Capital Markets Trading Floors, Current Practice

Sam Lipson

Capital Markets Trading Floors, Current Practice

Sam Lipson

ABSTRACT

Financial trading has been described as "technological warfare". Whether or not you believe the metaphor, trading systems certainly involve a tremendous need for highly reliable, interruption free, real-time data.

Real-time, in this case, means presented accurately and without delay. Current practice of medium to large trading floors often involves Unix based workstations and significant attention to reliability. There are issues to be addressed during the construction of the physical facility, the planning and installation of the systems, and various rules of the road for support personnel. This paper attempts to address many of these areas.

In this paper some of today's common practices and the rationale behind them will be described. It is the goal to take some of the mystery of out the black art of trading floor design and support.

The ideas in this paper have been installed on several of the largest trading floors in Boston, MA (USA) covering the market segments of Foreign Exchange (F/X), derivatives, equities, fixed income, and (US) government securities. The only market segment not represented is commodities.

Introduction

In order to understand some of the constraints placed upon trading systems it is useful to understand a small amount of how capital markets work (capital as in money, or currency). In each transaction there is a buyer and a seller, like any auction system the seller(s) present their merchandise asking for a certain price, and the buyer(s) bid the amount they are willing to pay.

Eventually, a price asked for (by a seller), and the bid (from a buyer) meet, and a transaction is consummated.

A market consists of a number of buyers and sellers, all of whose primary motivation is to make a profit.

The real-time requirements come from several distinct areas. One is the need to give correct prices to customers – quote a wrong price, and you may lose a customer or run into regulatory difficulties; another is to drive analytics or proprietary trading applications. There are also some styles of trading called momentum investing, and arbitrage – which rely on nearly instantaneous execution of trading decisions based on market motion or short-lived inefficiencies. Getting your trade in first is often the difference between making a profit and being out of the game entirely.

Today's capital markets operate on a tremendous scale. The foreign exchange market alone (foreign currency buying and selling) is approximately 1 Trillion dollars per day. F/X is often speculative, as only about 10-15% of the F/X transactions per day are actually required for commerce (i.e., import/export).

We build trading systems, which normally involve Unix workstations on the desk (in our case Sun SPARC, though other platforms are used in this application), real-time data feeds, and significant attention to reliability throughout the design.

Proper reliability design in a commercial environment (and specifically financial services) – means that you have to understand the cost of a failure. Once you understand the cost of a failure, and the frequency or likelihood of that failure, then you can make a business case for spending money (or effort, which later equates to money) in order to mitigate the effects of such a failure. The trading community makes their living by taking and controlling risk, and hence can often provide more than just budgetary guidance as to whether additional expenditures are justified.

In many parts of this paper we refer to the cost: either budgetary or business impact. What we are building here is a business infrastructure. Ideally, the infrastructure for a profitable business. As in any business, our profits (trades) need to amortize the cost of what we do. Since we are building infrastructure for very sophisticated financial people, we constantly ask for expert advice on matters of expenditure, risk/reward, and business impact.

You will see examples of where we apply systems, software or hardware redundancy in ways which may seem somewhat archaic, especially in an era where systems with internal redundancy are available. Once again, there is a cost/benefits trade-off. By spending additional money on the types of external redundancy we describe, it is often possible to get many of the benefits traditionally attributed to open systems. While there are internally redundant open systems on the market today (e.g., Stratus FTX), their use is not as widespread as the systems discussed here.

It is hard to pinpoint a specific reason for this, but it is a fact of life that it is often easier to grow a small system into a big one, than it is to build a big one from scratch. Financial people tend to like to make a small "trial" investment, and then grow it if it proves fruitful. This is yet another vote for "scalable" technology. Also, our Wall St. colleagues often have little patience for technology risk, and instead like to save their "risk exposure" for the financial domain.

A fair amount of the technology presented here would not be considered on the leading edge. Nonetheless, it has been chosen for it's proven utility, economics, and sometimes the sheer availability of skilled people for it's care and feeding. While many parts of the Wall St. community thrive on being at or beyond the state of the art, the bulk of the trading community focuses more on building highly reliable systems with a minimum of built-in obsolescence.

There are a multitude of possible decisions. Many are focused by the size of the installation, or the budget available, and sometimes there are corporate standards, or management guidelines. Hopefully, in each case the principles that guide your hand have sound business or technical factors behind them.

Divergent Routing of External Connections

Our trading systems revolve around a number of data feeds. These are commonly leased lines (though other techniques exist, including satellite and radio receivers), through which various data vendors and exchanges transmit market data. A simple example is presented by the equities market where the bid and ask prices for a stock, such as IBM, from the New York Stock Exchange (NYSE) are transmitted in real-time via a data feed.

Because it is absolutely essential to provide this data with near 100% reliability, the main feed link is always backed up by a secondary. We divergently route the primary and secondary links (i.e., they pass through disparate cables, telephone offices, etc.).

