A Census of Tandem System Availability Between 1985 and 1990

Jim Gray
Tandem Computers Inc., Cupertino

Key Words — **Availability, Fault-tolerance, Tandem, End to end**

Reader Aids —
Purpose: Case history
Special math needed for explanations: None
Special math needed to use results: None
Results useful to: Computer designers, reliability engineers, and users

Summary — **Tandem computer systems are designed to be single-fault tolerant. This paper takes a census of customer outages reported to Tandem. The census shows a clear improvement in the reliability of hardware and maintenance. It indicates that now (1990) software is the major source of reported outages (62%), followed by system operations (15%). This is a dramatic shift from the statistics in 1985. Even discounting systematic under-reporting of operations and environmental outages, the conclusion is clear: Hardware faults and hardware maintenance are no longer a major source of outages. As the other components of the system become increasingly reliable, software necessarily becomes the dominant cause of outages. Achieving higher-availability requires: 1) improvement in software quality and software-fault tolerance, 2) simpler operations, and 3) tolerance of operational faults.**

1. INTRODUCTION

Tandem builds single-fault tolerant computers. These computers have better availability than conventional systems, but they do occasionally have outages. It is interesting to examine the outage causes and to observe the trends in outage modes.

Fault-tolerant systems must deal with faults in all areas:

- environment
- operations
- maintenance
- software
- hardware.

If a system has perfect hardware and software, but is difficult to operate, difficult to maintain, and has no protection against power outage, the system will have frequent outages (typically once every three months). As this study shows, there have been substantial improvements in all these areas over the past 5 years — but the improvements have been more dramatic in some areas than in others.

Outages are rare events for most customers, so an adequate sample would study the operator logs of several hundred customers over a period of several years. Such a study is beyond my resources. There are, however, several sources of outage reports easily available to me:

- A database of time and materials for hardware repair by field personnel; this database is used for dispatching and accounting.
- A database of diagnosis and repair on returned field-replaceable units; this is also a tracking and accounting database.
- A database of software-bug reports (Tandem Problem Reports); it is used to track the progress of bug fixes.
- An electronic bulletin board which describes customer problems. This bulletin board is intended to inform Tandem executives of problems at customer sites. Consequently each entry is called an Early Warning Report (EWR).

Of all these sources, only EWRs capture the entire spectrum of outages, including environmental difficulties, operations mistakes, application errors, maintenance mistakes, as well as the more prosaic hardware and software faults. For this reason, I chose EWRs as the vehicle for estimating the causes of outages, and for evaluating trends in outages. I have been reading these reports since late 1984. The reports were analyzed in December of 1985, 1987, and 1989. The 1985 study was documented in [3]. The 1987 study was never published, but was presented at several conferences and was the basis of [5]. This paper summarizes those studies, and reports on the trends they indicate.

Glossary

(See also section 3.1, ''Nomenclature'')

Application: Software not written by Tandem: this includes customer software as well as third-party software packages.

Bug: A software error which, when encountered, becomes a software fault.

Bug fix: A new software version which repairs some software bug.

Case: A collection of Early Warning Reports (EWRs) which all relate to a particular problem at a particular customer. The case may involve multiple outages at multiple systems at multiple sites. For reporting reasons, cases are terminated at the end of each quarter and a new case begun if the problem persists.

Comm: Data communication hardware (phone lines) or software, eg, SNA.

Customer: An entity that operates Tandem equipment. Tandem, the largest single user of Tandem equipment, is not counted as a customer. Large customers, like the United States Government, are viewed in smaller units, like the US Treasury Department or the US Navy. In general, customers are aggregated at a very coarse granularity.

Database: When used as a fault category, this includes all software to support database applications including data storage and retrieval, database utilities, archiving software, transaction protection software, and transaction processing monitors.

0018-9529/90/1000-0409$01.00©1990 IEEE

Disaster: A major environmental difficulty, eg, fire, flood, storm, earthquake, or sabotage.

Early Warning Report: An electronic-mail message sent by a Tandem employee near the customer to Tandem executives to warn them of a customer problem. Frequent faults or an outage are both considered problems. But there are many other kinds of problems.

Environment: Physical resources outside the Tandem system. This includes: 1) external power, air conditioning, common-carrier data communication lines, and physical facilities, and 2) external issues like weather, fire, earthquake, insurrection, and sabotage.

Hardware: Computing equipment, including terminals, communication controllers, processors, memories, discs, tapes, printers, cables, connectors, power supplies, and battery backup power. Excluded are software, environment, common-carrier communication lines, microcode, uninterruptible power supplies, and air conditioning. See also *Software, Environment.*

Failure: *Failure* is not used in the text because it is too easily confused with *fault* and *outage*. See also *Fault, Outage.*

Fatal fault: The first non-tolerated fault in a fault chain; viz, the first fault in the chain to cause an outage. The number of fatal faults equals the number of outages. Every fatal fault is also an implicated fault. Loss of electric power is a fatal fault if it is not masked by an uninterruptible power supply (UPS). If the UPS is present but has a fault then the UPS fault is the fatal fault. See also *Fault chain, Outage.*

Fault: Behavior different from the specified behavior, eg, loss of electric power, or software that behaves improperly. *Failure* is not used in the text because it is too easily confused with *fault* and *outage*. See also *Latent fault, Fault chain, Fatal fault, Outage.*

Fault chain: A sequence of faults related to a single outage. In a single-fault tolerant system, all fault chains are of length two or more: This broke and then that broke.

