ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA TESE DEFENDIDA POR Antonio Cesar Sartoratto Dias E APROVADA PELA COMISSÃO JULGADORA EM 18/07/2007

ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Aplicação de Técnicas de Administração da Produção à Melhoria do Tempo de Resposta em Computadores de Grande Porte

Autor: Antonio Cesar Sartoratto Dias
Orientador: Prof. Dr. Marcius Fabius Henriques de Carvalho

28/2008

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE ENGENHARIA DE FABRICAÇÃO

Aplicação de Técnicas de Administração da Produção à Melhoria do Tempo de Resposta em Computadores de Grande Porte

Autor: Antonio Cesar Sartoratto Dias
Orientador: Prof. Dr. Marcius Fabius Henriques de Carvalho

Curso: Engenharia Mecânica
Área de Concentração: Engenharia de Fabricação

Dissertação de mestrado acadêmico apresentado à Comissão de Pós-Graduação da Faculdade de Engenharia Mecânica, como requisito para obtenção do título de Mestre em Engenharia Mecânica.

Campinas, 2007
SP – Brasil

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Dias, Antonio Cesar Sartoratto
F884p Título: Aplicação de Técnicas de Administração da Produção à Melhoria do Tempo de Resposta em Computadores de Grande Porte / Antonio Cesar Sartoratto Dias – Campinas, SP: [s.n.], 2007.

Orientador: Prof. Dr. Marcius Fabius Henriques de Carvalho
Dissertação - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. 2. 3. 4. I. , . II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Título em Inglês: Applying Operations Management Concepts to Mainframes Response Time Improvement
Palavras-chave em Inglês: mainframe, response time improvement, performance
Área de concentração: Engenharia de Fabricação
Titulação:
Banca examinadora: Prof. Dr. Marcius Fabius Henriques de Carvalho
Prof. Dr. Antonio Batocchio
Prof. Dr. Oscar Salviano Silva Filho
Data da defesa: 18/07/2007

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE ENGENHARIA DE FABRICAÇÃO

DISSERTAÇÃO DE MESTRADO ACADÊMICO

Aplicação de Técnicas da Administração da Produção à Melhoria do Tempo de Resposta em Computadores de Grande Porte

Autor: Antonio Cesar Sartoratto Dias
Orientador: Prof. Dr. Marcius Fabius Henriques de Carvalho

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:

Prof. Dr. Marcius Fabius Henriques de Carvalho
Universidade Estadual de Campinas

Prof. Dr. Antonio Batocchio
Universidade Estadual de Campinas

Prof. Dr. Oscar Salviano Silva Filho
Centro de Pesquisas Renato Archer

Campinas, 18 de Julho de 2007

Dedicatória:

Dedico este trabalho a meu pai Delço Otaviano Dias (in memorian) que me ensinou que o homem deve dominar a máquina e não tornar-se escravo dela.

Agradecimentos

Irei aos grandes e falarei com eles,
eles sabem o caminho do Senhor
(Jeremias 5.5)

Agradeço ao Prof. Dr. Marcius Fabius Henriques de Carvalho pela forma que conduziu esta orientação acadêmica, que também serviu para a vida pessoal e profissional, pelos incentivos, pelo aprimoramento das argumentações, didática e redação.

Agradeço ao Prof. Dr. Antonio Batocchio do Departamento de Engenharia de Fabricação da Faculdade de Engenharia Mecânica da Unicamp pelos ensinamentos dos conceitos de planejamento de projetos e pela atenção durante o período deste trabalho.

Aos professores do Programa de Pós-Graduação do DEF-FEM, pela oportunidade de conhecer novos conceitos, aplicá-los em processos de Tecnologia da Informação e reduzir o tempo de processos apoiados por computadores de grande porte.

Ao Sr. Gino Filippini Neto, diretor de suporte do Banco Mercantil de São Paulo S/A. pela primeira oportunidade de aplicar e validar esta metodologia em ambiente corporativo, pelas realocações de agenda que permitiram estas pesquisas fossem desenvolvidas na FEM durante os anos de 2002 e 2003.

Ao Dr. Paulo Varella, presidente da Prodesp, que permitiu a aplicação desta metodologia em seus *mainframes* dedicados ao atendimento público, no ano de 2004.

Ao Eng^{o} Gustavo Mazzariol, gerente de tecnologia da informação da Companhia do Metropolitano de São Paulo, cuja confiança permitiu atingir novos resultados e aprimorar este trabalho nos anos de 2005 e 2006.

Ao Sr. Gustavo Fonseca da *Fittipaldi International Marketing* pela atenção e sugestões.

À minha esposa Tania, pela compreensão e administração de alguns assuntos meus nesse período e à Isabel, nossa filha, que entre seus 5 e 9 anos, colaborou de alguma forma.

Aos homens de fé que me acolheram, apoiaram e indicaram o caminho nos momentos difíceis.

Resumo

DIAS, Antonio Cesar Sartoratto Dias, Aplicação de Técnicas de Administração da Produção à Melhoria do Tempo de Resposta em Computadores de Grande Porte, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2007. 94 p. Dissertação (Mestrado).

Esta dissertação propõe um método para reduzir o tempo de resposta de aplicativos batch e de transações on-line em computadores de grande porte, sem fazer modificação em seus códigos de programação. Devido às diferenças de velocidade entre discos e processadores, o método proposto visa reduzir o tempo de espera pelas solicitações feitas aos discos através da aplicação de alguns conceitos de administração da produção, como Teoria das Filas, Teoria das Restrições, Processamento em Lotes e Tempo de Setup. O método utilizado reestruturou a forma de uso do hardware disponível ao invés de modificar códigos de aplicativos ou aumentar a capacidade do hardware, duas das opções utilizadas para redução de tempo de resposta. As influências de um ambiente operacional competitivo sobre o desempenho dos aplicativos foram reduzidas a partir do mapeamento das etapas dos processos de Tecnologia da Informação e da utilização dos conceitos de melhoria de produção. O resultado obtido foi uma redução significativa do tempo de processamento de aplicativos e do uso de processador, sem oferecer riscos ao ambiente de apoio aos negócios.

Palavras-chaves:
Redução do tempo de resposta, melhoria do desempenho de sistemas, computadores de grande porte.

Abstract

This dissertation proposes a method to reduce process response time in mainframes without modifying the software codes. Due to the differences between disks and processors speed, the benefits were reached through the disk queue reduction applying some concepts of operations management such as Queuing Theory, Theory of Constraints, Batch Processing and Setup Time reduced the response time in mainframes. The method proposed restructured the use of the hardware available instead of modifying the software codes or increasing hardware capacity, which are two options for response time reduction. Mapping the steps in information technology processes and using concepts of production improvement reduced the influences of a competitive operational environment on software performance. The outcome was a significant processor and run time reduction, without business environment risks.

Keywords:
Response time reduction, system performance improvement, mainframes.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Equações

Nomenclatura

λ	Taxa de chegada
μ	Taxa de atendimento
ρ	Taxa de utilização do recurso
Aplicativo	Conjunto de instruções, escritos pelo homem, que determina para o computador o conjunto de tarefas a serem realizadas. Os termos aplicativo, programa e *software* são sinônimos uma vez que todos eles determinam regras de processamento de dados. A hipótese deste trabalho acrescenta que: aplicativo é um processo de transformação de dados em informações que recebe e causa influência no ambiente de produção.
Batch	Forma de processamento de dados realizado com o agrupamento dos dados em lotes. Outras definições utilizadas: (1) Um grupo de serviços, dados ou programas tratados como uma unidade por um processo de computador; (2) Uma forma de processamento de dados onde um número de serviços são agrupados para processamento em uma mesma execução. [dictionary.com]
Buffer	(1) Dispositivo de armazenamento temporário de dados até o momento que o computador esteja pronto para recebê-lo ou processá-lo. (2) Artifício utilizado para amortecer a diferença entre velocidades de dois dispositivos. [dictionary.com]
Busca	Operação mecânica realizada em um disco magnético em que a cabeça de leitura/gravação realiza para chegar até o cilindro onde está o dado solicitado.
Cache	Dispositivo de armazenamento temporário de alta velocidade localizado no interior da unidade central de processamento de um computador. Também chamado de memória cache. [dictionary.com]
Capacity Planning	Planejamento da capacidade computacional projetada para o futuro baseada em dados históricos.
CICS	*Customer Information Control System*: gerenciador de transações para *mainframes* IBM.

CPU	*Central Processor Unit*: componente principal de um computador que contém o circuito necessário para interpretar e executar as instruções de um programa. [dictionary.com]
EMC	Um dos fabricantes de unidades de discos utilizados por *mainframes*.
Exabyte	Unidade de memória de computador ou de armazenamento de dados equivalente 1.000.000 de Terabytes ou 2^{60} bytes, abreviado por EB.
I/O	Abreviatura de *Input/Output*, são as operações de leituras e gravações realizadas por programas em um computador.
Latência	Operação mecânica realizada em um disco magnético em que a cabeça de leitura/gravação aguarda o início da trilha onde está o dado solicitado.
Log	Arquivo de um computador onde são registradas as atividades realizadas pelo sistema operacional, gerenciadores de bancos de dados, gerenciadores de transações e aplicativos.
Mainframe	Computadores de grande porte voltados para o processamento de dados corporativos, capazes de atender mais de um usuário e processar mais de um aplicativo ao mesmo tempo. Os mainframes utilizados nos ambientes de estudo deste trabalho processavam aproximadamente 4000 tarefas simultâneas.
MCP	Sistema operacional proprietário da Unisys instalado no equipamento NX-5820, utilizado em um dos ambientes de testes deste trabalho.
Megabyte	Unidade de memória de computador ou de armazenamento de dados equivalente a 2^{20} bytes ou 1.048.576 bytes, abreviado por MB. [dictionary.com]
MIPS	Milhões de instruções por segundo. Medida de capacidade de processamento de um computador.
ms	Milissegundo, equivalente a 1^{-3} segundo, também abreviada como mseg.
ns	Nanosegundo, equivalente a 1^{-9} segundo.
RAID	Redundant Array of Inexpensive Disks, sigla dada por Patterson et al (1988) para discos de melhor desempenho, menor consumo de energia elétrica e menor volume que aqueles utilizados pelos computadores de grande porte. São desenvolvidos a partir de discos utilizados por microcomputadores.
RISC	Reduced Instructions Set Computer. Conceito de processador criado pela IBM e pesquisadores da Universidade de Berkeley, entre eles David Patterson.
RMF	Resource Measurement Facility. Gerenciador de recursos computacionais desenvolvido pela IBM para seus *mainframes*.