As an example, an installation in downtown Boston – there are two telephone company central offices which serve our building (this is fairly common in medium to large cities), Franklin (St.) and Harrison (St.), as well as a variety of leased line providers – ATT, NYNEX, MFS, Teleport, etc. This is in addition to radio and satellite receivers, which can also be used to receive data feeds.

If we bring in our primary leased line from the Franklin St. central office, we will likely back it up with a dial-up link, served from the Harrison office. The second set has the reverse configuration.

We generally ask for, and, if necessary, pay for a separate telephone cable entry point to the building which comes from the second Telco central office. It is surprising how often street repairs cause cable breaks. A cable break on the only feeder cable (or both cables which run side by side) into our building can easily knock out trading for a very long time even days.

Feed Redundancy

There is a distinct possibility that we will experience an equipment failure in the feed system, to mitigate this we duplicate all the feed equipment and insure that we have divergent routing – perhaps to another office of the feed vendor (i.e., no single failure takes down our data source). Depending on your business needs (i.e., risk tolerance and budget) the "secondary" system can be used to provide half of your capacity. You would then operate with lessened capacity in event of a failure – this saves some money, and creates some risk. Once again the business people help guide your hand.

The Teknekron Software System that is uses on our example floors does an automatic roll-over when one feed machine dies (and has been somewhat augmented to provide automatic failure notification to the system staff).

If a particular feed dies during the day, the roll-over occurs automatically, and the traders continue to see real-time data, as if nothing happened. A real sign of success is that the traders never notice these infrastructure failures. However, even though data is still available, it is important to restore our failed feed as soon as possible. I like to use the metaphor: "When one of your legs is broken you do not necessarily have a mobility failure, however it is extremely important to repair it ASAP. Especially in case the other leg breaks."

The failure could be caused by any number of things, but it is not uncommon to have a leased line failure (repeat, not uncommon). If this is the case, it is usually a simple matter to go to a dial-up modem, backup solution (hence dial backup).

Checkerboarding

If you take a look into the trading floor infrastructure you will see that the floor is laid out, at minimum, in a checkerboard fashion. Take a map of trading positions the floor (grouped by business unit is best) and color alternate desks light and dark. The dark colored squares are served by one complete set of equipment and links, and the light squares by another (including cabling, power, hubs, servers, links and as much else as is within your budget and/or practical).

In the case of a catastrophic failure which takes down an entire color of squares, those traders can always look on with their neighbors. We're particularly fond of reminding the telephone group that long phone (handset) cords are part of our backup strategy. If a trader's workstation goes down then they can often continue to trade as long as their phone reaches to the next desk.

Note – it can be quite difficult to avoid all single points of failure – often you cannot get building management (in leased facilities) to insure that your cable runs (from the telephone company points of entry) come through different risers in the building (risk – fire/damage in the riser). There may be only one UPS, or one elevator, perhaps only one entry door. The failure of any of these things can make it difficult, if not impossible to trade.

Keep in mind also, that through floor expansion you will want to maintain your checkerboard layout. It's not always as easy at it may look at first glance. Our best advice: layout the entire checkerboard at the beginning and fill expansions into the original scheme.

Market Data

One of the primary types of information that traders base trading decisions on is market data. Market data consists of real-time prices, news and analytics driven by both intra-day and historical data. The sources and delivery mechanisms for this data are critical to both it's accuracy and reliability. In order to provide the configuration best suited to the business needs it is often necessary to understand the a number of aspects of market data: sources, costs, and data formats.

The primary sources of market data, for exchange traded instruments such as equities, are the exchanges themselves. For the over-the-counter traded markets of foreign exchange and money market instruments. The primary sources are dealers and inter-dealer brokers.

Large institutions may bring direct exchange feeds in-house. However, most firms contract for the services of a market data provider. Market data providers act as resellers of market data, consolidating the data from many sources, and re-distributing the data based on an institution's interest. The market data providers collect information from exchanges, dealers, inter-dealer brokers, inside sources and news services, and deliver a consolidated subset of this data to trading firms via digital data feeds.

The feed handler machines are on their own private subnet (feed network) because the universe of information available from our data providers is so large, and also to provide a measure of isolation. The feed vendors provide catalogs of the data they have available, some of them are distinctly reminiscent of the Manhattan, NY telephone directories (i.e., large).

Different market data providers transmit data at differing speeds. While the current line bandwidth used by these providers runs from 9.6Kbps to 128Kbps, the majority of these feeds are delivered at 56Kbps for broadcast feeds, and 19.2Kbps for interactive feeds. Most market data providers transmit their feeds using some type of data-compression and/or encryption scheme.

The data providers charge a handsome sum for their services, and they have a vested interest in making certain that their subscribers receive only the data they contract for, as well as keeping their cost of doing business as low as possible. In many cases, the feed vendors have a regulatory incentive to protect their raw data feeds as they are specifically liable to exchanges for user fees (often based on the number and types of people looking at exchange data).

Market data provider feeds fall into three categories from a record format perspective: record-based (elementized), page-based, and hybrid. The management of each type of feed requires specialized software.

