Colama^{TM} Univ Edition, an introduction.

Summary : This document introduces *Colama^{TM} - Univ Edition*, a virtual lab management suite from Coriolis Technologies Pvt. Ltd. designed specifically for the Indian university environment.

Motivation

Having been through the Indian engineering education system, we were acutely aware of the shortfalls the lack of adequate laboratory facilities lead to. While the Internet brought to bear advances in technology to the field of web based education, we still lacked a scalable laboratory environment that was affordable. We, at Coriolis Technologies, sought to bridge that gap.

Background

With building a scalable and affordable lab environment as our objective, we explored a little to understand better the related issues involved. It quickly became apparent that for our efforts to be truly useful we needed to address the following -

- the learning experience for students
- enable teachers to benefit from technology
- lower the total cost of ownership for a lab
- security, compliance and liability related issues

Let's now take a closer look at the four areas we identified earlier.

- **the learning experience for students** - now what difference could one possibly want you'd be asking yourself. For starters, here's a list we wish we had when we were students.
    a. Multiple private machines - given the proliferation of networking, computer education cannot be deemed complete without one having gained some degree of familiarity with it. How better to gain that experience than by writing a few client-server applications and testing them out? What if, we could provide student with machines to test and learn all this? Too, expensive right? Wrong! With the technology available today we can do this without having to buy a cart load of additional computers...
    b. Pause/resume/roll-back sessions - The lab class is getting over in 5 minutes, but my experiment is going to take a while longer ... Well, what if we could pause the computer, much like we would a DVD movie while dinner is being served, and resume it at the next class? and do that without interfering with intervening classes?
    c. Collaborate better with staff - Every time we need our teacher to help us out, what if we could seek it without disrupting the whole class? what if we could privately converse/chat with the faculty to resolve doubts while still in class?
    d. Improved access to specialized software - Remember having to wait in line to get access to the printer and how networking solved it? What if we can do that today for access to specialized software that is too expensive for all the labs to own? and yes, do that legally?
    e. Potential for industry interaction - While the objective of an engineering education is to prepare us for the real world out there, today, we don't really see any real interaction between academia and industry in a scalable model, do we? Wouldn't it be nice if we could provide a platform for industry and academia to interact?

f. Remote access to labs - While this isn't one of those things we could have dreamt of having in our days, given the proliferation of computers and mobile devices any modern day solution should support it. What if we can do that seamlessly and create "openLabs" no longer confined to a single classroom?

- **enable teachers to benefit from technology** -
  a. Course content sharing - Given the paucity of good teachers, wouldn't it be great to enable a teacher to reach out beyond the four walls of a classroom in a scalable way?
  b. Sharing student consoles - Wouldn't it be nice to be able to see and interact if need be with a student via his/her computer console without having to walk over?
  c. Demo mode - Wouldn't it be nice to be able to demonstrate something clever a student has done without the need for a projector? Wouldn't it be nice to share an example with every student?
  d. Inspect live assignments - Rather than pore over assignment report, wouldn't it be nice to simply run the assignment and try it out yourself?
  e. Detailed audit reports - Is there a point to having faculty take attendance at a lab? Wouldn't it be simple enough to generate a report automatically based on activity?
- **lower the total cost of ownership for a lab** - Money has always been in short supply, and especially so when it comes to spending in the education sector. Given the estimated shortfall for trained manpower in the years to come, every penny saved on this front can be reinvested to bridge the gap. When it comes to running a lab, and accounting for the costs we need to consider the following heads of expenditure -
  - Capital expenses - They are the one time expenses on initial setup and perhaps on upgrades.
    i. Fungible labs, re-purpose on demand - If labs can re-purposed between classes, one can potentially save real estate investments by creating fewer, but better utilized labs.
    ii. Substitute desktops by thin clients - Thin clients are cheaper that desktops and last longer between upgrades.
    iii. Save on test machines - We could create/provide test machines on demand and provide for test machines for student use as and when they need them without necessarily having to buy more hardware.
    iv. Smaller backup power supplies - We could make do with much smaller generator and/or battery backups if we used thin clients instead of desktops.
    v. Smaller outlays on cooling equipment - Air-conditioning needs would go down to about 20% of the capacity required for a traditional lab.
    vi. Reduced storage requirements - Advances in technology can be harnessed to dramatically reduce storage requirements, from about 80GB per terminal to about 100MB per seat.
  - Operational expenses - These typically are meant to cover the recurring expenses incurred to keep a lab operational.
    i. Power consumption - About 80% of the power consumption in a traditional lab in a University environment is wasteful and can be saved ... about Rs.18/- per terminal per day can be saved. Interesting, isn't it?
    ii. Cooling costs - What holds for power, pretty much holds for cooling costs as well. Just that the savings involved are roughly 30% more than what you would save on power.
    iii. Maintenance charges - Annual maintenance contracts (or AMC) constitute a major expense in running a computer lab. By reducing the overall outlay on the lab we can reduce these expenses significantly as well.
    iv. Better utilization of expensive software - Once expensive software is acquired, it behooves us to increase its utilization to the maximum.
    v. Real estate - One typically does not expect to see this under operational expenses, but one should consider rent on lab space as an operational expense and being able to reduce the number of labs would go a long way towards lowering expenses.
  - Administrative costs - While often considered as part of recurrent expenses and classified under labour, we think this constitutes a significant outflow in a computer lab and deserves to be looked at separately. We've listed below some capabilities that would ease the daily chores of a lab administrator.
    i. Simplified patch management - The thought of having to patch the 400 computers in the department for a security vulnerability is daunting. Having to do that 3 or 4 times a quarter is a real problem that can be avoided with judicious use of technology and some