SSD	Solid State Disk: disco montado com memórias de computadores e precisa de fonte de energia contínua para compensar sua volatilidade. Atualmente, apenas fabricantes menores têm esse produto para oferecer. A EMC, que desenvolveu essa tecnologia, abandonou a fabricação.
tempo de *setup*	Tempo destinado à preparação do equipamento ou ambiente utilizado para a montagem de um determinado produto

Capítulo 1

Introdução

1.1 Breve Histórico

A indústria de informática desenvolveu, ao longo de sua existência, linguagens de programação e metodologias que visaram reduzir o tempo e o custo de desenvolvimento de sistemas, sem que houvesse foco no desempenho do aplicativo gerado [SEBESTA, 2007]. Essas técnicas permitiram criar grande volume de programas em prazos cada vez menores. Além da facilidade de criar sistemas trazida pelos compiladores e pelas ferramentas de geração de aplicativos, começou a surgir no mercado os pacotes de sistemas, desenvolvidos por empresas especializadas e entregues prontos ou quase prontos para serem implantados, como por exemplo, os sistemas ERP (*Enterprise Resource Planning* ou Sistemas Integrados de Gestão Empresarial).

Com essas tendências, as empresas passaram a ter uma quantidade de aplicativos cada vez maior, levando um *mainframe* rapidamente ao seu ponto de saturação.

Quando um computador aproxima-se de sua capacidade máxima, existe o aumento do tempo de resposta das transações e conseqüentemente a diminuição do nível de serviço oferecido aos usuários do sistema. Esse aumento no tempo de resposta não é linear devido à característica da curva de utilização de um recurso, conforme demonstrado pela Teoria das Filas [PRADO, 1999]. Já a diminuição do nível de serviço deixa as empresas sujeitas a perdas de prestígio ou penalidades contratuais.

O impacto causado pelos gargalos ou enfileiramentos são agravados pela diferença de velocidade entre discos e processadores, conforme pesquisas de Hennessy&Patterson (2007). Como a evolução das velocidades dos componentes eletrônicos têm sido superiores aos mecânicos, existem diferenças no interior de um computador de até um milhão de vezes. Partes mecânicas dos discos funcionam em milisegundos e as partes eletrônicas dos processadores em nanosegundos.

O desafio em encontrar uma solução capaz de reduzir o tempo de resposta aumenta de acordo com a quantidade de processos que o *mainframe* é capaz de executar simultaneamente. Os fatores que contribuem para isso são:

- A competição que os programas fazem pelos recursos computacionais disponíveis no momento de seu processamento;
- O risco de desestabilizar o ambiente de produção e reduzir o nível de serviço oferecido aos usuários em conseqüência de modificações de códigos de aplicativos visando redução do tempo de resposta.

Devido às diferenças de desempenho entre processadores e discos este trabalho buscou alternativas para melhorar o desempenho de aplicativos sem que seus códigos fossem modificados e concentrou-se na redução do tempo de espera das solicitações feitas aos discos.

1.2 Evolução da tecnologia

A Figura 1.1 representa a diferença de evolução entre discos e processadores. A distância existente é causada porque os processadores evoluem a uma taxa de 58% ao ano e os discos em 35% ao ano. [PATTERSON&CHEN, 1993; HENNESSY&PATTERSON, 2007]

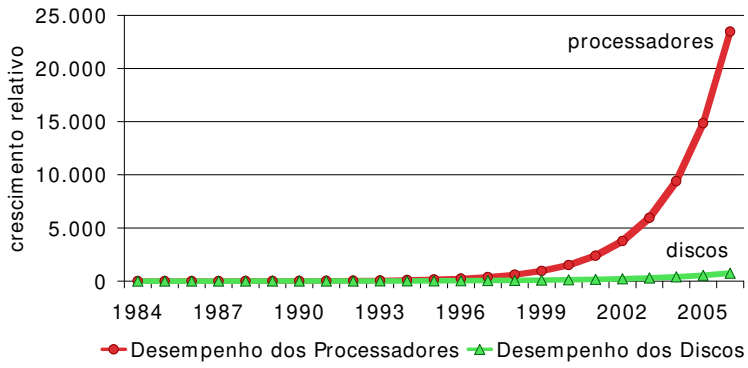

Figura 1.1 – Comparativo entre evolução do desempenho de processadores e discos

A Figura 1.2 representa a observação e projeção feita por Gray&Shenoy (2000), comparando que a capacidade de armazenamento dos discos melhorou em mais de 1000 vezes, coerente com a lei de Moore, entretanto a capacidade na taxa de transferência, que depende da movimentação mecânica da cabeça de leitura e gravação, foi melhorada em apenas 15 vezes.

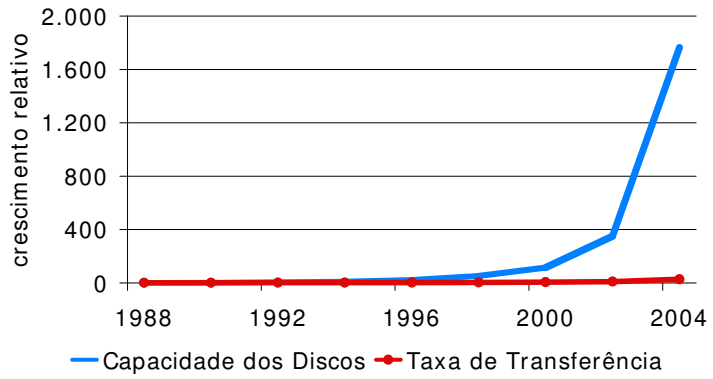

Figura 1.2 – Comparativo entre evolução da capacidade dos discos e taxa de transferência

A Figura 1.3 representa o crescimento da capacidade mundial de armazenamento de dados [PORTER, 2005] em relação à previsão da Lei de Moore [MOORE, 1965]. Esse crescimento incentiva pesquisar sobre o desempenho dos discos para que os dados solicitados trafeguem com velocidade compatível com a do processador.

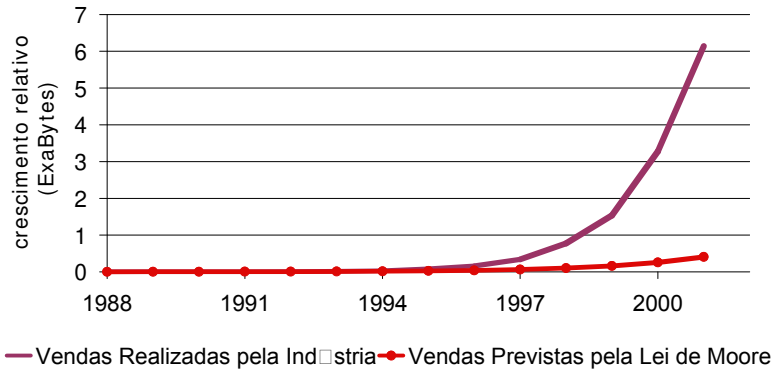

Figura 1.3 – Comparativo entre vendas realizadas e previstas pela Lei de Moore

1.3 Proposta deste trabalho

Este trabalho propõe que o tempo de resposta de aplicativos seja reduzido a partir de melhores formas de utilização dos discos, sem intervenção nos códigos de programação e sem acréscimo de novos *hardwares*.

As três pesquisas anteriores, registradas nas Figuras 1.1 a 1.3, indicam haver boas perspectivas nessa linha de atuação: (1) o desempenho dos processadores evoluiu muito mais que o desempenho dos discos [HENNESSY&PATTERSON, 2007], por isso os ganhos obtidos com foco na redução de uso de processadores não trazem, atualmente, os resultados observados no passado,

(2) a capacidade de armazenamento dos discos tem crescido em níveis muito superiores à capacidade de acesso aos dados [GRAY&SHENOY, 2000], causando aumento no tamanho das filas de espera nos discos, e (3) a quantidade de dados armazenados em discos pelas empresas cresce 112% ao ano [PORTER, 2005], indicando a continuidade da tendência de aumento das filas em discos.

Diante deste cenário de mudanças tecnológicas, a modificação dos códigos de aplicativos, chega a alcançar bons resultados, mas existem situações onde seus recursos são esgotados e as transações não atingem o melhor ponto de desempenho.