Page-based digital data feeds deliver data on a "screen-by-screen" basis. To maximize communications efficiency, pages are transmitted as compressed text segments with each message header denoting the screen coordinates at which the data should be displayed. Control sequences that determine display characteristics such as color can also be contained in these text segments. Consider the daily foreign currency quotes from a newspaper and you will have an example reminiscent of what traders are seeing updated on a real-time basis.

In order to maximize the efficiency over the limited data link bandwidth, only the part(s) of the page which are changed need to be transmitted. These deltas can be applied to the "base" page producing the current completely data display.

Page based digital feeds are a carry-over from the traditional "green screens" of the past – where "traders" would watch the markets on one formatted page of an old terminal (with green phosphor). Today, while there are still green screens installed (and, alas, new ones being installed even now) – various methods are used to provide the similar functionality.

Some of our vendors transmit the basic screen to our system when it is requested (or subscribed) by a user. This is the mark of an interactive feed, where the user requests are sent up to the vendors equipment (i.e., outside of your site). There are others that provide a broadcast feed where they send literally all of the information they have available (in a highly compressed form), and the local system(s) maintain all of the information at all times.

There are advantages and disadvantages to both methods. The interactive system tends to be a more complicated overall system, and new requests during periods of heavy activity may suffer a delay in processing.

The broadcast feeds have a simpler overall system design, but since there is a limited broadcast bandwidth, they often send base pages, and other "core" database information during periods of low activity. During the day, they are sending either price updates (whereupon the local equipment can expand those into formatted pages), or page updates. The disadvantage of this is failure recovery time. It often takes 48 or 72 hours to totally refresh the local database in your machine. Which is a maddeningly slow recovery from a system failure. Likewise, there can be a lag or lost data during periods of high market activity, as the broadcast bandwidth is limited. Since they are sending ALL the data they have that you may possibly request, some data of interest may be delayed or dropped.

While you might think that it is permissible to drop old prices for an instrument (say the asking price for IBM), if you don't have time to send them all down the line – often traders want to see ALL of the prices, even those they can't execute on. It's also potentially the difference between a price graph with a reasonable slope, and one which is dramatic.

In order to provide programmatic access to the data distributed via page-based feeds, it is necessary to apply an additional processing step, often called shredding. A shredder, or parser, will unformat a page breaking down each line of data into individual elements. Page shredding adds a small time delay before the data can be put to use. If the market data provider changes its page format, you will need to update your shredder. Unannounced page format changes are maddeningly common.

The product of page shredding is a record-based, or elementized, data set. This type of record format is suitable for use by other programs, where it can be used for trading and analytical applications. These elements could be stock prices, exchange rates, bond yields, currency quotes, news headlines, etc.

Record-based data feeds, also known as elementized feeds, deliver data according to a defined set of record formats. The main benefits of this type of feed are: no shredding is necessary; the data is presented in a known format regardless of its source; and updates can be processed without the delay, or potential hazard of shredding.

Real-time elementized market data is used as input to a number of decision support systems. These systems include real-time spreadsheets, for instance performing simple calculations such as the conversion of bond prices to yields; screening and analysis programs, which generate alerts based on certain events such a IBM trading above 103; and risk-management systems, which interface the real-time feed to a position management system.

In addition to real-time feeds, the typical trading floor will integrate numerous other data sources into its applications environment. These other data sources will supply data that, while still critical to the trading decision process, does not change as often. These data feeds may include batch downloads of holdings, historical databases and databases of SEC (or other regulatory body) filings by institutions. The data can also be delivered via FTP, over dial-up lines, or on magnetic media.

Data Distribution

Now that we understand the sources and record formats of market data, we must investigate what is done with it after it is delivered to the trading floor computer room. This brings us to the market data distribution plant.

At the heart of the market data distribution plant are the feed handlers and ticker plants. Feed handler software generally runs on a dedicated computer, sometimes with specialized hardware (i.e., high speed serial interface). The feed handler manages all communications with the market data provider. The feed handler is attached to the market data provider's system by means of a serial or dedicated LAN connection. It is attached to the feed network on the trading floor side.

Also sitting on the feed network, are the real-time database applications, typically called ticker plants. These ticker plants maintain a memory resident database of current values and subscribers. Current values are the latest values for a specific record or page transmitted on the feed. A current value might be the most recent image of Knight-Ridder page 45000, or the latest bid and ask prices for IBM. Subscribers are applications running on the traders' workstations, analytics, and sometimes the software which drives the large wall-displays which are so popular in trading rooms.

There are usually real-time database machines (which store, at minimum, the latest prices for various instruments), and these are also duplicated in (at least) a fault-tolerant pair. The real-time database machine is necessitated by the need to supply the latest quote for a particular symbol – essentially the instant it is subscribed to.

Since you must also cover the possibility of a machine failure the realtime database machines are duplicated. So, it is not a requirement to use disk mirroring or RAID.

The trader workstations reside on a trading network different from the feed network. This requires the ticker plant to be dual-homed to the feed network and the trading network.