Fault tolerance: The ability to tolerate or mask faults. Typically, systems are designed and rated as n-fault-tolerant for some number n. Tandem systems are designed to be single-fault tolerant (n = 1), meaning that they are designed to mask most single faults within a repair window. If two faults occur within the repair window then the fault might not be tolerated.

Implicated fault: Any fault in a fault chain containing an outage. See also *Fault chain, Fatal fault, Outage.*

Installed System Database: An internal Tandem database that records Tandem customers and their use of Tandem hardware and software.

Latent Fault: A fault which happened much earlier than it was discovered, eg, software faults or dead-on-arrival spares.

Maintenance: Hardware maintenance of equipment in the field. It explicitly excludes software maintenance (bug fixes). Installation of software bug fixes is included under operations (if it works), and outages caused by incorrect bug fixes are charged to software outages.

Mean Time To Repair: The mean time to repair and restart a system — the average duration of an outage.

Operations: The configuration process of installing new software, configuring the hardware and software (sysgen), and the procedures to keep the system operating, eg, performing archive dumps or restarting faulty systems.

Outage: The denial of service to end users. This is subjective. If 50% of the database is available to 50% of the users, is it available? The customer decides when a fault is a outage. *Failure* is not used in the text because it is too easily confused with *fault* and *outage.* See also *Fault.*

Process: Everything that is not software, hardware, maintenance, operations, and environment, eg, the personality of the salesman, the speed at which bugs are fixed by the vendor, the quality of vendor training, and whether or not the next release is on schedule.

Product: Broad product categories of Tandem software or hardware, eg, all discs are treated as the DISC product, all processors plus memory are treated as the CPU product. On the other hand, problem products (ones with many outages) are each given a separate category.

Site: A customer location or place. A customer with a distributed system can have many sites, and a site can have many systems. See also *System.*

Software: Computer programs, eg, microcode, disc and communication-controller firmware, work-station software, and the full collection of Tandem and customer host software.

System: A node of a Tandem network. In 1989, a typical system consisted of 4 processors, 16 discs, and several hundred terminals.

Acronyms

EWR	Early warning report
ISDB	Installed system database
MTBFt	Mean time between faults
MTBO	Mean time between outages
MTTR	Mean time to repair
OS	Operating system software
VLSI	Very large-scale integration

2. WHAT IS A TANDEM SYSTEM?

A Tandem system typically consists of 4 processors, 12 discs, a few hundred terminals and their communication gear. For example, the "terminals" might be:

- gas pumps or other point-of-sale terminals
- robots in an automated warehouse
- bar-code readers in an automated factory
- automated-teller machines
- form-processing terminals used for hospitals, police or ambulance dispatching, electronic mail, order entry and processing.

Small systems typically consist of 2 processors, 6 discs, and about 100 terminals, while large systems typically have 16 processors, 100 discs, and 1000s of terminals. System prices (excluding terminals) range from 50k$ to 20M$. Customers with more than 16 processors, partition them into multiple systems, each system having about 10 processors. These systems are then networked to form a complete application system. This partitioning gives an extra level of fault isolation — each system is single-fault tolerant. Large customer sites have over 100 processors partitioned into about 10 systems.

Given this huge spread, a factor of 200, between big and little systems, it must seem strange to lump outages together. Outages are the key metric because Tandem systems are designed to be single-fault tolerant; at least two faults should be necessary to cause a outage. One could normalize the outage severity:

duration × number_of_processors × processor_speed

so a short outage on a small system would be treated as less important than a long outage on a big system. Alternatively, one could focus on outages of production systems, and exclude systems being installed, developed, relocated, or maintained. The database allows both refinements; but such refinements quickly lead to an explosion of data.

Our attitude is: Fault-tolerance, Outage-intolerance. The focus is on why systems fail rather than on the consequences of the outage. The goal is to reduce all causes and forms of outage. So this report treats all outages alike; and focuses on outages rather than on tolerated faults.

Another confounding variable is the *shifting product mix* over the 5-year interval. When the study began, most hardware modules were delivering one year MTBFt. At the end of the measurement period, new hardware had 10 times the MTBFt. This period saw a transition from medium-scale integration to VLSI, from removable discs to sealed units, and a 90% reduction in cabling and connectors — as technology moved to surface mount devices and fiber-optics. Much of the old hardware has not been retired. Therefore the old equipment contributes disproportionately to the fault statistics. If the old hardware is only 10% of the installed base, it still contributes over 50% of the outages. Similarly, the software base grew by a factor of 3 (measured as lines-of-code). No attempt was made to segregate these outages, rather they are all lumped together. The heterogeneity of systems is a reality which each vendor and customer must deal with. Despite this heterogeneity, some trends do emerge when the data are examined over this 5-year period (1985-1990).

3. TANDEM EARLY WARNING REPORTS

3.1 *Nomenclature*

Each outage of a Tandem system should generate an Early Warning Report (EWR). Unfortunately, this is not always the case. Many environmental and operations outages are never reported to Tandem. We believe that:

- Outages caused by maintenance, hardware, or Tandem software are generally reported
- Outages induced by application software, operations, or environment are under-reported.

This under-reporting is difficult to quantify, but is considered in section 4, ''Experience of a Specific Customer''.

EWRs are typically written by a Tandem employee close to the situation: the account representative, the software support person (analyst), or hardware support person (customer engineer). The report begins with some standard information: customer name, system number, system type, software version, cause, and duration of outage. Then follows a free-text description of the situation and customer attitude. These reports range from a few paragraphs to 20 pages. If the situation persists, there are more update reports until the case is closed.