A Figura 1.4 representa a decomposição do tempo de resposta de transações realizadas em uma instituição bancária. O tempo médio de processador utilizado pelas lógicas foi de 0,01 segundo, entretanto, o tempo de resposta foi de 2,69 segundos. Esse valor foi o resultado da soma de 3 itens: (1) tempo de espera para início da transação (*WAIT TIME*) = 1,61 seg., (2) tempo de processo (*CPU TIME*) = 0,01 seg., e (3) tempo para realização dos procedimentos de encerramento da transação (*TASK DISPATCH AVG*) = 1,07 seg.

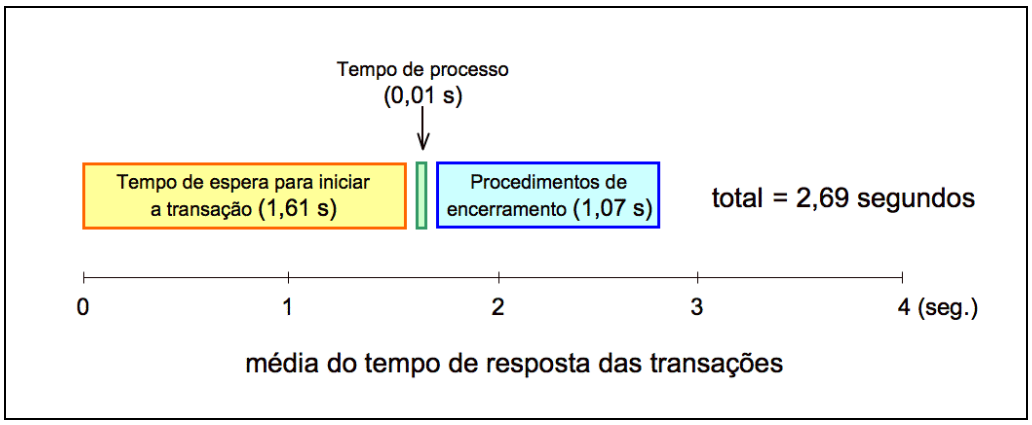

Figura 1.4 – Decomposição do tempo de transações observadas em um banco[1]

Para realizar a análise acima, foram coletados dados referente ao desempenho de 25 milhões de transações bancárias em uma instalação que reduz o tempo de resposta através da modificação da lógica de aplicativos. De fato, os processos foram refinados e apresentaram pequena utilização de processador, entretanto, permaneceram em fila de disco a maior parte do tempo.

[1] A coleta foi realizada em 24/04/07 e utilizou o *software* ASG-TMON for CICS/ESA, em ambiente IBM z/OS.

1.4 Importância e justificativa da escolha do tema

Os computadores de grande porte estão presentes no cotidiano da sociedade. No Brasil, existem 420 mainframes [GODINHO&SAITO, 2007]. São utilizados na retaguarda de serviços bancários, instituições governamentais e empresas de grande porte. Portanto, a redução do tempo de resposta das transações processadas nestes equipamentos trará melhor nível de serviços para os clientes e menores investimentos devido ao melhor uso de recursos.

Alguns fatos justificam o desenvolvimento de um método para reduzir o tempo de resposta sem a necessidade de modificação do código de programação:

1 – Apesar de existir tecnologia para discos de acessos mais rápidos conhecidos como SSD (*Solid State Disk*), os que predominam no mercado são de acesso mecânico, capazes de oferecer tempo de resposta na ordem de milisegundos, tempo 1 milhão de vezes superior ao nanosegundo, velocidade em que atuam os processadores.

2 – Duas das pesquisas utilizadas como referência para este trabalho apontam que *no futuro, a capacidade dos computadores como repositórios de informações irá exceder suas capacidades de computação e comunicação* [GRAY, 2004], [HOARE&MILNER, 2005]. Essa tendência motiva buscar o aperfeiçoamento das formas de utilização dos discos magnéticos.

3 – *Com o incremento do uso de bancos de dados textuais nas empresas, crescerá a quantidade de acessos seqüenciais. Será necessário pesquisar grandes áreas de discos seqüencialmente para localizar o dado necessário* [GRAY ET AL, 2005]. Essa tendência dará maior oportunidade de aplicar os conceitos propostos neste trabalho.

4 – Foi constatado por Denning (2005) que apenas 20% do código de um aplicativo é utilizado com freqüência, os outros 80% são utilizados eventualmente. Como conseqüência, existe a possibilidade que falhas em modificações realizadas na porção de 80% não sejam detectadas nos testes de homologação e insiram riscos no negócio da empresa. Essa constatação incentiva a busca de um novo método de melhoria de tempo de resposta para aplicativos que preserve a maturidade da codificação original.

5 – Mueller&Chaudhry (2000), dois pesquisadores da Microsoft citaram em sua publicação que *melhorar o desempenho de um aplicativo baseado na troca de processador, não é a melhor maneira de resolver problemas de desempenho.* Esta citação, originada em um setor onde os

processadores tem custos menores, motiva pesquisar formas alternativas de melhorar o desempenho de aplicativos em *mainframes*.

6 – Evolução dos processadores versus necessidades dos bancos de dados:

As necessidades dos bancos de dados de suporte à decisão têm crescido 250% ao ano (Lei de Greg), enquanto que os processadores têm crescido 60% ao ano (Lei de Moore) [PATTERSON&KEETON, 2000]. Estes dados motivam a encontrar uma solução para que a distância entre a tecnologia disponível e as necessidades das empresas não sejam aumentadas pelas esperas nas filas de discos.

A Figura 1.5 ilustra a distância entre o desenvolvimento tecnológico dos processadores e as necessidades dos bancos de dados de apoio à decisão utilizados nas empresas [PATTERSON&KEETON, 2000].

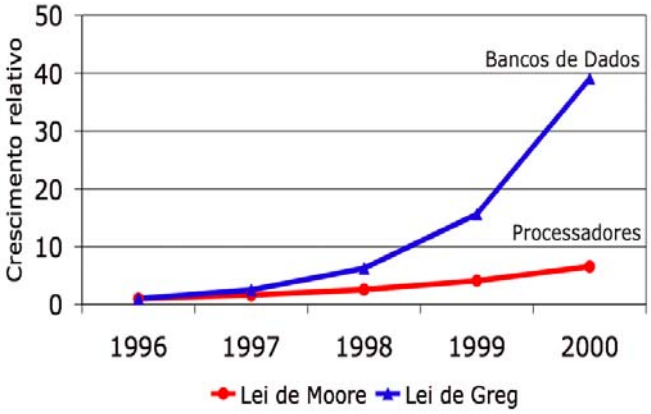

Figura 1.5 – Comparativo entre evolução dos processadores e bancos de dados

7 – Contribuição na redução do tempo de atendimento ao público

Os resultados obtidos pela metodologia proposta contribui para reduzir o tempo de processos de atendimento ao público apoiados por computadores de grande porte. Atuando na melhoria dos serviços de retaguarda e contribuindo para a diminuição do tempo de espera em filas de atendimento.

A Figura 1.6 representa a relação entre os serviços de retaguarda e a frente de atendimento ao público. Segundo Fitzsimmons&Fitzsimmons (2005, p. 95-110) existem duas tendências predominantes: (1) o cliente com a expectativa que a linha de frente ofereça conforto cada vez

maior, ao mesmo tempo que espera que a retaguarda ofereça rapidez, (2) para que a frente de atendimento fique cada vez mais dedicada ao contato direto com o cliente, tende a transferir parte de suas atividades para a retaguarda.

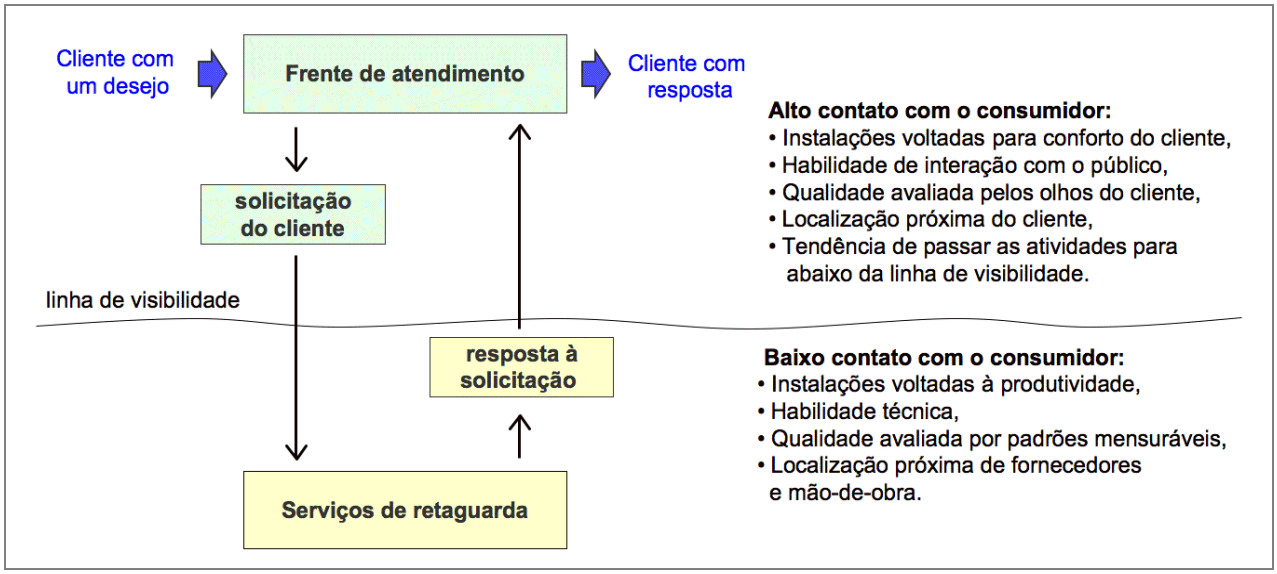

Figura 1.6 – Relação dos serviços de retaguarda com a frente de atendimento

1.5 Delimitações do assunto

Esta pesquisa está delimitada ao estudo da melhoria do tempo de resposta de aplicativos nas seguintes condições:

- Que estejam hospedados em computadores de grande porte, por 2 motivos: (1) os custos elevados de seus processadores, em relação aos equipamentos de menor porte, oferecem maior motivação à busca de eficiência e justificativas financeiras para sua realização, (2) pela escassez de recursos causada pela desproporção entre os milhares de aplicativos em processamento simultâneo e a quantidade de processadores que os atendem, na ordem de unidades.
- Que sejam de apoio às atividades comerciais, por realizarem acesso aos discos com maior intensidade que os aplicativos de finalidades científicas.
- Que utilizem processadores com picos de utilização acima de 75%, pois a partir desse ponto, segundo a Teoria das Filas, existe aumento brusco no tempo de

resposta de um recurso. Abaixo desse ponto, um trabalho de melhoria de desempenho pode não trazer resultados significativos.

