantly) more storage space than they "own". However, the disk usage chart shows that during this period, the disk usage remains at a stable and secure level because the soft quota notification system efficiently suppresses unhealthy global file system overflows. As a result, while a momentary exceeding of individual designer's storage space is allowed and automatically managed, the overall file system usage remains at a static and healthy level at all times similar to the operation under a traditional hard quota system (see figure 4). Once traditional hard-quota support is available, it will be set to protect the system from overflowing but to an individual level that is higher than the allowed individual storage space which will still be monitored by our highly useful soft quota system.

Implementation Timing, System Testing and Launch

As the final project phases, the network and data migration and testing phases were carefully scheduled to hand the system over for general departmental use at a time (mid-term) when getting famil-

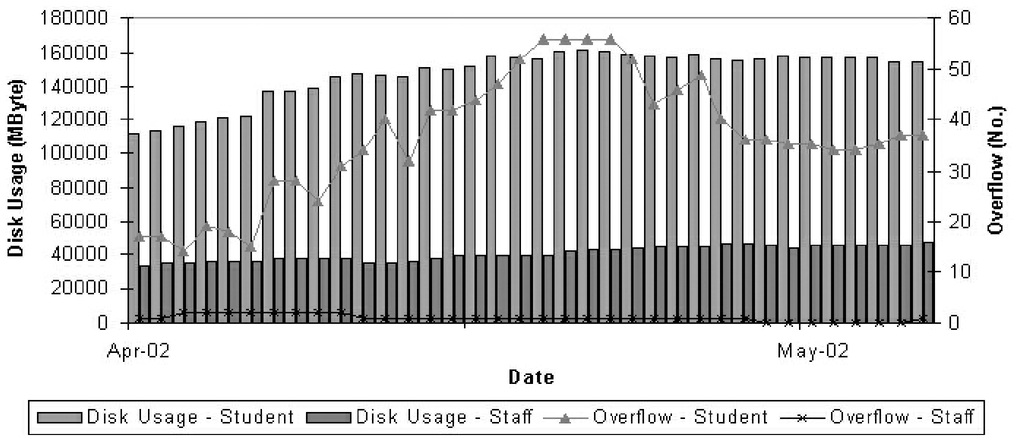

Figure 4. DMAN storage usage.

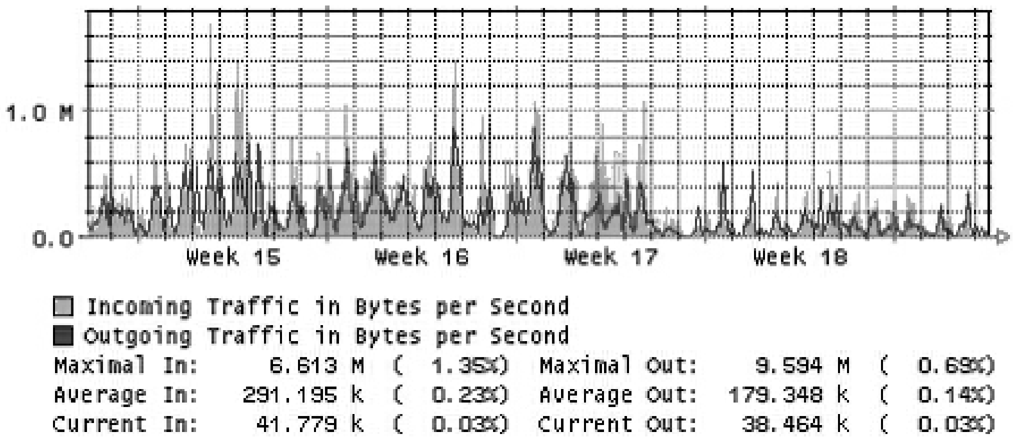

Figure 5. DMAN traffic statistics during end of term

iar with the new environment did not interfere with other important events such as student examination.

As under the open conditions of design computation, especially in large user groups, where usage patterns are neither static nor easily predictable, system testing is of particular importance. Under well-defined, closed conditions, system integrators can limit testing procedures to easily defined sets of transactions and requirements: Complete migration of previously existing data, number of possible simultaneous network connections, correct fail over behaviour etc. In addition to this testing from the system perspective, we had to observe and analyse a second, user-oriented real-life usage scenario for one month. As a testing group, the School of Design's Design Research Technology Centre (DTRC) was chosen. This research group comprises a core of 20 design researchers who used the system on a day-by-day basis as "normal users", who, being a computational expert group, were able not only to identify problems but also to provide highly competent suggestions for tackling them. Even after this initial testing phase, the system requires close monitoring and tuning to comply with the immense CPU, network and storage loads especially during the work-intensive phase at the end of term.. Figure 5 shows the difference in average network load during term (until week 17) and term break (week 18 and later).