Cases involve only one customer. But a case can involve many systems at many sites. A particular case has one or more reports, and a report describes one or more faults: situations where a component did not behave correctly. Some faults give rise to outages: a denial of service. The definition of outage is not precise; if most of the system is available, then most clients usually consider it available. But there have been cases where the unavailability of a single communication line, or even very bad response time, have been declared outages. The definition of outage is left to the customer; the EWR form has a field asking: ''Did the customer regard his system as down?''. If the answer to this question is *Yes*, the EWR reports one or more outages.

The fault-sequence that causes an outage is called a fault chain. For example, suppose a software bug halts a processor — that is the first fault in the chain. The system tries to move all processing out of the faulted processor to other processors. If this works, there is no outage and fault-tolerance has masked the fault. Now the operator tries to get a picture of the processor state (a dump), so the bug can be diagnosed and fixed. He then restarts the processor and it rejoins the system. If the operator makes a mistake during this procedure, eg, he restarts a functioning processor, then that is a second fault in the chain.

Faults are categorized as:

all: Any reported fault — whether it was implicated in an outage or not.

implicated fault: Any fault related to an outage (in a fault chain containing an outage).

fatal fault: The first non-tolerated fault in a fault chain; viz, the first fault in the fault chain to cause an outage. The number of fatal faults equals the number of outages. If the hardware is faulty, and the software masks the fault, then there is no fatal fault. If the software does not mask the hardware fault, then the software is faulty and the software fault is the fatal fault.

Counts of all 3 fault-categories are reported here. Fatal faults are the most critical because they cause outages. Once a system enters an outage state, the chance of further faults is much increased. The analysis here focuses on fatal faults.

In summary, customer situations give rise to cases which may have many EWRs. The cases report chains of faults which may produce outages.

3.2 *EWR Data*

As of late 1989, about 3 reports arrived each day. About 2 of these are new cases. The reports are analyzed in bulk at year-end. It takes about an hour to understand, categorize, and record each case. The 5 MB of EWRs for 1989 boiled down to about 2 MB of text and 0.2 MB of structured information. For comparison, Tolstoy's *War and Peace* is about 2 MB.

The EWR statistics, integrated over 5 years, give impressive numbers: approximately 7 k customer years, 30 k system years, 80 k processor years, and over 200 k disc years. Table 1 summarizes the information for the 3 periods.

TABLE 1
Summary EWR Data

	1985	1987	1989
Customers	1000	1300	2000
EWR Customers	?	?	267
Outage Customers	176	205	164
Systems	2.4 k	6 k	9 k
Processors	7 k	15 k	25.5 k
Discs	16 k	46 k	74 k
Cases	305	227	501
Reports	491	535	766
Faults	592	609	892
Outages	285	294	438
System MTBF	8 years	20 years	21 years

The number of customers and systems changed during the reporting period, so the period mid-point value is reported. The customer and system numbers are accurate to 10%. The Tandem customer-file had over 5000 entries in 1989. Often "customers" are sites of a larger application system, (eg, the Singapore node of a network). This study aggregates customers into corporate groups, like US Navy, or New York Stock Exchange, rather than individual departments or sites. Tandem has built about 13000 systems. The number of systems reported here excludes retired systems, internal systems, and Unix systems.

The previous study [3] discounted outages caused by beta-test software and outages of systems which were not configured with fault-tolerant hardware (mirrored discs). Such outages comprise less than 5% of the total and so the subtraction was not important — there is more than a 5% error in the reporting process. In addition, previous studies subtracted outages caused by *infant* software or hardware. This *infant*-subtraction was important: about 30% of all outages both in 1985 and 1989 were due to a few troublesome products. There will always be *infants* in the field, so it is unfair to subtract them from the statistics. Hence, the numbers reported here include all error-prone software and hardware — there is no subtraction of beta-test software, or of non-fault-tolerant systems. Interestingly enough, in 1985 the *infants* were mostly hardware, in 1987 *infants* were split between hardware and software, and in 1989 most of the *infants* were in software. By 1989, when hardware was implicated, the culprit was often firmware (viz, software).

Table 1 shows that cases, reports, faults, and outages all increased over the period. But during this period the number of systems grew even more rapidly, so the outages per system actually decreased. Put another way, the mean time-between-reported-outages increased from 8 years to about 20 years. Recall that there is under-reporting of environmental, operations, and application faults. So the actual MTBOs are considerably worse. But the trend to longer MTBOs is unmistakable.

Interesting trends emerge when faults are analyzed by category. The broad categories are:

- software (application and vendor)
- hardware (vendor)
- maintenance (typically by the vendor)
- operations (management of the system)
- environment (power, facilities, comm lines)
- process (the infrastructure that supports the system such as software distribution, and project management).

Using this decomposition, table 2 summarizes the census of various kinds of faults by year.

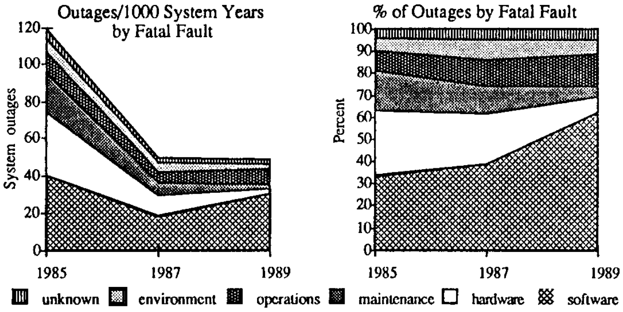
TABLE 2
Faults by Year by Cause

	Outages by Fatal Fault			by Implicated Fault		All Faults
Year	1985	1987	1989	1987	1989	1989
Software	96	114	272	135	297	515
Hardware	82	66	29	106	77	157
Maintenance	53	37	22	42	28	28
Operations	25	35	66	49	86	27
Environment	17	28	26	37	27	103
Process	?	?	0	?	9	61
Unknown	12	14	23	17	23	21
Total	285	294	438	386	538	892

Tables 1 and 2 have the raw data and are included because it is frustrating to extract numbers from graphs. Figures 2 and 3 display the information graphically. Figure 1 shows the basic trends: outages per 1000 years (per millennium) improved by a factor of two by 1987 and then held steady. Most of the improvement came from improvements in hardware and maintenance, which together shrank from 50% of the outages to under 10%. By contrast, operations grew from 9% to 15% of outages. Software's share of the problem got much bigger during the period, growing from 33% to more than 60% of the outages.