- Que realizem leituras e gravações na ordem de milhões de registros, causando saturação na capacidade de acessos dos discos.
- Que armazenem seus dados em discos magnéticos com cabeças de leituras e gravações de acionamento mecânico, devido à maior diferença de velocidades entre eles e o processador.

1.6 Formulação de Hipóteses

Originalmente um aplicativo é um conjunto de instruções, escritos pelo homem, que determina para o computador o conjunto de tarefas a serem realizadas. Os termos aplicativo, programa e *software* são sinônimos uma vez que todos eles determinam regras de processamento de dados. Sob o ponto de vista da administração da produção, utilizada como base deste trabalho, um aplicativo é um processo de transformação de dados em informações que recebe e causa influência no ambiente operacional. Sua forma de atuação é semelhante ao comportamento dos sistemas industriais de produção e a análise de seu desempenho deve incluir o ambiente que dá suporte ao seu funcionamento.

A Figura 1.7 representa um modelo de transformação, utilizado pelo setor industrial, que descreve a natureza da produção e sua interação com o ambiente. Qualquer operação produz bens ou serviços através de um processo de transformação que utiliza recursos para mudar o estado ou condição de algo e produzir *outputs* desejados. [SLACK ET AL, 2002, p. 36]

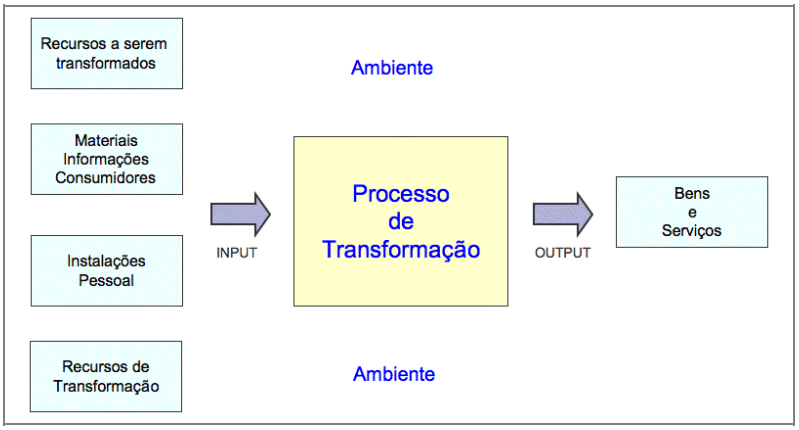

Figura 1.7 – Modelo Geral da Administração da Produção

A Figura 1.8 ilustra uma analogia entre processos industriais e computacionais. A forma de atuar de um computador de grande porte assemelha-se ao comportamento dos sistemas de produção. Quando uma tarefa é solicitada começa a execução de um aplicativo que trabalha na transformação dos dados armazenados e grava os resultados utilizando discos.

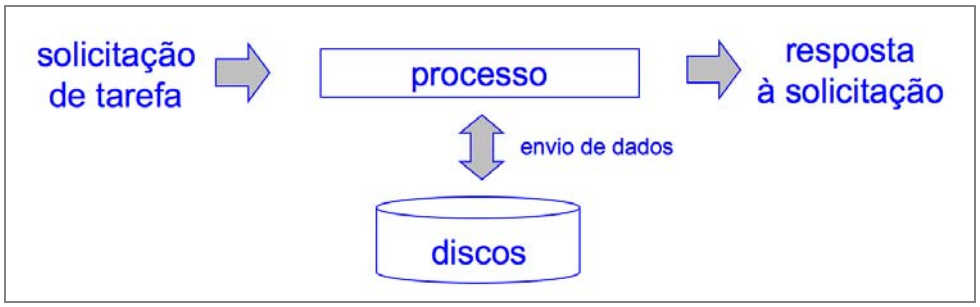

Figura 1.8 – Representação do ambiente operacional de computadores de grande porte

A Figura 1.9 representa uma adaptação do modelo de Slack para processos de Tecnologia da Informação. O modelo recebe uma entrada, faz o tratamento através de um processo de transformação e gera uma saída. Durante o tempo de duração do processo existe interação com o ambiente operacional do *mainframe* que o hospeda.

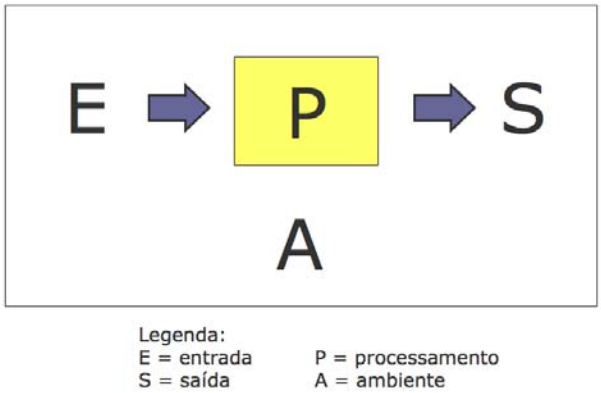

Figura 1.9 – Modelo de Slack adaptado para processos de Tecnologia da Informação

Devido à semelhança observada entre os processos industriais e os computacionais, este trabalho propõe que as técnicas de melhoria utilizadas pela administração da produção sejam aplicadas na construção de uma metodologia que auxilie na redução do tempo das transações de Tecnologia da Informação.

A metodologia proposta por este trabalho visa melhorar o desempenho de aplicativos a partir da reorganização do ambiente operacional e da busca da melhor forma de utilização dos recursos disponíveis. As alterações nas lógicas dos aplicativos apresentarão melhores resultados depois que as influências ambientais forem diminuídas ou eliminadas. As modificações do ambiente computacional estão focadas na observação da competição entre aplicativos e no equilíbrio do uso de recursos computacionais disponíveis.

Os seguintes argumentos amparam a proposta da construção de uma metodologia alternativa para redução de tempo de resposta de aplicativos hospedados em computadores de grande porte:

- Os tempos das operações de leituras e gravações são maiores que as realizadas pelo processador, por isso, as melhorias obtidas com foco no uso de processador serão menores que as obtidas visando acesso aos discos.
- O processamento de dados realizado por um computador de grande porte é composto pela soma dos tempos de espera em filas e os tempos necessários para a realização das tarefas.
- O ambiente operacional causa influência no desempenho de um aplicativo em maior grau que a qualidade do código de programação utilizado na lógica. Dessa forma, as melhorias de aplicativos através da revisão de código só trarão bons resultados depois que as competições por recursos forem reduzidas ou eliminadas do ambiente operacional. A Figura 1.10 ilustra o resumo desta hipótese.

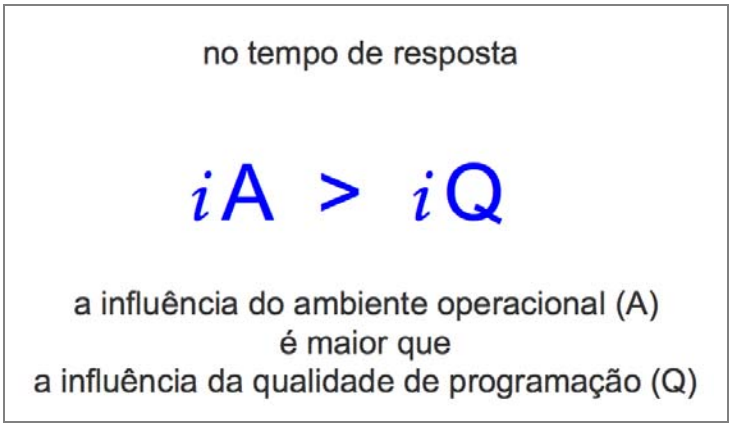

Figura 1.10 – Resumo da hipótese deste trabalho

1.7 Organização deste trabalho

Este trabalho foi organizado da seguinte forma:

- Capítulo 1 - Introdução: Faz um breve relato das evoluções da forma de criação de aplicativos, da tecnologia de discos e processadores. Em seguida, propõe a criação de uma metodologia alternativa baseada em conceitos da administração da produção para reduzir o tempo de transações. Justifica a proposta com dados de pesquisadores ou observados em campo. Estabelece as limitações de atuação e formula as hipóteses para o desenvolvimento da metodologia.