Processing mechanisms for the feed handler vary according the the type of feed, page-based or record-based broadcast/interactive. The main difference between page-based and record-based feed handlers is in the methods they employ to request and process data. In a page-based feed, requests are submitted as they occur. This usually, but not exclusively, translates to a user "punching up" a given page on a trader workstation. Requests can also be transmitted when the feed handler is started as a means to immediately populate the database with the most frequently requested pages. The feed can also be told to mark these pages so that they are always fixed in cache. This is a method of ensuring that when the maximum page capacity is reached, these pages will not be swapped out to service new requests.

In a record-based feed, records may be selected via a filtering criteria which might include instrument type or exchange name (i.e., all NYSE stocks). These feeds operate at speeds equivalent to the delivery of hundreds or thousands of records per second. The This feed handler selects the required data and transmits in internal format onto the feed network. The values can then be inserted into the current value database of the ticker plant.

Recovery methods also differ between record-based and page-based feeds. As the page-based feed is interactive, refreshing the current value database is a matter of re-requesting pages. This is not an option with the record- based feeds. A record-based feed would need to receive a "refresh" (i.e., values for all symbols of interest), in order to re-populate the database with all instruments,

As with any other part of the trading system, the design of the market data distribution plant is required to include mechanisms for reliability and fault-tolerance. As a rule, for any real-time feed, we install a minimum of two feed handlers and two ticker plants. These four processes run on up to four different computers, are connected to the market data providers using divergent routing and back-up dial lines (or ISDN) using alternate facilities.

The software installed on the example floors, enables us to run the two feed handlers as a fault tolerant pair. We also run the ticker plants as a fault tolerant pair. As with the ticker plants, the feed handlers operate in primary or secondary mode and maintain this relationship by use of a time-stamped heartbeat. In addition to taking over as the primary feed handler (or ticker plant) upon failing to receive heartbeats within a time period, the new primary feed handler will re-broadcast the data it received since the last heartbeat was received. This assures that no data will be lost for the time needed to "fail-over".

On the client-side, between the ticker plant and the trader workstations, Teknekron generally delivers data to the desktop using UDP. A reliable delivery mechanism is built into the data portion of the packets which contains a sequence number. The UDP broadcast is forwarded to all IP subnets on which client workstations require access to the data. Cisco routers have an "IP-helper" feature which helps implement this.

Trader Workstations

The term trader workstation refers to the computer system that sits on a trader's desktop. The trader workstations we install on these floors are Sun Sparcstations; at present, running SunOS. The windowing system is the X-Window system with a virtual window manager. These trader workstations offer a cost-effective scalable platform; capable of real-time display and processing of graphics and compute-intensive applications.

This platform is a conservative choice, from the viewpoint of its being widely supported and proven. Yet it affords us the flexibility to take advantage of emerging operating system and network technologies.

Servers

In the "computer room" you will find the usual NFS, or other file servers; but, generally speaking, all the critical disks are either mirrored or RAID protected. In some installations, the file server itself is backed up by either a hot or cold spare. Depending on your needs, it is also possible to build redundancy into the CPUs, or file servers (using specialized hardware and/or software).

The "back-end" machines are also Sun SPARCS, running mostly SunOS, which follows our preference for proven and reliable technology. You may see occasional specialized servers running Solaris (e.g., SparcServer 1000/2000 used as a Sybase machine). However, in general we stick with technology which is proven and known reliable (at this writing, more significant Solaris transitions are taking place).

Network Topology and Cable Plant

Similar trade-offs to the decision regarding choice of trader workstations come into play when we design the network topology. We can't afford to put "inexperienced technology" into production. So, when it comes to the network we try to position ourselves to take advantage of emerging technologies without having to undertake a major infrastructure retrofit.

We start with this perspective and also the need to support the special network needs of trading applications, including redundancy and protection against single points of failure. We must also keep in mind the need for expansion, such as additional market data feed services, additional protocols, and even remote sites such as branch offices.

The example trading floors are currently using 10Mb Ethernet. It is extremely important to characterize the LAN traffic generated by trading applications including the heavy UDP broadcasts in order to determine an appropriate number of trader workstations per subnet/segment. This also helps when setting alarms for network utilization rising above/below the nominal band, which is often an indication of impending problems.

Shared Ethernet is also potential problem. While it is a technology that makes it easy for multiple machines to communicate, the latency of the network is impossible to predict. Some installations have multiple Ethernets running to each workstation in order to minimize delay and the potential that an unintented network "event" (i.e., user FTP of a large file) will block time-sensitive market data. One of the Ethernets is dedicated specifically to transmitting market data, which effects a hard partitioning of network bandwidth for real-time traffic.

We also wanted to protect the trader workstations from network problems that could be isolated to a specific subnet, such as a hub failure or a broadcast storm. To this end, we installed hubs (connected by routers or bridges – not wire or fiber) with redundant power supplies. This protects not only against a failure in a power supply but potentially a power circuit failure as well. If your device has redundant power, be sure to plug each power cord into a different power source, or at least a different circuit.