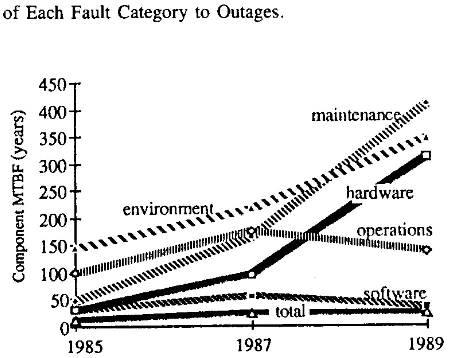
Figure 1 seems to say that operations and software got worse; that is not true. Figure 2 shows that software and operations MTBOs stayed about constant, while the other fault sources improved considerably.

I cannot explain the reported improvement in environment. There is extreme under-reporting in this area since Tandem executives do not need to be warned that the customer had a power outage — recall that EWR stands for *Early* Warning Report. Certainly, Tandem systems tolerate environmental problems better than they did 5 years ago — but as the study of a specific customer shows (section 4), virtually no environmental problems (eg, fire, flood, or earthquake) are reported as EWRs unless special support is required from Tandem.



[Figure 1b shows a shift from hardware and maintenance as the main cause of outages. Now software, and to a lesser extent operations, are the main causes of outage. Under-reporting of environment and operations outages should be considered when reading these graphs.]

Figure 1. a) Declining Frequency of Outages by Cause, and b) Relative Contribution of Each Fault Category to Outages.



[Software and operations held about constant while hardware, maintenance, and environment improved dramatically.]

Figure 2. Trend in Mean Time-Between-Outages by Fatal Cause.

3.3 *Why Did Maintenance Get so Much Better?*

The improvement in maintenance is both impressive and real. Two forces improved maintenance: technology and design. Discs give the best example of both forces. In 1985, each disc had to be serviced once a year. This involved powering down the disc, replacing an air filter, adjusting the power system, and sometimes adjusting head alignment. In addition, the typical 1985 disc had one unscheduled service call per year. This created: 1) a huge workload for customer engineers (32 k tasks per year in 1985), and 2) many opportunities for mistakes. In addition, the disc cabinets and connectors were not designed for maintenance — everything was awkward, and special tools were required. If this had not changed, Tandem customer engineers would now be performing 150 k of these tasks per year: 175 full-time people just doing the fault-prone task of disc maintenance. Instead, current discs have no scheduled maintenance, use no tools (only thumb screws), and have fiber-optic connectors which reduced cabling and connectors by 95%. All field replaceable units have built-in self-test, and light-emitting diodes which indicate correct operation. Disc MTBFt has risen from 8 k hours to over 100 k hours (observed) since 1985. Disc controllers and power supplies have experienced similar dramatic improvements. The net result: the disc population has grown by a factor of 5 while the absolute number of outages induced by disc maintenance has shrunk by a factor of 4: a 2000% improvement. Virtually all the reported disc maintenance problems were with the "old" discs (ones sold prior to 1986), or were incident to installing new discs.

This is just one example of how technology and design changes have improved the maintenance picture. Since 1985, the size of the Tandem customer-engineering staff has held almost constant, and has been able to shift its focus from maintenance to installation — even while the installed base has tripled.

3.4 *Why Did Hardware Get so Much Better?*

Since the Tandem system is single-fault tolerant, two hardware faults are required for hardware to cause a reported outage. In the 1989 period, there were well over 10 k hardware faults, but only 29 resulted in a reported outage — the vast majority were masked by the software. The MTBFt of a duplexed pair goes up as the square of the MTBFt of the individual modules [9], so minor changes in module MTBFt can dramatically affect MTBO. As mentioned in section 3.3, the processor, disc, connector, and controller MTBFt improved by a factor of 10 over the period — due to the shift to VLSI, non-removable discs, and fiber-optics. This should give a factor of 100 improvement in the MTBFt of pairs. In fact, only a 9-fold improvement was observed. The three obvious reasons for the shortfall are:

1. There are still many "old" boxes out there with the "old" MTBFt
2. Installation is still not fool-proof
3. Faults are neither physically independent nor statistically independent.

Item 1 alone can explain the entire shortfall. The trend is clear: hardware designers have done a wonderful job and software is able to mask most residual hardware faults — hardware caused only 4% of the reported outages.

3.5 *Why Did Operations Not Improve?*

Operating the New York Stock Exchange is not easy. Likewise, operating the US Navy inventory is not trivial. These are just two of the many large systems covered by this study (actually the New York exchange is about 20 systems, and the US Navy has about 50 systems — one or more at each US Navy base in the world). According to figure 1, every 150 system-years some operator made a mistake serious enough to crash a system. Clearly, mistakes were made more frequently than that — but most mistakes were tolerated by the system.

Operations mistakes were split evenly between two broad categories: Configuration and Procedures.