- Capítulo 2 - Revisão Bibliográfica: Relata o conhecimento disponível sobre o tema relativo a técnicas de programação para melhora de desempenho e tecnologia relacionada ao melhor desempenho de processos de T.I.: melhoria de código, uso de memória cache, processamento distribuído e balanceamento de carga. Pondera o risco de modificar lógicas de aplicativos maduros e inserir riscos no negócio. Apresenta, através de testes realizados em ambientes reais de produção, o que cada tecnologia agrega na redução do tempo de resposta de aplicativos. Uma vez que os métodos e as tecnologias analisadas estavam presentes nos ambientes que apresentavam tempos de respostas altos, foi necessário recorrer ao conhecimento de outro setor. Assim, foram estudados e assimilados alguns métodos utilizados pela indústria para melhorar o desempenho de suas tarefas.

- Capítulo 3 - Materiais e Métodos: Organiza a metodologia proposta. Inicia com a decomposição do tempo de resposta em tarefas menores para localizar aquelas que agregam tempo e as que agregam serviço ou tarefas de transformação. O mapeamento de processo é feito através de conceitos de administração da produção. Os resultados são demonstrados em dois ambientes distintos de mainframes, um da Unisys e outro da IBM. Os conceitos utilizados permitiram criar uma metodologia aderente a dois sistemas operacionais de conceitos diferentes e criar uma linguagem imune aos jargões de cada um dos fabricantes. Apresenta os resultados iniciais obtidos.

- Capítulo 4 - Resultados e Discussões: Apresenta com mais detalhes os resultados obtidos pela metodologia nos dois ambientes de testes.

- Capítulo 5 - Conclusões e Sugestões de Trabalhos Futuros: relata as principais conclusões desenvolvidas por este trabalho e sugere os próximos passos para a continuidade das pesquisas.
- Capítulo 6 - Referências Bibliográficas: apresenta as fontes bibliográficas utilizadas no desenvolvimento deste trabalho.

Capítulo 2

Revisão Bibliográfica

Fundamentals stay the same.
(Dizzy Gillespie)[2]

2.1 Métodos tradicionais de melhoria de aplicativos

Tradicionalmente, as melhorias de tempo de resposta em aplicativos são realizadas pontualmente através de serviços divididos nas seguintes categorias:

1 – Melhoria de Código (*Code Optimization*): baseia-se em reescrever partes de um aplicativo para que seu processamento seja mais rápido. Essa linha de pensamento teve origem no início da história dos computadores. [JAMES, 2001]

2 – Uso de Memória *Cache* (*Caching Strategy*): consiste em colocar ou aumentar a memória *cache* de um computador para reduzir a diferença de velocidades entre discos e processadores. Dessa forma os dados serão disponibilizados em tempos menores.

3 – Processamento Distribuído: obtém o aumento de desempenho de uma operação através da realização de tarefas em paralelo por mais de um processador. Esses processadores podem estar dentro de um mesmo computador ou em vários computadores interligados em uma rede. Esse conceito também é chamado de *grid computing*.

4 – Balanceamento de Carga (*Load Balancing*): realizados através de *softwares* que equilibram a carga de trabalho em um ambiente de processamento distribuído com a finalidade de evitar sobrecarga de um lado e ociosidade de outro.

[2] Citado por James Martin e William Ulrich em *The Systems Redevelopment Methodology*, James Martin & Co. (metodologia para correção do *bug* do milênio)

2.2 Melhoria dos códigos de aplicativos

A busca da redução do tempo de processamento de dados através da melhoria de algoritmo teve início com as primeiras codificações de programas para computador. Os pioneiros desse tema queriam tirar todo o proveito do potencial de uma máquina através de aplicativos bem escritos e bem estruturados.

Hopper e Knuth destacam-se entre os defensores da melhoria de código de aplicativos como forma de obter melhor desempenho. Denning pondera a eficácia da revisão e modificação dos códigos de aplicativos.

2.2.1 Grace Murray Hopper

A Almirante Grace Hopper foi uma pioneira na área de desempenho de aplicativos. Entrou para a história da computação como a idealizadora da linguagem COBOL. Foi a primeira a ter usado o termo *bug*, quando um inseto interrompeu o funcionamento do computador que utilizava em 09/09/45 15:45h, conforme anotou em seu diário de bordo. [O'CONNOR&ROBERTSON, 1999]

As analogias e exemplos de Hopper tornaram-se legendários. Costumava entrar em sala de aula com seu nanosegundo em volta do pescoço: um pedaço de fio com aproximadamente 30 cm. Explicava que aquele era o espaço que um elétron corre em um milionésimo de segundo. Por vezes contrastava esse nanosegundo com o microsegundo, uma bobina de fio com mil pés de comprimento, que balançava e fazia ondas enquanto incentivava seus alunos a não desperdiçarem, em seus algoritmos, nem mesmo um milisegundo. [JAMES, 2001]

2.2.2 Donald Ervin Knuth

Os elementos fundamentais de um programa de computador são tempo e espaço. Tempo é a velocidade com que o programa realiza suas tarefas. Espaço é a memória necessária para o programa trabalhar e armazenar os resultados. Mas Knuth não se preocupa apenas com *bytes* e micro-segundos, mas com o conceito de elegância que está por traz da codificação e que se aplica em qualquer nível de programação. Elegância proporciona legibilidade, codificação modular e facilidade para adaptação ou inclusão de tarefas adicionais. [O'CONNOR&ROBERTSON, 2002]

2.2.3 Peter Denning

As pesquisas de Denning permitem refletir sobre a eficácia das melhorias de códigos na redução do tempo de resposta de aplicativos. Sua conclusão é que apenas 20% do código de um aplicativo é utilizado com freqüência, outros 80% são colocados para situações eventuais.

Denning desenvolveu pesquisas e participou de projetos voltados ao gerenciamento de memória e memória virtual. Seu trabalho influenciou muitos dos fabricantes de computadores. Em 1964, o conceito de memória virtual foi introduzido nos computadores de terceira geração fabricados pela Burroughs, General Eletric, RCA e UNIVAC. Apenas em 1972, a IBM incorporou esse conceito na série IBM/370. Em 1980 Denning compilou o trabalho de outros 200 pesquisadores do tema. Essa pesquisa contribuiu para fortalecer o conceito de "Princípio da Localidade" criado em 1959 para melhorar o desempenho do gerenciamento de memórias de computadores. Durante o período de pesquisa, desenvolvimento e implementação desse conceito na indústria de computadores, foi possível observar que grande parte do código de um aplicativo não é utilizado durante sua execução. [DENNING, 2005]

O texto a seguir é o resumo de um artigo de Denning (2005) ilustrado com números publicados pela IBM (2005).

Princípio da localidade

Esse conceito foi criado na Universidade de Manchester em 1959 para aprimorar o funcionamento das memórias virtuais de computadores, que por sua vez foram criadas em 1949, na mesma universidade, para simplificar o gerenciamento de memória. Para equilibrar disponibilidade e necessidade, porções em desuso de aplicativos eram transferidas da memória para disco, cedendo seus espaços para outras e retornando quando fossem utilizadas novamente. Essas porções foram denominadas "páginas". Durante o processo evolutivo do gerenciamento das memórias virtuais, as páginas passaram a receber marcações que indicavam sua freqüência de uso durante a execução do aplicativo. Aquelas páginas de uso mais freqüente eram colocadas em uma memória de velocidade intermediária entre as da memória principal e dos discos, reduzindo o tempo de processo. Ainda hoje, o princípio da localidade tem influência direta na concepção de memórias *cache,* hierarquia de armazenamento de dados, sistemas de bancos de dados, mecanismos de busca, métodos e *softwares* de interface e ambientes de internet. [DENNING, 2005]

Para este trabalho, foi importante um dos frutos das pesquisas de Denning: a observação que nem todas as páginas de um aplicativo são chamadas durante o período de sua execução.

A Figura 2.1 representa a freqüência de uso das páginas dos aplicativos segundo o algoritmo LRU (*least recently used*) desenvolvido pela IBM (2005, p. 331). Esses dados indicam que apenas 20% do código de um aplicativo entra em atividade rotineiramente, outros 80% são de uso eventual.

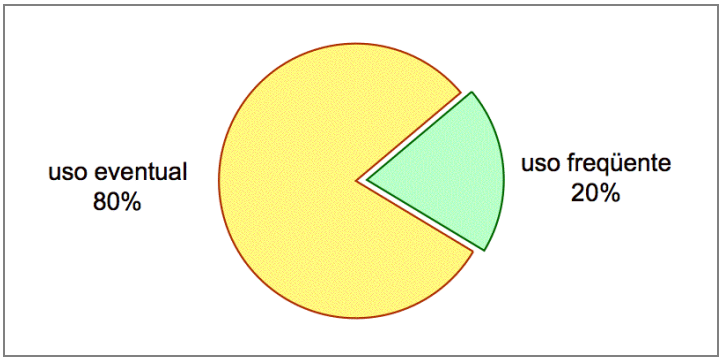

Figura 2.1 – Freqüência de uso das páginas dos aplicativos, segundo a IBM

Esse resultado permite observar que as modificações realizadas na porção dos 80% de menor utilização podem reduzir a confiabilidade e maturidade dos aplicativos. Para reduzir esse risco é necessário que as rotinas de testes sejam aprimoradas e que incluam a simulação de cenários com menores probabilidades de ocorrência.

Além das modificações nas lógicas de aplicativos incluírem riscos nos negócios, seus resultados são inferiores aos observados no passado, conforme ilustrado na Figura 1.4.