processes or best practices. What if we could do that just once and it can be replicated across every computer in the department?

ii. Powerful search capabilities - By the time we have upwards of few dozen computers, it becomes difficult to locate individual pieces with specific capabilities. What if we could help locate a machine running Windows XP with say AutoCad installed? How about you being able to do that just as simply as you'd use Google today to search for information on the Net?

iii. Reduce preparation time for examinations - Lab administrators dread having to re-install every machine that is going to be used for a board exam. In theory, one should be doing it before every practical session during the exam. What if, one could re-installation a machine in 1 minute flat? Now, what if we could reinstall an entire lab ... 0 to 60 in under 5 secs?

iv. Automated reports - How often have you heard excuses like "the machine was down" from students? How often have you heard the complaint "the machines are too slow"? Have you ever been able to verify the fact? What if you could have automated online reports that you could refer to whenever you needed them?

v. Easy roll-backs - How often has the fix for one problem created another one that was worse? How often have we done something inadvertently and messed things up? What if we could simply roll-back time? If you would settle for being able to that with just your computer, we might just be able to do it ...

vi. Simplified user management - Spare a thought for the lab administrator who has to create and delete hundreds of user account every semester. Wouldn't it be nice to be able to easy this chore?

- **security, compliance and liability related issues** - For better or for worse, we live in times that aren't safe and that is true about the digital world around as well.
  a. The Indian IT Act - Like it or not, any compute infrastructure we setup today has security implications that we need to be aware of to effectively deal with. And the Indian IT Act, irrespective of its intentions, is draconian in as much as it presumes guilt until proved otherwise. The owner of the computer is assumed responsible for any/all activity that originates at a computer. That said, do you know what the computers in your lab are being used for? Would you be able to vouch for that?
  b. Private workspaces and access control - What if we could assign a private computer to every user and ensure only that user has access to it? It then becomes trivial to tie accountability for its use to the end user as well, doesn't it? This single capability precludes the need to provide shared resources that can be misused with impunity.
  c. Infrastructure security - How many individual pieces of equipment strewn around your campus network can you reliably protect in an affordable manner? What if we could centralize all important pieces of equipment and provide controlled access to their capabilities from across your campus network?
  d. Detailed audit reports - Wouldn't it be nice to be able to audit the goings on in your computer lab? Who all were using your lab on Sept 26th, 2008? Especially when you have an inquiry regarding misuse in progress?
  e. Image inspection - In a world that is perfect, we'd still want to be able to locate a computer that has say Office 2007 installed, in order to be able to view a .pptx presentation someone sent us. In the real world, we might also be interested in knowing what software has been installed on a given computer, especially if it is unauthorized.
  f. Demonstrable compliance - Wouldn't it be nice if one could prove compliance in the event of a software assets audit? Wouldn't it be nice if we could get reports on actual usage statistics as opposed to predicted requirement in order to better allocate finances?
  g. Controlled peripherals usage - How often have you rued the day someone introduced malware into your network via an USB thumb drive? What would the capability to control/block access to these USB ports be worth to you?
  h. Reduced carbon footprint - Ok, the Himalayas will still be snow capped in 2035, but given that 300 computers left running for 8 hours every working day, can power 500 homes in India's villages, can you afford not to make every effort to reduce your carbon footprint? and what if you can do so at no extra cost? shouldn't it be criminal not to?