- Configuration mistakes involve such things as having a startup file that asks the transaction manager to reinitialize itself. This works fine the first time the system starts, but causes

loss of transactions and of data integrity when the system is restarted from a crash. Mixing incompatible software versions is the most common configuration fault.
- The most common procedural mistake is letting the system fill up: either letting some file get so big that there is no more disc space for it, or letting the transaction audit trail get so large that no new log records can be written.

No clearer pattern of operations faults emerges from this study. Anyone reading the Tandem manuals can see that much can be done to simplify and automate the operations process. This has been a major focus of the software development effort since 1986, with particular emphasis on distributed-system management (managing a network of systems).

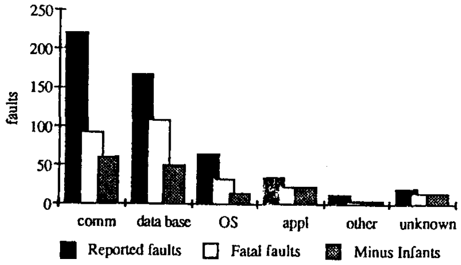
The Tandem disaster recovery product, which replicates applications on two independent systems, was called on once during the 1989 period to save a system — the disaster was an operator command requesting the system to forget its database. The primary system did just that, but fortunately the backup system took-over and continued to offer service [4]. This is an example of software tolerating operator mistakes. The takeover did cause a short outage — but no transactions were lost.

3.6 *Why Did Software Not Improve?*

By 1989, software caused most reported outages (over 60%). During the 5-year period, the software base grew by a factor of 3 to include an SQL implementation, a disaster recovery facility, an application generator, and many aspects of OSI, SNA, TCP/IP, and LAN protocols. In addition, support was added for 3 new processor families, for many new peripherals, and for distributed-system management. Third-party and customer software experienced similar growth. So the system-software complexity increased dramatically. It is surprising that the software-fault rate per system held constant. As the other components of the system become increasingly reliable, software necessarily becomes the dominant cause of outages. This seems to put a 30-year MTBO ceiling on the reliability of computer systems unless a better strategy for tolerating software faults can be found.

Figure 3 gives counts by component of the 1989 reported software faults. Shaded areas show the fatal fault statistics when infants (three trouble prone products) are subtracted. These three products accounted for 25% of all reported outages.

Figure 3 shows that over 40% of the software faults were indeed tolerated by the system (did not cause an outage). The actual ratio is much higher, since tolerated faults rarely cause customer complaints, and so rarely generated an EWR. As reported [3], the process-pair mechanism seems to mask more than 99% of all software faults in system processes. In addition, the transaction mechanism masks many application software faults. So the situation is not hopeless — software quality and tolerance of software faults can be improved. A reasonable goal is to try to build a system with a 100-year MTBO. Allowing software 50% of the outages, implies a 200-year software MTBO — a 7-fold improvement in software quality or software-fault tolerance.



[There is under-reporting of application software faults. The first bar shows all reported faults. The second shows all faults which caused an outage (fatal faults), and third shows the pattern of fatal faults when three error-prone software products are removed from the statistics.]

Figure 3. Broad Categories of Software Faults Reported in 1989.

3.7 *Other Interesting Statistics.*

There is a wealth of information in the EWR database.

MTTR. Hardware, maintenance, and software outages have a MTTR of 4 hours with a median of 1 hour, while operations outages have a MTTR of 10 hours, and environmental outages have a MTTR of 18 hours. In all cases, the outage distributions have very high standard deviations (long and flat tails), so the difference between the median and mean is substantial. Once a system has been down for several hours the distribution begins to look flat: it could be fixed in the next hour, or it could take a day more to fix.

Length of Fault Chains. One might think that in a single-fault tolerant system, almost all fault-chains are of length two. Chains of length one represent cases where the fault is not tolerated. Chains of length three represent three faults within the repair window, a rare event. For the 1986-1987 period —

- 20% of the chains were shorter than 2 (length one). These chains were caused by disaster (fire, flood, ...), by non- fault-tolerant hardware configurations, and by operations disasters.
- 20% of fault chains were longer than 2 (length three or more) — one was of length 8.

There are several causes of long fault chains. The system is designed to tolerate all single faults, but it tolerates some multiple faults. A fault chain of length three results when two faults are tolerated and the third fault is fatal. The length of fault chains is exaggerated because once a system begins to fail, it is in jeopardy. Human-error rates are relatively high; recovery procedures are complex, and are often not well tested. Recovery software suffers from similar complexity and limited testing. Latent faults further increase the chance of multiple faults — if the system has many latent faults, then a double fault may well turn into a triple fault.

To end on a positive note, fault-tolerance works most of the time. Most Tandem customers reported no outages in the 5-year period 1985-1990 (see table 1). The fault-tolerant software masked most hardware faults and tolerated many software and operations faults. There is under-reporting, but it is common to meet a customer who has never had an unscheduled outage — or to meet one who was surprised when he did indeed have an outage. At present, I believe well-managed production systems experience an unscheduled outage about once every 4 years. Extremely well-managed systems — ones with disaster backup, careful procedures, and so on — will do much better than 4-years. Clearly, fault-tolerance is working — 4 years is a factor of 10 better than the comparable figure for a conventional system.

4. THE EXPERIENCE OF A SPECIFIC CUSTOMER

In 1984, I was involved in the initial design of a customer application, and thereafter adopted the customer as a reality check. Since 1986, they have kindly sent me their operator logs each week. I analyzed the operator logs for the period 1986 June to 1987 December. This covered 937 system-weeks (18 system-years).