Budgeting

The budgeting of a computer and network system of this scale comprises a large and variable number of different items, some of which might be easily overlooked during the early budgeting stages. Here, we give an overview of DMAN's budgeting (figure 6) as a reference and orientation for other, similar proj-

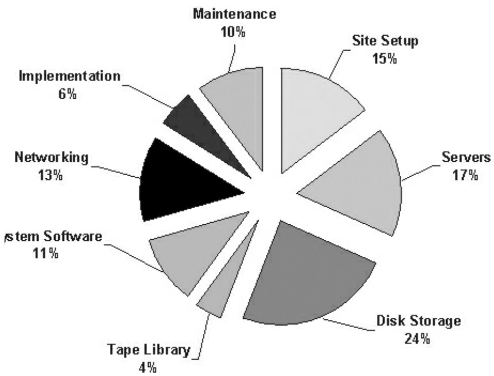

Figure 6. DMAN budgeting

ects. Note that these numbers are based on a limited (office hours-based) service coverage and a largely in-house, faculty-made system specification. Different arrangements might change this chart accordingly.

Conclusion

With the implementation of the DMAN project, we demonstrated that large-scale computing infrastructure projects can be successfully developed according to a vision of performance and ease of use, without initial regard to specific hardware and operating system architectures even under the open conditions of design computation. By placing our end-user concerns first, and having the flexibility to extend the project to ensure that these were addressed by the solutions found, DMAN has materialized as a uniquely versatile, scalable and powerful computing platform that has rapidly transformed the School of Design's ability to engage computing technology within the education process.

Students have gone from having to pack work on optical disks to having 600MB each of online server space; faculty and researchers have gone from 10MB of unprotected user space on a previous file server to a full 2GB with comprehensive file-sharing, security and online publishing facilities on a system that allows us to scale these figures up easily in the future. We have also developed 'smart' facilities that allow a new flexibility in storage space allocation which is expected to soon be reflected in future student design output.

Acknowledgements

The authors gratefully acknowledge the support from academic and research staff and students of the School of Design, the Design Technology Research Centre (especially for their support as test users), the Information Technology Services Office, especially its director Richard Li, the Estates Office and the Finance Office of The Hong Kong Polytechnic University as well as UNISYS China/Hong Kong and VERITAS Software Hong Kong. The DMAN project was funded by The Polytechnic University of Hong Kong's Earmarked Provision for Special Purposes.

References

Ceccato, C., Falk, L. and Fischer, T.: 2000, The Silk Road: An Interactive Online Encyclopedia as a Foundation for Networked Education in Design. In: Kos, Jose Ripper (ed.): SIGRADI2000 Conference Proceedings, Facultad de Arquitectura – Universidad Nacional de Mar del Plata, Rio de Janeiro, Brazil 2000.

Falk, L., Ceccato, C., Hu, C., Wong, P. and Fischer, T.: 2000, Towards a Networked Education in Design. In: Beng-Tiang Tan et. al (eds.): CAADRIA2000. Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia, Singapore, pp. 157-167.

Thomas F., Burry, M. and Woodbury, R.: 2000, Object-Oriented Modelling in Computer-Aided Architectural and Educational CAD. In: Beng-Tiang Tan et. al (ed.): CAADRIA2000. Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia, Singapore 2000; p. 145-155

Fischer, T., Herr, C. M. and Ceccato, C.: 2000, Towards Real Time Interaction Visualization in NED. In: Clayton, M. J.: ACADIA2000 Conference Proceedings, Catholic University, Washington D. C. 2000.

Frazer, J. H.: 1995, An Evolutionary Architecture. The Architectural Association, London

Revett, M., Boyd, I. and Stephens, C.: Network Computing: A Tutorial Review. IEE Electron. Commun. Eng. J., February 2001, 13, (1), pp. 5-15

Rittel, W. J. Horst and Webber, M.: 1973, Dilemmas in a General Theory of Planning, DMG-DRS Journal, vol. 8, no. 1, pp. 31-39, reprinted from Policy Sciences, vol. 4, no. 2.

Schmitt, G.: 1993, Architecture et Machina: Computer aided Architectural Design und Virtuelle Architektur, Verlag Friedrich Vierweg, Braunschweig and Wiesbaden.