Hubs are often given short-shrift by networking people, as they are "uninteresting" devices. However, to us the failure of a hub can take down a large number of traders – so we treat them with respect.

The cable runs to the trader workstation, consist of home run Cat. 5 UTP – which we consider fairly future-ready. No two adjacent desks are connected to the same segment. In the event of a segment-specific problem, the affected trader can "look on" with a neighboring trader. This is the hub view of "checkerboard" or "salt and pepper".

Whether to run (or even use) fiber at this time we find is largely a matter of cost or taste. On average, you want move or do a major technology upgrade of trading floors every 3-5 years. You can spend money today in an effort to delay the major upgrades, but your success is a matter of how well you can predict technology and business change.

Many times installations will attempt to "future proof" their site. Some designers choose to preinstall significant additional cabling (or fiber), and others install a conduit to each desk. This makes it relatively easy to place the appropriate future cable system in place. The majority of trading floors that I've seen are built on raised tiles, this aids future installations, and in often required by the sheer volume of cable running to the desks.

Technology which delivers video over Cat. 5 UTP cable, whether through the use of passive (baluns) or active electronics is commonplace. This can be a major consumer of Cat. 5 "pairs", and should be considered as part of the cable plant design. Several vendors of systems which require video delivery and previously required proprietary cables, now affirm these systems, or install them themselves. This can be a major consumer of Cat. 5 pairs to the desk and should be considered as part of the cable plant design.

Keeping proprietary, and unique cables out of the environment is extremely important. Not only from a "cleanliness" and cost standpoint but also to assist future moves and changes. To move a trader from one position to another, when all the wiring is structured, involves merely disconnecting the equipment at one location, connecting at the other, and arranging for the appropriate cross-connects in the cable plant. Contrast this to special cables, which will need to be run to the new location, irrespective of whether they are removed from the old one.

System and Network Monitoring

The need for system and network monitoring, in this environment, can be daunting. The management platform employed must be capable of several functions in order to be effective. In addition to detecting and reporting problems, it must also be capable of utilization and performance trend analysis and sometimes business-related functions (i.e., charging for usage); and it must be possible to incrementally expand it.

We use a combination of Sun Net Manager, application-specific tools and other vendor products. We monitor the health of every computing device on the network. Monitoring elements act on events to change the color of component icons and beep or send e-mail to the operations staff. We have found that the most useful notification device is the alphanumeric pager, which is driven automatically from the monitoring system. [It is possible to have redundancy here whether via multiple lines to the beeper vendor, multiple beeper vendors, or even multiple people receiving the same message.] The propagation delay of a message from our system to the beeper itself, is sometimes a problem. We have tuned our system so that only critical messages are sent during sleeping hours.

If you have a particularly large floor, or a somewhat inexperienced systems staff it can be quite helpful to scan in a floor plan of the trading area. You can then have the appropriate desk position flash when there's a problem. This can saves significant time tracking down the user.

In addition to checking connectivity and resource utilization for every trader workstation, we also monitor application processes and scan log files for clues that there may be an impending problem. We often inform a user that there will soon be a problem with his/her workstation unless we can restart this application. We apply the same vigilance to monitoring the health of our servers, filesystems, etc. The amount of mileage you get from a very small amount of code that triggers various (warning) actions based on matching particular regular expressions from a log file is surprising.

When used heavily with real-time updates even some of the best known/loved software develops memory leaks and other anomalies [some favorites are spreadsheets which are driven to recalculate by the change of a symbol's price (i.e., IBM trades up or down) – this, in turn, can cause significant amounts of recalculation, and after a many commercial spreadsheets show their bad manners). One or two applications in this state, and your workstation starts to thrash it's disk. All applications run slower, and real-time applications are delayed.

Routers, hubs, ethernet switches, and multiplexers are all monitored for performance according to their specific function. A high count of dropped packets on a router interface could point to an looming problem that could have a significant impact on trading applications and delivery of real-time market data (circuit failure?).

Application Wrappers

Another thing that we often do is "wrap" critical processes in a shell script, or other device. This serves not only to notify on process failure (i.e., core dump), but also gather any potentially useful information and restart the process. You might be inclined to think that if a process dumps core it should be left dead until repaired. However, if the process works most of the time or at least a fair amount of the time, and it produces data which is essential to trading – you must restart it during the trading day. The true repair or upgrade can be scheduled during off hours.

Systems Staff

No matter the level of sophistication of the System and Network monitoring infrastructure, it is most important to have an operations staff that is capable of responding swiftly and decisively to resolve problems. This is the greatest challenge in supporting the technology of a trading floor.

Our number one focus is (near) 100% reliability during hours of interest. In some markets there are defined "trading hours". For instance, 10AM to 4PM for New York Stock Exchange (NYSE) regular trading session. Any failure during these hours can be extremely expensive.

We're quite used to hearing "this just cost me 1 million dollars" (converted to the currency of your choice), it seems our traders always have a million, or so on the line, and a system failure of any proportion always has that price tag. If indeed this is the case, then it should be relatively easy for us to justify additional redundancy to the business managers.