The customer is a division of an international chemical manufacturer. The division converted its entire operation to Tandem equipment in 1984. The application manages order entry, inventory control, work-flow scheduling, and the actual manufacture of the chemicals. One interesting fact: it costs about $15/gallon to dispose of ''waste'' chemicals, so there is a real incentive to mix those chemicals into a saleable product. Much of the sophistication and benefit of the application derives from picking an optimal set of formulae, so that the plants do not waste chemicals.

The customer has 13 sites: one each in Canada, England, France, Italy, and Mexico; the rest are spread around the continental United States. One site has both a production system and a development system so there are 14 nodes in the network — these nodes have about 54 processors and 124 discs. The system has not grown or changed much since 1985 — some processors and discs have been upgraded to newer models, but the basic application has not changed much. The international nodes were added in 1986. The application uses the Tandem *Encompass* transaction-management system, the *Expand* networking system, and the *Transfer* time-staged-delivery system to manage the flow of information among the nodes.

There are some surprises in the operator logs of this customer. For example, the operators recorded 199 outages, a 4-week MTBO — 1000 times worse than the 20-year MTBO reported in section 3! But 26% of these outages were caused by power faults. If the power goes down, the factory stops; so the customer does not need nor have uninterruptible power supplies for the computers. The system ability to continue processing *after* a power loss is an important asset to this customer. There were 8 power losses exceeding the 2-hour power buffer inherent in all Tandem systems. The average outage due to power losses was 50 minutes; the median was 30 minutes.

All studies show environment to be the most serious cause of outages (75% of unscheduled outages in figure 5). This customer has no environmental protection (uninterruptible power supplies), and is located in some fairly hostile environments. Nevertheless, power losses are a problem for everyone — in urban Northern Europe the rate is one per three years with an average duration 20 minutes, but in most of Europe and North America the rate is two per year with an average duration of 2.5 hours [6,8]. So anyone interested in high availability should have emergency power. Similar remarks apply to redundant communication links.

There were 22 outages related to data-communication equipment. The consequent MTBO was 10 months. The minimum duration was 2 hours and the mean duration was 6 hours. The maximum comm outage was 66 hours. These outages were concentrated in Houston, Texas USA where there are serious electrical storms and flooding.

Over one third (37%) of the outages were scheduled for installing new software or for reorganizing the database. Subsequent to this study, Tandem introduced online, database-reorganization software. Online reorganization will —

- In general: Reduce operator tasks because it involves only issuing a command (no tape handling)
- For this customer: Improve system availability by 25% since it can be done while the database is in use.

But for this customer, such outages are scheduled in advance and so do not affect perceived availability.

There were 99 unscheduled outages, giving a 10-week MTBO. Vendor hardware and software caused relatively few (7%) of these outages — the Tandem equipment delivered a 2.5-year MTBO. There were 2 Tandem-caused software outages — both caused by a bug in a new disc server (DP2). Both outages were reported to Tandem. There were 5 outages caused by disc faults! Two were double-disc faults and were reported as EWRs. The other 3 happened on discs that were not mirrored, so the customer did not complain to Tandem, and no EWR was filed. The discs owned by this customer are ''old'' and rated at 1-year MTBO; this customer saw approximately that MTBO. He subsequently bought more modern discs.

The unscheduled operations outages all centered around the need to archive files before the discs fill up. In 8 cases, the operators did not empty the discs in time; so service was interrupted while the operator moved data to archive storage. There were 10 outages due to application software. No outages related to maintenance. Figure 4 is a pie chart of unscheduled outages. These outages were experienced by the customer over 18 system-years as based on the customer operator logs. The highlighted wedge shows the fraction of outages induced by the vendor. The background wedge shows the outages not reported by EWRs, thus indicating the magnitude of under-reporting of environmental, operations, and application software outages — none of these outages were reported! The unreported hardware outages relate to faults of unmirrored discs. Figure 4 shows that the EWR statistics give an optimistic picture of outages:

- under-reporting was drastic
- the MTBO was 2.5 months, not 20 years. The system gracefully recovered from power losses and so masked half these faults.

Communication line outages were serious. Subsequently, the customer installed a high-bandwidth communication net and

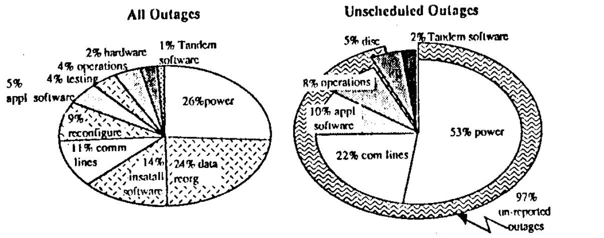


Figure 4. Sources of the 199 Reported Outages and of the 99 Unscheduled Outages

these problems have all but disappeared. The system had application software and operations problems as well. Even discounting these problems — the vendor created 7% of the outages: 3% from unmirrored discs, 2% from mirrored discs, and 2% from a software bug in a new disc server. The disc problems were due to "antique" discs (vintage 1984). The software problems were real.

In sum, the fault-tolerant system offered a 2.5 year MTBO — discounting the antique discs it would have been a 9-year MTBO. This brackets the 4-year MTBO estimate for a well-managed modern system. Why is this customer happy? His *NonStop* system is not especially reliable. Well, two things make the customer happy:

1. The system masks most faults
2. The vendor creates almost no problems.

Without disc mirroring, the antique discs would have caused 90 other outages and a 1-month MTBO — so disc mirroring saved the customer considerable pain. Without battery protection from power-loss, the operators and users would have manually cold-started the system 106 times; instead the system just paused and when power returned it continued processing. The customer does not view power-losses as a computer problem, because the factory stops too. Many other hardware faults were masked. Viewed from that perspective, the system has a 5-month MTBO.