2.3 Uso de memória cache

Em 1985, Jim Gray, propôs a Lei dos Cinco Minutos, que gerou o conceito de memória *disk-cache* presente nos discos dos computadores de grande porte atuais. O objetivo era baratear o custo de acesso às informações em um computador de grande porte. Baseado em preços de discos e memória, desenvolveu fórmulas para definir onde a informação deve permanecer para se obter o acesso mais barato. "A informação em memória têm um custo e o acesso a disco para resgatá-la têm outro. A Lei dos Cinco Minutos diz que os dados utilizados nos últimos instantes

devem permanecer em memória, obtendo assim menor custo de acesso. Passado esse tempo, é mais rentável buscar o dado em disco". [GRAY&PUTZOLU, 1985]

Dez anos depois, com as mudanças de preços e capacidades de discos e memória, Gray reviu sua lei com dados atualizados. Definiu que o ponto de equilíbrio é encontrado quando o custo da memória extra para armazenar dados equivale ao custo de acesso a disco necessário para trazê-los de volta. [GRAY&GRAEFE, 1997]

A Figura 2.2 permite observar a redução no tempo de acesso a disco proporcionado pela memória cache. O teste foi realizado em um dos ambientes de estudo onde este trabalho foi desenvolvido.

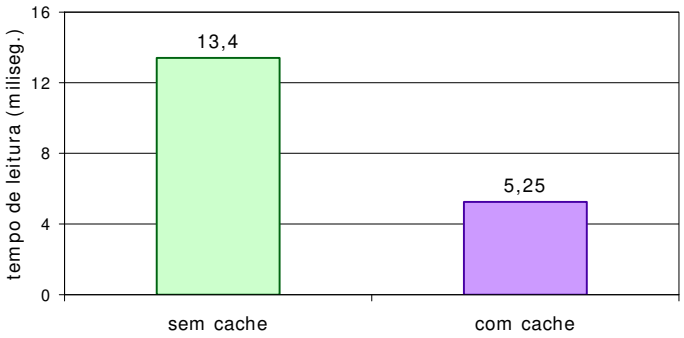

Figura 2.2 – Comparação entre tempos de leituras com e sem cache

Os testes realizados permitiram observar o grau de contribuição da memória *cache* na redução do tempo de acesso a discos. Entretanto, esse efeito ocorreu apenas em processos que utilizaram grande quantidade de leituras seqüenciais. Não houve benefícios para transações que utilizaram acessos aleatórios.

2.4 Processamento distribuído

Gene Amdahl mostrou na IBM sua visão de melhoria de processos através do aperfeiçoamento de hardware utilizados em seus projetos [AMDAHL, 1967]. Durante o desenvolvimento dos *mainframes* IBM/360 e IBM/370 criou algumas leis para definir um sistema de hardware balanceado. Essas leis foram publicadas em 1967 e continuam válidas nos dias atuais. [GRAY&SHENOY, 2000]

Gene Amdahl tornou-se conhecido por muitas regras da engenharia de dados, entre elas a Lei do Paralelismo [GRAY&SHENOY, 2000]:

2.4.1 Lei do paralelismo

Esta lei trata o processamento paralelo como uma forma de aumentar o desempenho de um equipamento e reduzir o tempo de execução de um aplicativo. Ela demonstra que o aumento do número de processadores melhora o desempenho de um computador e diz: se um processo têm uma quantidade de componentes seriais S e uma quantidade de componentes paralelos P, então a máxima aceleração (S_{up} ou *speed-up*) possível para esse processo é dada pela Equação 2.1:

Equação 2.1 – Fator de aceleração em função da quantidade de processos paralelos

$$S_{up} = \frac{(S + P)}{S}$$

Esta lei mostra como reduzir o tempo de um processo em *mainframe*, fragmentando-o e paralelizando-o usando P processadores. A redução de tempo é obtida com a divisão de um processo em fragmentos e cada fragmento é colocado em execução simultaneamente em processadores paralelos.

Gene Amdahl, durante o desenvolvimento dos *mainframes* IBM/360 e IBM/370, criou outras 3 leis importantes para o balanceamento de um processo de Tecnologia da Informação [GRAY&SHENOY, 2000; GRAY ET AL, 2006]:

2.4.2 Lei da Memória

Um sistema equilibrado terá a relação memória/processador = 1, cada 1 MB (*megabyte*) de memória corresponderá a 1 MIPS (milhões de instruções por segundo).

2.4.3 Lei dos Sistemas Equilibrados

Um sistema equilibrado consumirá 8 MIPS (milhões de instruções por segundo) para processar cada MB/s (*megabyte* por segundo) transferido de/para discos.

2.4.4 Lei de Leitura e Gravação

Um sistema equilibrado realizará 50 mil instruções para cada operação de leitura ou gravação.

2.4.5 Associando as leis de Amdahl

Em essência, se 8 instruções são executadas por *byte* lido ou gravado (lei 3), e se 50 K instruções são executadas por leitura ou gravação (lei 4), então as leituras e gravações consumirão 50 K instruções dividido por 8 instruções por byte lido ou gravado, resultando em 6KB por operação de leitura ou gravação.

2.5 Balanceamento de carga

Dentre os autores que utilizam os conceitos de balanceamento de carga para a melhoria do tempo de resposta, destacam-se duas duplas de pesquisadores: uma de brasileiros, formada pelos pesquisadores Daniel Menascé e Virgílio de Almeida e a segunda formada pelos pesquisadores John Hennessy e David Patterson, idealizadores dos processadores RISC e dos discos RAID.

2.5.1 Daniel Menascé e Virgílio Almeida

A infra-estrutura que dá apoio aos serviços de Tecnologia da Informação compreende muitos recursos de *hardware* diferentes, incluindo estações de trabalho, servidores com seus processadores, subsistemas de armazenamento de dados, *LANs*, *WANs*, balanceadores de carga e roteadores. Vários processos necessários ao ambiente de produção, como gerenciadores de bancos de dados, gerenciadores de transações, protocolos, aplicativos e sistemas operacionais compartilham recursos de *hardware*. O uso compartilhado desses recursos aumenta a disputa, gerando filas de espera. Um tempo de resposta à uma solicitação de um usuário é composto pelo tempo que os vários recursos realizam trabalho e pelo tempo de fila à espera pelo atendimento desses recursos. [MENASCÉ&ALMEIDA, 2002, p. 68-71]

A Figura 2.3 ilustra através de um exemplo, a composição do tempo de resposta de um processo de tecnologia da informação que utiliza apenas os recursos de processador e disco. Para que esse processo fosse completado, ele necessitou realizar 4 visitas ao processador e 3 visitas ao disco. Cada visita realizada foi precedida de uma fila de espera pela liberação do recurso. É possível observar a influência dos tempos de filas no tempo de resposta de um processo realizado por computadores de grande porte.

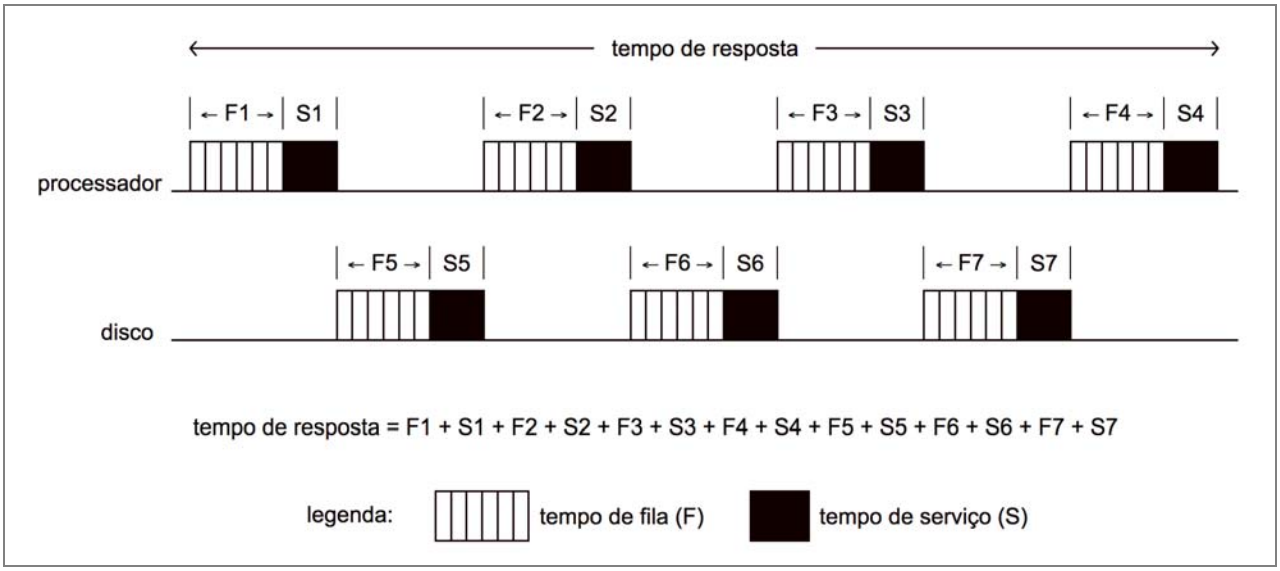

Figura 2.3 – Tempo de resposta de um aplicativo que utiliza processador e disco [MENASCÉ&ALMEIDA, 2002]

Entre as filas formadas para organizar o atendimento de discos e processadores, as relativas aos discos terão maior impacto no tempo de resposta final devido seu menor desempenho em relação aos processadores. Estes operam com velocidade de nanosegundos, enquanto que os discos operam em milisegundos.