The rule of the road is accurate timely data, or no data at all (preferably an indication of failure). Because of the complexity of the systems, and the number of external feeds (providers) – constant vigilance is required. You might be tempted to think that in a situation where you don't have the current price for a symbol that displaying the last price is OK. Actually that is bad data (stale data), and another rule is Bad data is worse than no data.

If a trader doesn't have a price he needs, he can pick up the phone or use some other piece of equipment to find it, but if you're giving him an old price, which is, in fact, a bad price – then you are actively working against him. And your technology is, in fact, worse than having nothing at all.

It's unfortunately all too common to lose a particular data feed during trading hours. This can be the effect of any number of failures – from a wire being pulled or loose in our cable plant (it's amazing the number of times electricians working in your building will accidentally interrupt service) – to any number of feed vendor infrastructure problems.

The nature of the problem determines the action we take to restore service. If we lose a leased line (or some of the data comm. equipment involved in that circuit), we can immediately go to a dial back-up scenario. In general, we will ensure that we have dial-up access to an access point other than the one our feed comes through. An example is a leased line which originates at the vendor office in New York, then our dial back-up number could be at the vendor's home office in Kansas. Or, quite frankly, anywhere in the world where a direct dial modem connection can be made. The volume of trading on our floor, and the potential exposure from faulty or missing data often quickly covers the cost of a call ANYWHERE.

There may come a time when the cost of the call becomes prohibitive. However, the mere fact that you have the ability to restore service over a dial-up (perhaps long distance) connection, means that you have the ability to make this cost/benefit decision.

Remember that we have spent a great deal of time engineering divergent circuits. In the unlikely event of a problem at one of our Telco central offices – a fair number of circuits on ONE SIDE of our redundant plant can be affected. We can initiate dial backups on all of those feeds and be back to fully operational in a short time. [Our dial-up lines are engineered through a different CO than the leased lines.] A very similar circumstance is a cable cut on the main feeder cables into my building. Once again, we can restore full service from this catastrophic event very very quickly (given multiple cable entry points to the building).

Depending on the nature of the problem – we contact the appropriate people to restore (normal) service ASAP. It sometimes requires a keen political sense, as well as good technical grounding (in a number of disciplines) in order to assist vendor personnel of varying abilities to resolve our problems quickly. We have often walked various vendor personnel through trouble shooting procedures over the phone. Procedures one would hope were part of their basic "tool kit".

During an outage, the keys are to find the malfunctioning subsystem and patch around it as soon as possible. Fixing blame, whether it be organizational or personal, is an activity that is best left to the post mortem. We are of the opinion that once a problem is resolved the appropriate follow-up is to document what happened, how it happened and what steps are being taken (have been taken) to insure it never happens again.

After we've restored service, we have the latitude to schedule repair or replacement of the circumvented, failed device(s).

The necessity of near 100% service during trading hours often forces us to fix problems twice. The first time, during the heat of the trading day is simply enough to get things operational again. Later, you schedule an actual repair or upgrade.

We're performing tasks which are mission critical, often with technology which is not designed for this purpose.

Elimination of NFS

In our quest to remove as many single points of failure as possible it is quite common for us to remove dependencies on NFS. What that means is, as far as file access is concerned, making each trader workstation stand-alone.

In this age of multi- GB internal disks, the hardware itself offers few limitations. However, the prospects of keeping a network of 200 300, or 1000 workstations all up to date with the latest copies of applications (in some cases, varying the version depending on the configurations needed by specialized applications used by this trader) can be daunting.

Not only keeping the workstations up to date, but keeping track of what versions are running on each workstation – in case a new software "roll-out" causes a problem. Also required: a quick and reliable way to roll back to a previous version.

One might think that the need for rolling back to a previous version points to a certain lack of Quality Assurance in the process. And frankly, that's true. However, given the complexity of the environment, and the dependency on real-time data which is not easily characterized (were it easy to characterize the data (e.g., predict the future price of IBM) you could be making big profits in the markets) – it is VERY difficult to test applications thoroughly.

That's not to say we don't QA things, it's just that we have found it to be prudent to always be ready to roll-back to the previous version of a newly installed application.

Keep in mind that the amount of money at risk on a daily basis easily dwarfs our paltry salaries.

Automounter

While automounter provides a valuable service, it is not often welcome in a redundant configuration which uses network file service. The reason is that you cannot reliably determine which of the multiple potential servers for a filesystem a client will bind to. If you take a salt and pepper scenario, and you have dependencies which cross colors, you may have made the system less reliable than if you had done nothing at all. Certainly you have not been able to guarantee that a particular client machine depends on nothing which it's neighbor requires.

NIS

Because the failure of a centralized NIS server can be disabling, we often have each client machine act bind to itself (NIS slave) or remove NIS dependencies entirely. The update process, which usually does not take place during "production hours", is not trading critical, hence propagation from a central host is acceptable.