So, the customer has a relatively trouble-free system. That's why he's happy.

5. RELATING CUSTOMER DATA TO EWR DATA.

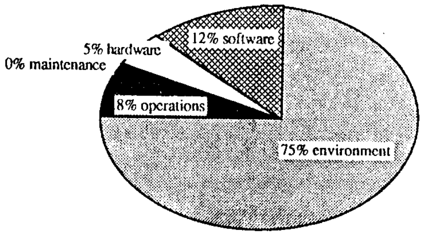
The customer in section 4 is not typical — there's no such thing — but the customer is interesting. The application is spread all over the globe. It's run by people speaking 4 different languages (not counting the differences between British, Canadian, and American) in 7 countries. The application is quite complex:

- It has 5000 different programs
- It involves a centralized development staff of about 100 people who evolve the system software
- Each location manages its own system, applications, and data.

Given the diverse skills and environments at these locations, it is a great benefit to have a fault-tolerant system. For this customer, virtually none (3%) of the outages were reported in EWRs. As predicted, there was good reporting of outages caused by Tandem staff, software, and hardware. But there was no reporting of outages caused by application software, operations, or environment. In addition, no scheduled outages were reported. I believe this customer is an extreme case. Most customers are not spread so far and wide across the globe, and are not located in rural chemical plants. So this customer's statistics are probably as bad as it gets. If figure 4 is viewed in the framework of the EWR analysis then:

- comm-lines and power are both environmental faults
- discs are hardware faults
- all software faults are lumped together.

There were no maintenance outages. Figure 5 is the resulting pie chart of the unscheduled outages in figure 4.



[Figure 4 is recast in the taxonomy of the EWR analysis. All software faults are lumped together; comm-line and power faults are lumped as environment.]

Figure 5. Unscheduled Outages of A Customer

The reason for adopting the EWR database, as opposed to some other data source, was its easy availability and its coverage of the entire outage spectrum. In retrospect, the EWR database is a good indicator of customer pain caused by Tandem products. As such, it is a valuable tool for Tandem — it clearly warns of troublesome products or procedures; and it clearly indicates successes from improved design and procedures. On the other hand, it is not a good fault-tolerance metric because it does not capture the other 97% of the outages — specifically it misses most environmental, operations, and application software outages.

There seems no easy way out of this dilemma, a correct study requires customers who keep careful operator logs, and requires a careful study of the logs of many customers by people with a deep understanding of how the system works and how it fails — not a part-time job.

6. CONCLUSIONS

Fault-tolerant systems have better availability than conventional systems, but they do fail occasionally. Tandem Early

Warning Reports (EWRs) give a good indication of outages caused by Tandem software, hardware, and maintenance. Unfortunately they do not indicate the frequency of outages caused by application software, operations, or environment. For one customer, these were 97% of all outages.

The EWR data show a clear trend: hardware faults and maintenance faults have virtually disappeared as causes of outages — together they cause less than 12% of reported outages (1/5 of the software outage rate). The actual fraction is even lower because of under-reporting of outages in other areas. All evidence points to one conclusion: maintenance and hardware are a minor cause of outages — software is indeed masking most such faults.

On the other hand, the reported software outage rate has held about constant, while the software base has grown by a factor of 3. This is laudable, but at present software seems to have a 30-year MTBO. This puts a ceiling on system availability. This statistic does not include time for scheduled outages. Scheduled outages for software upgrades, for reconfiguration, or for data reorganization are usually recorded as operations outages. Such outages are really a hidden form of software outage — true high-availability software would allow such tasks to be done online, while transactions are accessing the data. At present, Tandem software allows many forms of online maintenance, installation, reconfiguration, and reorganization, without disrupting service. But important gaps remain. For example, an outage is required to install most system software. Future disaster recovery software may provide a way to upgrade software online.

Operations is an important cause of outages, second only to software outages in the EWRs, and third after software and environment in the one customer study. To state the obvious: operators are people, they will not be less faulty in the future. The only option is to simplify, or eliminate operator tasks. Configuration must become automatic; routine tasks must be automated.

7. THE FUTURE

Someday, software will mask almost all hardware and maintenance faults, will eliminate almost all operations tasks, and will mask environmental faults by replicating systems at different sites. When that day comes, only software faults will be left — billions and billions of them! We are surprisingly far along this path already.

I am skeptical of plans to build perfect software [2]; rather I hope that mechanisms to contain and tolerate software faults will help mask them [1]. It seems reasonable to try to build systems with a 100-year MTBO. This will require a 7-fold improvement in software MTBO. Software-fault containment via processes, and software-fault masking with process pairs and transactions might be the keys to tolerating software faults, and could give the necessary 7-fold improvement in software MTBO.

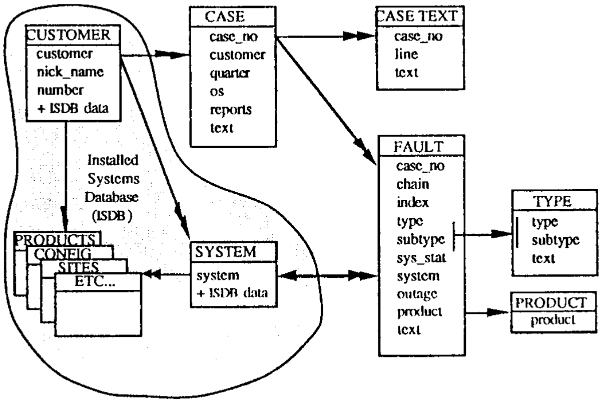
DISCLAIMER & ACKNOWLEDGMENT

This paper is not an official Tandem statement on fault-tolerance. Rather it documents my research on the topic. The views expressed herein are mine, and are not necessarily shared by Tandem Computers Inc.