Design considerations

Given economic realities, it was imperative that every design decision we took ensured that we brought to bear the best technologies to provide state of the art capabilities at the cheapest costs possible - in effect we set out to build the *Nano* for computer lab management.

Going by our market's reluctance to willy nilly change an existing hardware setup, we decided that whatever we designed should fit into an existing University computer lab with the minimum of disruption. To that end we decided our solution should work on pre-existing lab equipment as far as possible and support a gradual cross over to newer hardware technology.

Considering the proliferation of web based technologies in today's world, we decided that our solution will only support browser based end user interactions.

Server virtualization as a technology has matured to the point it is now part of the mainstream compute environment. It was an easy pick to build our solution around virtualization, but a lot harder to remain vendor neutral with respect to the virtualization technology.

Colama Architecture

The *Colama™ - Univ Edition* platform has the following logical components.

1. The Colama *server* which
    a. maintains a repository of all virtual machines with appropriate annotations
    b. provides a web based interface for users to interact with the platform
    c. orchestrates on demand provisioning of virtual machines
2. A pool of *managed hosts* which
    a. are used to run the virtual machines as per user needs
3. A collection of thin clients which
    a. serve as end points for users to interact with their virtual environment via a web browser
    b. At some installations, managed hosts also double up as thin clients.

In keeping with our design considerations, we support very different mappings of the logical components enumerated above to actual physical entities subject to the following constraints.

1. The colama server should have sufficient disk space to house the repository, preferably configured in a mirrored configuration to ensure resilience, a moderately powerful CPU with about 1GB of RAM.
2. The managed hosts should have sufficient compute power, be preferably hardware virtualization enabled, configured with striped storage to increase i/o throughput and have large memories to accomodate multiple virtual machines without swapping. support for wake-on-lan would be preferable, but not essential.
3. The thin clients need to be able to run Firefox 3.5 or above, JRE 1.5 or above, support flash and other similar audio video plugins, good graphics capabilities and 256MB of RAM or better.

As can be expected, we do support a wide variety of mappings of logical to physical components that satisfy the constraints called out.

1. Our preferred way to approach a green field deployment is to map each of the logical components to corresponding physical components.
2. Our alternate approach is to map the colama server and a thin client to an existing physical machine, and use all other available machines to map a combination of a managed host and a thin client each.
3. Where the existing lab equipment is too old to be used as managed hosts, we support a hybrid approach where we recommend mapping the colama server and managed hosts onto a small set of new machines and using the existing set of machines to map thin clients onto.

On the software side, our objective has been to provide the end user with a seamless experience while interacting with a collection of separately sourced components. To that end, Colama™ - Univ Edition provides a web based interface supporting a number of workflows to manage and interact with the virtual infrastructure at a lab.

Users connect to the Colama server via a web browser, typically on a thin client to post requests. We currently require the user to have Firefox and a few plugins installed.
The Colama server runs a typical LAMP stack and services individual user requests.
Each managed host runs Linux/KVM/Qemu and an agent that interacts with Colama server via a proprietary protocol.
On demand, virtual machines are run on these managed hosts and their consoles/screens made available, via a browser interface, to the end-user on a thin client.

Does Colama™ - Univ Edition meet your lab's technical requirements?

From the compatibility perspective we've tested the system out extensively at two leading engineering colleges and have verified that the following oft used platforms and software packages run without issues.

1. Windows (NT, XP, Vista) 32bit & 64 bit versions where applicable
2. Linux (CentOS, Ubuntu) 32bit & 64 bit versions
3. TurboC, Borland C packages on Windows NT
4. Rational Rose
5. Oracle Clients
6. Visual Studio on Windows XP, Vista
7. Ecllipse on Windows XP
8. MathLab on Windows XP

User roles and capabilities

The *Colama™ - Univ Edition* platform supports 3 categories of users.

1. Administrators - who are primarily responsible for
   a. creating and maintaining approved versions (golden images) of virtual machines for other to use
   b. managing (creation/deletion/modification of) user accounts
   c. patch management for the various golden images required by the users

Maintaining servers

2. Faculty - who, in addition to the privileges enjoyed by students, can
   a. also attach to consoles of student machines to collaborate/oversee more efficiently

    b. tag any VM instance for a *demo*

3. Students - who can
    a. deploy specific virtual machines of interest to them for work
    b. preserve/retrieve versions of their individual machines at will
    c. reset their individual machines to any previously stored state
    d. share their consoles with their teachers whenever required.