Hardware configurations

For critical servers where the software is not equipped for real-time redundancy (or fail-over) we often keep a stock of backup machines. This can vary from one "cold" backup machine per critical server to a smaller number which serve to backup many. Here the rule of the day is to equip the cold spare with the right amount of memory or the max. of all the machines you are covering, and NO SERVERS have internal disks.

Given a cold spare with the right internal configuration (memory, adapter cards), and well laid out, well labeled cabling – you should be able to swap out a failed CPU in a few minutes, plus reboot/restore time.

This implies that the software running on this server is not hostid specific. It also implies that your computer room is designed well enough that you can easily access your spares (as necessary) and the cables of the failed machine.

The best intentioned software licensing scheme can often wreak havoc on the best intentioned backup scheme. Our best advice – test your backup scenarios regularly.

What do we do when a user workstation fails? Generally speaking, we have a stock of identical spare machines on hand. The user's workstation is swapped for a spare, they are back in business, and we have a broken machine to deal with at some time in the future.

Computer Room Layouts

We have fairly strict rules about the way hardware and cabling is placed in our computer rooms. Besides the normal cleanliness and orderliness comes:

0) Physical security limits access to the computer room, and specifically, trading critical elements – to those with proper authorization (and training).
1) no machine will be directly placed on top of another independent machine. If you do this, then during production hours, the machine on the bottom will always be the one to fail. And you will be faced with the uncomfortable need to take down the one on top in order to fix the other (thereby adding a problem in order to fix one).
2) no machine, and it's redundant pair will be directly adjacent to each other. This affords us the luxury of possibly surviving some significant disasters, i.e., water leaks, VERY well-contained fires, mechanical failures.
3) no machine, and it's redundant pair share the same power circuit. In cases where the size of the installation warrants, and budget permits, two independent UPSes are installed.
4) All critical equipment is protected by UPS, and UPS protection is tested. We're particularly fond of computer rooms where only UPS power is available. But in instances of limited UPS battery life (and no backup generator) it is often helpful to judiciously apply UPS power. In the latter case – you can often save a significant amount of power, and hence extend run-time on battery by removing non-critical monitors from UPS. Multiple power feeds with divergent routing, is not unheard of, in our business.
5) All cables are secured. Whether this means the screws on cables are locked down, or ethernet AUI retaining clips are used. Modular/universal power cords have a tremendous affinity toward working themselves loose at the back of the machine, so we make sure they are not under tension. Some manufacturers have fitted their equipment with a retaining clip, and we applaud them.
6) All cables are managed. Excess cable is either eliminated (i.e., you get the right lengths), or secured, where necessary.
7) We color code things wherever possible, and label EVERYTHING.

It may seem pedantic to spend this much time and energy on something as seemingly boring as a computer room installation, but during an outage, a few seconds saved in restoring service can be a LOT of money.

Given Murphy's law, the person closest to the computer room during an outage is the one with the least experience. So we're also trying to make it easy (almost trivial) for them to work on the hardware.

Yes, sometimes even trading support people go to the bathroom, or, god forbid, take lunch.

License Managers

Quite often, vendor software uses some sort of network license manager. Generally today's crop of software found on trading floors (in fact PERMITTED on our trading floors) uses a fault-tolerant and redundant license manager scheme. We routinely run redundant license managers for all of our licensed software, and there, the remaining exposures come from network partitioning (so that a quorum of managers cannot be achieved) and the fact that keying a license manager to a particular hostid will often be problematic if we execute a quick changeover to a cold back-up CPU. Oftentimes you can work with software vendors in order to alleviate concerns you have in these areas.

Production vs. Test

As the technology marches on, we must stay on top of it. This affords us the need to test and experience a wide range of technologies. After characterizing and evaluating new elements, we sometimes put them into production in a controlled way.

Introducing a new piece of hardware is little different from new software. Even the best testing in the world is only a simulation of the actual environment. Hence we are generally prepared to back out failing "upgrades".

However, one place where we are not always successful is in user developed applications. The TSS system discussed here give users/traders the tools to develop their own analytics and applications, and "publish" new data elements to the floor. While this is generally a good thing, and to be encouraged, if not carefully managed it can lead to subversion of the redundancy/reliability design, and unexpected dependencies on system elements considered non-critical.

We generally consider the failure of an individual trading workstation problematic, but non-critical, in the sense that the impact to the floor is well contained. However, if that failed workstation runs a spreadsheet application, or some other custom analytic which is publishing to the floor, and many traders are dependent on it, then you have a problem.

Our best approach to this problem (short of discouraging innovation from our users) is to carefully educate them as to the meaning of test and production. If there are applications developed by our users which have a need to go "production", then we take control of them, add appropriate redundancy, run them on a server machine in a controlled environment, and finally institute strict change control.

Most spreadsheet programs are woefully inadequate in the change control area, and given that vendors encourage users to develop large business critical applications (i.e., financial models), change control is a necessity.

Moves and Changes

As if keeping one of these floors operational weren't a difficult enough problem – there comes a time in every trading floor's life when a move or major change is necessary.