Andrea Borr, Pat Helland, Franco Putzolu, Praful Shah, Robert Shaw, Hans Jorge Zeller, and two anonymous referees gave valuable advice on presenting this material, and pointed out many areas where the ideas and conclusions could be improved. Bob Horst has been especially helpful in the past and in this study — both in challenging my assumptions and conclusions, and in providing insights and interpretations of some of these statistics. Jim Christen, Jim Hilinski, and Joe Lombardi have generously relayed the operator logs of a Tandem customer each week for the past four years.

APPENDIX. THE EWR DATABASE

Figure 6 show the schema for the SQL database that describes the EWRs.



[The circled tables on the left are extracts of the Tandem Installed Systems Database (ISDB). The remaining tables on the right represent EWR data that extend the ISDB data.]

Figure 6. Schema for the SQL Database for EWRs

Prior to 1989, a combination of spreadsheets and edit files was used to analyze the EWR data. This was adequate in early 1985, when about 100 cases were reported (in a 7-month period). This was cumbersome in 1987 which dealt with 490 outages. So, the 1989 EWR data were entered into an SQL database. Figure 6 is the schema for the database. The SQL data definition statements for the database along with brief comments on each field are included below.

Tandem maintains an Installed Systems Database (ISDB) which tracks all its customers and their systems. Wherever possible, ISDB terminology has been used and that database was extended to include EWR data. In particular, the ISDB customer table was extracted, capturing the customer name and number.

CASE is a table with one record for each case. Recall that a case applies to a problem at a customer. The case can involve many reports and many faults. For reporting reasons, cases end on quarter boundaries. If the problem persists, a new case is begun. Each case record carries a unique case number, the version of software running on the system, the number of reports included in this case, and a 1-line text description of the problem.

CASETEXT is the raw text of the reports, with electronic mail distribution lists removed. Each report line is a separate record. All reports of a case are concatenated in chronological order within casetext.

The FAULT table is the hub of this database. It describes each reported fault. Each fault has some index in some fault chain of some case — this is the record key. The fault has a type (environment, operations, maintenance, software, hardware), and a subtype (eg, power) The types and subtypes are listed below. Each fault record gives the system number and system status (install, development, production, hardware upgrade, software upgrade, off-line maintenance). If the fault caused an outage, the outage duration (minutes) is recorded. The product involved in the fault, along with a 1-line description of the fault, are also in the record. The list of fault types and subtypes gives a good idea of how faults are classified.

Environment	air conditioning
	fire
	halon
	lightening & storm
	maintenance
	power
	quake
	sabotage
	telephone lines
	flood
Hardware	communication controllers & lines
	discs, tapes, controllers
	processors, memory, power
	spare parts
	terminals, printers, workstations, ...
	cables & wiring
Maintenance	communication controllers and lines
	discs, tapes, printers and controllers
	facilities (power, cooling, lights, ...)
	processors, memory, power
Operations	configuration
	install
	overflow of system limits (files full, ...)
	move
	procedures
	upgrade
Process	third-party software house is messing up
	poor analyst support
	too many bugs or bug fixes too slow
	poor customer engineer support
	poor education
	poor marketing support from corporate
	inadequate spares/delays in ship
	poor sales, account control
	announced-software not meeting schedules
Software	software (details unknown)
	customer application
	communications
	data base, recovery, transactions, archive
	languages and tools
	microcode
	operating system
	publications/documentation
	software house application/tool

REFERENCES

[1] A. Avizienis, "Software fault tolerance", 1989 *Proc. IFIP World Computer Conf.* (at San Francisco, California), 1989 Aug; IFIP Press.

[2] R. Grady, "Dissecting software failures", *Hewlett-Packard Journal*, 1989 Apr, pp 57–63; Hewlett-Packard, Cupertino, California.

[3] J. Gray, "Why do computers stop and what can be done about it?", Tandem TR85.7, 1985 Jun; Tandem Computers, Cupertino, California.

[4] J. Lyon, "Tandem's remote data facility", *Proc. CompCon 90* (at San Francisco, California), 1990 Feb, pp 562-567; IEEE Press.

[5] R. Horst, J. Gray, "Learning from field experience with fault tolerant systems", *Proc. Int'l Workshop Hardware Fault Tolerance in Multiprocessors* (at University of Illinois, Urbana), 1989 Jun 19-20, pp 77–79.

[6] A. Reuter, Univ. of Stuttgart, Private communication on the power-failure rates of various European countries.

[8] N. Tullis, "Powering computer-controlled systems: AC or DC?", *Telesis*, 1984, v11.1, pp 8–14; Bell Northern Research.

[9] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components", *Automata Studies*, 1965; Princeton University Press.

AUTHOR

Jim Gray; Tandem Computers Inc.; 19333 Vallco Pkwy; Cupertino, California 95014 USA.

Jim Gray works in the Software Development Dept. of Tandem Computers where he contributes to the design and implementation of database and transaction-processing systems. His research interests include distributed databases, fault-tolerant systems, I/O subsystem architecture, and high-performance transaction-processing systems. During 1990 he is on leave to write a text on transaction processing. He has worked at IBM on database and operating system topics, and at Bell Laboratories. He has taught at Berkeley, Stanford, and Bucharest.

Manuscript TR90-301 received 1990 January 20; revised 1990 May 18.

IEEE Log Number 37713 ◄TR►