Devido às diferenças de velocidades entre os componentes de um ambiente de Tecnologia da Informação é importante que exista equilíbrio entre as cargas de trabalhos distribuídas entre eles. Quando um mainframe possui mais de um processador, o sistema operacional encarrega-se de distribuir equilibradamente a carga de trabalho [MENASCÉ&ALMEIDA, 2002, p. 184-241]. A distribuição da carga de trabalho também é realizada nos discos, conforme registrado a seguir.

2.5.2 John Hennessy e David Patterson

Na década de 1980, um grupo de pesquisadores da Universidade de *Berkeley* propôs um novo tipo de disco criado a partir dos discos utilizados por micro-computadores. A tecnologia desenvolvida recebeu o nome de RAID (*redundant array of inexpensive disks* ou arranjo redundante de discos de baixo custo). Esse mesmo grupo já havia participado do desenvolvimento dos processadores RISC (*reduced instrution set computers* ou computadores de conjuntos de instruções reduzidas) e assim era esperado que processadores de maior velocidade tornassem-se disponíveis. Suas dúvidas eram: (1) o que poderia ser feito com os discos pequenos que acompanhavam os seus microcomputadores? (2) o que poderia ser feito na área de leituras e

gravações para acompanhar os processadores muito mais rápidos? O grupo cogitou substituir uma unidade de disco de *mainframe* por 50 unidades pequenas, pois seria possível obter muito maior desempenho com essa grande quantidade de braços mecânicos independentes. A grande quantidade de unidades pequenas oferecia ainda economia no consumo de energia elétrica e na ocupação do espaço. [HENNESSY&PATTERSON, 2003, p. 772; HENNESSY&PATTERSON, 2007, apêndice K-7]

A proposta do ganho de desempenho a partir do fracionamento de tarefas é coerente com a Lei do Paralelismo de Amdahl [AMDAHL, 1967] revalidada em trabalhos mais recentes [GRAY&SHENOY, 2000; GRAY ET AL, 2006].

Em seu artigo original, o grupo de pesquisadores da Universidade de *Berkeley* mencionou que "o crescimento do desempenho dos processadores e memórias serão desperdiçados se não forem acompanhados de um crescimento de desempenho similar nas operações de leituras e gravações. A capacidade dos discos SLED (*single, large and expensive disk* ou disco individual grande e de alto custo) têm crescido rapidamente, mas seus desempenhos continuam modestos, pois a taxa de transferência não têm crescido na mesma proporção, conforme gráfico apresentado no Capítulo de Introdução deste trabalho. Os discos RAID, construídos a partir de discos desenvolvidos para microcomputadores são uma alternativa atraente com melhorias em desempenho, consumo de energia elétrica, espaço ocupado e escalabilidade" [PATTERSON ET AL, 1988].

A Figura 2.4 representa o conceito de discos RAID. Um registro é fracionado antes de sua gravação para a dividir a carga de trabalho entre os discos que formam o arranjo e assim reduzir o tempo de acesso. Nas operações de leituras essas porções serão localizadas e reagrupadas pela controladora de discos. [PATTERSON ET AL, 1988; MENASCÉ&ALMEIDA, 2002, p. 72-84; HENNESSY&PATTERSON, 2007, p. 362-365]

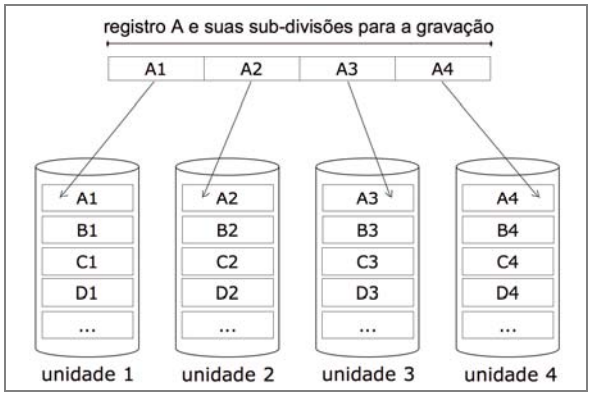

Figura 2.4 – Representação do conceito de discos RAID

No próximo item é possível observar a diferença do tempo de acesso a disco quando são utilizadas as tecnologias RAID e SLED.

2.6 Comparação entre os desempenhos de discos RAID e SLED

A diferença de desempenho entre discos SLED e discos RAID foi observada em um dos ambientes de estudo deste trabalho. A Figura 2.5 compara os tempos de acesso de um mesmo aplicativo utilizando discos de tecnologia SLED e a tecnologia RAID que a substituiu. A Figura 2.6 ilustra a redução no tempo da transação causada pela troca de tecnologia de disco. Os discos utilizados nesta comparação foram: (1) SLED: IBM-9395-B23, (2) RAID-5: IBM-2105-F20

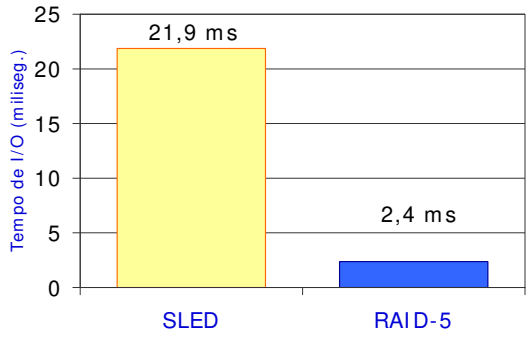

Figura 2.5 – Tempo de resposta em discos SLED X discos RAID

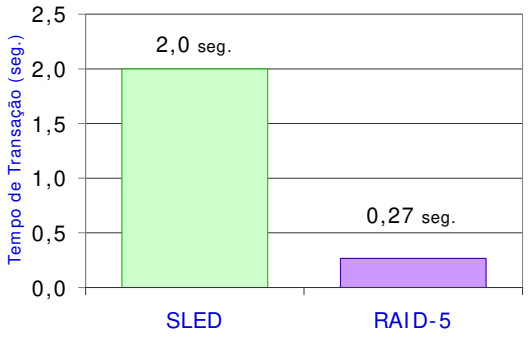

Figura 2.6 – Tempo de transação usando discos SLED X discos RAID

A troca de tecnologia de discos, no ambiente de estudos, proporcionou redução no tempo de resposta, fato coerente com o previsto pela Equação 2.1. Entretanto, mesmo com a nova

tecnologia houve crescimento constante no tempo das transações causado pelo crescimento vegetativo da quantidade de dados armazenados. Essa relação está descrita no próximo item.

2.7 A influência do tempo de acesso aos discos nos tempos das transações

As pesquisas de Porter (2005) apontam que a quantidade de dados armazenados pelas empresas cresce 112% ao ano, provocando aumento no tempo de acesso aos dados, conforme pode ser observado nos dois gráficos a seguir.

A Figura 2.7 registra a evolução linear do tempo de acesso a disco causada pelo crescimento vegetativo dos dados armazenados. A Figura 2.8 registra a evolução do tempo de transação com tendência de crescimento mais acentuado.

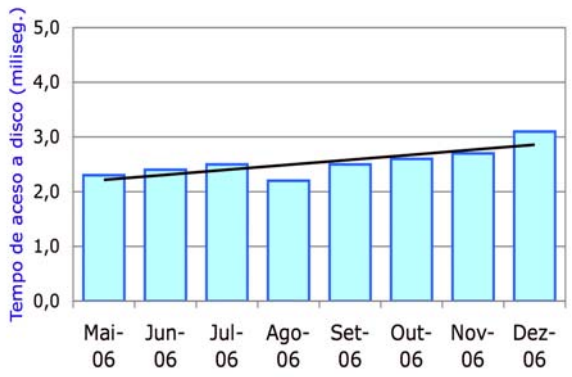

Figura 2.7 – Evolução do tempo de acesso a disco

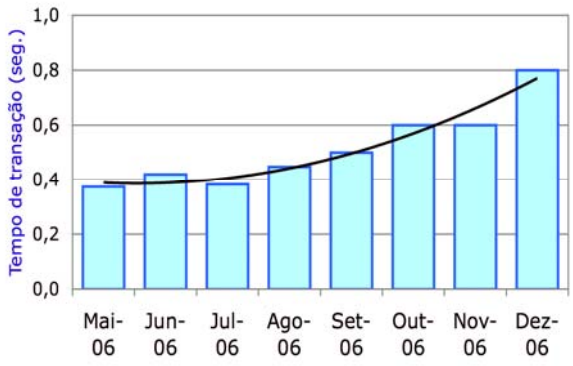

Figura 2.8 – Evolução do tempo de transação

Estas evidências motivaram a realização de pesquisas para localizar formas de reduzir a influência do tempo de acesso a disco no tempo final de um processo, fosse ele um aplicativo *batch*, que processa mais de um lote de dados a cada execução, ou uma transação *on-line*, que processa pequenos volumes de dados sob solicitação do usuário.