Moving a trading floor has similarities to moving other production environments, with the added factors of many outside data feeds, very short period of allowable downtime (i.e., one weekend) and absolute criticality for 100% uptime.

Of course, it takes a number of years to outgrow the present quarters, or amortize the original expense, and over those years sins of omission tend to mushroom. We can't exactly blame our predecessors, few of them ever imagined that their "quick and dirty" installation which was meant to be temporary ("we'll go back to fix it later") would become a permanent part of the production environment, and in fact, critical to the trading process.

The process of moving a trading floor can be likened to "putting the left-over spaghetti back into the box". Basically, we need to completely understand the present environment (IN ADVANCE), design the destination environment to at least duplicate it, and then move in a manner with well understood and controlled risk.

This is often a case where spending some money makes it possible to lower the risk. There's often a fortunate correlation between the size and importance of the installation and it's profitability, hence a large a critical installation may well be amenable to spending money in order to reduce their risk.

The tolerance for risk is highly dependent on the organization, but in general it is common practice to pre-install all the infrastructure equipment. Each feed (all the redundancies), each server, all the hubs, routers, etc. If you are very successful then all that is required on move day is to move the users and/or their workstations.

Of course, the possibility of breakage during the move exists, so you prepare yourself with a number of spares, whether they be prebuilt for each configuration, or prebuilt with the "base", and a facility for quickly adding the required special pieces.

It takes a tremendous investment to prebuild all of the infrastructure, and this, again, is a demonstration of the dollars at risk, and potential profits inherent in the trading game.

If you are clever, or fortunate, you can often devise various ways to optimize on the amount of equipment you must buy new for you new environment. Some people try to align moves with the upgrade of various technologies, where they install the upgraded technology at the new site, and deinstall the old site to the scrap heap. This sort of "upgrade during the move" is highly dependent on your organization's tolerance for technology change risk.

If all goes extremely well on move day, you may find yourself in a slightly sleep deprived state, answering questions on how to use the new phones, and where the plumbing infrastructure is located.

Scheduling Upgrades/Changes/Outages

It is highly unlikely that you will be executing changes or upgrades during the trading day. In general, you will do this sort of work off-hours. We tend to like to start these activities just after the traders leave (if on a weekday) – in order to give ourselves all night to fix any problem/s that may occur.

Here, again, is a place where your organization, and the business influences the amount of risk that is permitted. In some shops NO changes of a substantial nature are allowed during options expiration week. That means that you need to keep more on your calendar besides your friend's birthdays, and the next technical conference.

The technical risk of change has to be balanced against the fundamental business need for improvements. If your financial engineers come up with a new, faster or better algorithm for computing an analytic – then there is a potential business gain for pushing that out as soon as possible.

This is in fundamental opposition to the operation manager's goal of limiting change (and hence, hopefully maximizing reliability) – and the opposing views must come to closure.

Support People

What kind of people do we hire to run trading floors? Well, to say that they are extremely bright is a given. It's hard to characterize them simply, because the job has many facets.

On one hand, an exceptional view of the trading plant from a conceptual level is essential. That helps the person quickly home in on the component or components that may be causing our problem. It's not essential that they know each and every piece of the technology, but if they are not expert – then they must be able to help (manage) the experts in resolving the problem(s).

Another given is people who are fairly tireless. In this day of limited profits/budgets for Wall St. firms (as well as the world-wide "streamlining" of corporations) the staff available is going to be limited.

Since it is quite common to be doing floor support during the day, and upgrades on nights or weekends, a willingness to work more than a "standard" work week is often essential.

We also like people who are cool thinkers under stress. There are times when various people are describing a problem in an, ahem, excited way. It requires a cool head in order to filter the emotional from the technical and actually resolve the problem.

At times, just like in other environments, problems which are non-technical come our way. Probably one of my ubiquitous favorites, and one which really does deserve as immediate a response as any other – is the coffee drenched keyboard. We keep a few spare keyboards handy just in case of this (all too common) eventuality.

It also doesn't hurt to have a sense of humor – in fact, that came to mind when asked by a major Wall St. firm what the ONE most important quality in a floor support person is (the others being well known, and more obvious). Second would have to be the ability to build things quickly out of whatever parts are available (reminiscent of the American Television show MacGyver).

Acknowledgments

Some of the material presented in this paper was garnered from discussion and collaboration with Alan Kadin, Fidelity Investments, and Robert Suyemoto, formerly of Fidelity Investments and a consultant to Bank of Boston.

Author Information

Mr. Lipson spent numerous years designing and implementing systems software, LAN and WAN protocols at ComputerVision, BBN Communications, BBN Labs, and various consulting clients. In several years at the Open Software Foundation he worked on OS internals, including OSF/1 and OSF/1AD, international standards and DCE. The last several years he has consulted for financial services institutions including Morgan Stanley (US), Fidelity Investments and Bank of Boston. At present, Mr. Lipson is a consultant to Morgan Stanley Japan Ltd. where he manages the distributed systems and market data groups in the relocation of Tokyo offices and trading floors. Reach him via email at srl@kr.com.