2.8 A influência das filas em discos nos dias atuais

Através da Figura 2.9 é possível observar que o tempo de resposta está relacionado com o tamanho da fila em disco. Foram comparados os tempos de acesso a disco variando o tamanho da fila entre 1 e 5 processos. O valor de 2,15 milisegundos, observado para fila = 1, é o *benchmark* para o tempo de resposta do disco marca IBM, modelo DS-4800, lançado no ano de 2007, segundo dados publicados pelo instituto Storage Performance Council. [SPC, 2007]

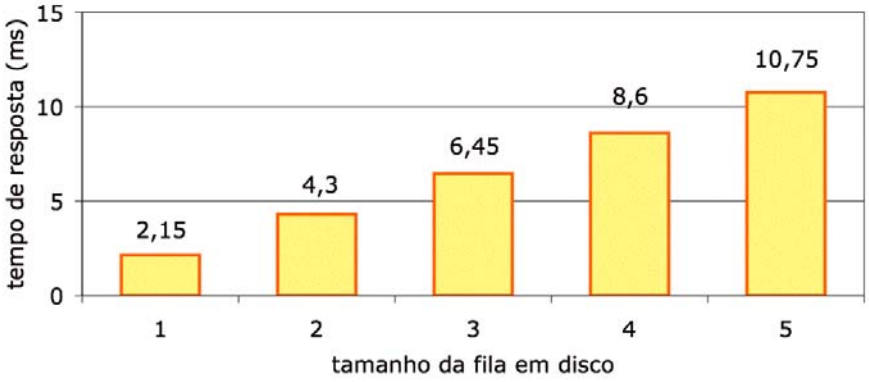

Figura 2.9 – Tamanho da fila X tempo de resposta em disco

A relação que existe entre o tamanho da fila de espera e o tempo de resposta do disco influencia o tempo de processamento de um aplicativo *batch* ou *on-line*, mesmo nas tecnologias mais recentes.

Existe uma proporção do crescimento do tempo de leitura e gravação em disco em função do tamanho da fila de espera. Um pequeno aumento no tempo de acesso de um disco pode gerar uma aumento maior no tempo de resposta. Isso acontece porque o efeito do tempo de resposta para o último processo da fila é o resultado da multiplicação de dois fatores: (1) o tempo de acesso ao disco e (2) a quantidade de processos à sua frente.

Mesmo com o uso de tecnologia modernas, como memória *cache* e discos RAID, o tempo de resposta nos ambientes de estudo estiveram acima das expectativas dos usuários. Por isso, foi importante a utilização da metodologia proposta por este trabalho e que inclui conceitos dos processos de fabricação.

2.9 Processamento de Dados e Processos de Fabricação

Esta revisão de literatura tem o objetivo de compartilhar conhecimentos entre ciências da computação e engenharia de produção, sobretudo os conceitos de administração da produção. A base da motivação para buscar esse compartilhamento de conhecimento tem início nos termos comuns utilizados pelos dois setores, como por exemplo, *setup*, *buffer* e *batch*. Durante o desenvolvimento deste trabalho, foi observado que existem também outros casos de

compartilhamento de conhecimento entre as duas áreas, como, por exemplo, os conceitos de custo de propriedade e família de produtos.

A Figura 2.10 representa tanto um processo de fabricação como um processo de Tecnologia da Informação.

Figura 2.10 – Representação do conceito de processo

Em uma linha de produção é realizado um conjunto seqüencial de operações que processam e transformam matéria prima em produtos disponibilizados para os consumidores finais. Seja qual for o ramo de atividade onde essa linha de produção esteja, ela utilizará a força do trabalho para transformar o insumo de entrada num produto de saída. Os insumos e os produtos finais são organizados em forma de estoque para aguardarem o momento de seguir seu fluxo na cadeia logística.

Em um processo de Tecnologia da Informação, o conjunto seqüencial das operações está previamente escrita no interior do aplicativo. Utilizam um dado como entrada e, após o processo de transformação, realizado pela lógica de programação, geram uma informação disponibilizada para o usuário final. Essa seqüência é válida, tanto para aplicativos *batch*, que processam um grande lote de uma só vez quanto para uma transação acionada a cada vez que o usuário solicita. Os dados, de entrada, de saída ou auxiliares, podem estar armazenados em arquivos convencionais, em bancos de dados ou uma combinação entre eles.

2.10 Conceitos de processos de fabricação utilizados neste trabalho

2.10.1 Conceito de linha de produção

O agrupamento de pessoas num mesmo local para realizarem suas tarefas foi motivado pela disponibilidade das fontes de energia. Primeiramente foram os moinhos de vento e os moinhos

de água, depois as máquinas a vapor. Para melhor aproveitamento da energia, os equipamentos eram alinhados para utilizarem o mesmo eixo movido por uma dessas fontes. [BOTELHO, 2000]

O alinhamento de tarefas foi uma característica que marcou as linhas de produção durante anos. Nos modelos anteriores à energia elétrica, o alinhamento era necessário para compartilhar o eixo motriz. No modelo de Henry Ford, o alinhamento foi motivado pelo uso da esteira. Dessa forma o trabalho era levado até o operário que não necessitava se deslocar pela fábrica em busca de peças ou matérias-primas utilizadas no processo. Ford denominou esse mecanismo de "serviço de transporte". A fixação do trabalhador em postos de trabalho foi a característica marcante no interior da indústria de Ford. Esse modelo foi utilizado para produção dos automóveis Ford modelo T[3]. [BOTELHO, 2000]

Para Womack&Jones (2004) a chave para a produção em massa não reside apenas na linha de montagem contínua. Os lotes sempre significaram longas esperas aguardando a passagem para os departamentos seguintes, que devem funcionar sem parar, justificando equipamentos dedicados de alta velocidade. A busca pela eficiência dos equipamentos é um erro, assim como pensar que a realização de tarefas em lotes seja mais fácil e mais eficiente que as realizadas em fluxo contínuo. Essa forma de pensamento introduz nas empresas os desperdícios e o antídoto para isso é o "pensamento enxuto". Ele é uma forma de especificar valor, alinhar na melhor seqüência as ações que criam valor, realizar essas atividades sem interrupção toda vez que alguém as solicita e realizá-las de forma cada vez mais eficaz, fazendo cada vez mais com cada vez menos, menos esforço humano, menos equipamento, menos tempo e menos espaço e ao mesmo tempo oferecer aos clientes exatamente o que eles desejam.

Womack&Jones (2004) citam ainda que Taiichi Ohno, executivo da Toyota, responsabilizou os primeiros agricultores pela produção baseada no estoque em processo. Segundo ele, esses agricultores ficaram obcecados pelos lotes das colheitas anuais e perderam a visão sábia dos caçadores que encaravam uma tarefa de cada vez.

[3] A denominação "T": os projetos de automóveis desenvolvidos por Henry Ford e sua equipe iniciaram pelo modelo "A", conforme iam sendo aperfeiçoados eram denominados pela letra seguinte. Foram produzidos 15 milhões de modelos "T" entre 1908 e 1927. O tempo de montagem de um automóvel chegou a ser de 93 minutos. O preço inicial foi US$ 850 e terminou em US$ 300, equivalente a US$ 3.000 atuais. [BOTELHO, 2000]

Entre os muitos modelos de produção existem alguns itens comuns, como a seqüencialidade e a inter-dependência das tarefas. Esses dois itens continuam presentes mesmo na produção modular onde é permitido o paralelismo de tarefas.

A revisão dos conceitos de Linha de Produção permitiu aplicar, nas transações realizadas por computadores de grande porte, os conceitos de eficácia desenvolvidos para programação de produção. Durante os projetos que forneceram os dados para esse trabalho foi observado que a busca de altos índices de utilização dos discos em processos de informática provocam aumento no tempo de resposta e maior uso de processador. Incoerências similares são relatados por Goldratt&Cox (2003) sobre custos de estoques provocados pelo compromisso de manter equipamentos em pleno funcionamento para justificar seus custos de aquisição.

2.10.2 Produção modular e tarefas paralelas

Uma alternativa à linha de produção tradicional foi apresentada por Starr (1965) para a IBM viabilizar a produção de computadores com grande variedade de modelos, permitindo atender às diferentes necessidades dos consumidores. Segundo Starr (1965), a divisão de um produto em módulos pré-montados otimiza a montagem final e permite aumentar a variabilidade do produto. A idéia surgiu a partir da percepção de que era preciso produzir vários tipos de computadores para atender às diferentes necessidades dos consumidores, o que seria impossível da forma como a produção era organizada. Comparada ao método tradicional, a montagem em módulos reduz o número de componentes manuseados e o tempo de produção, pois os módulos são montados em tarefas paralelas. Esse método foi adotado pela indústria automobilística onde as tarefas passíveis de padronização foram separadas das mais complexas, que são retiradas da linha principal e pré-montadas na forma de módulos. Assim, a colocação do módulo no veículo consiste de poucas tarefas e o trabalho na linha final torna-se mais simples. [GRAZIADIO, 2004]

Hoje a indústria automobilística aplica esse conceito em muitas montadoras. *O Brasil é considerado um centro de desenvolvimento e teste da estratégia com as seguintes plantas modulares instaladas: GM/RS, VW/RJ, VW-Audi/PR, Ford/BA, DaimlerChrysler/MG e Renault/PR. A produção modular acelera a montagem final pois os módulos podem ser pré-montados paralelamente, e não de modo seqüencial, como no sistema tradicional de produção, o que reduz significativamente o tempo de produção do veículo* [GRAZIADIO, 2004].

2.10.3 Produção industrial em lotes

Na indústria, a técnica de "produção em lotes" é utilizada para processar ou produzir grupos de um mesmo produto. Um exemplo de produção em lotes pode ser encontrado numa padaria, onde os pães são agrupados em lotes para aproveitarem a capacidade de um forno. Quando um processo termina o padeiro inicia um outro lote, até que as necessidades de produção sejam satisfeitas.

O tempo gasto para a mudança de lotes, conhecido como tempo de *setup*, é um item importante nos processos industriais e algumas técnicas têm sido desenvolvidas para reduzi-lo [FRENCH, 1982]. Na indústria de tintas, por exemplo, a produção começa pelas cores claras, como o amarelo claro, seguido pelo amarelo escuro, depois o laranja, vermelho, bordô, até chegar nas cores mais escuras como o preto. Essa seqüência diminui a necessidade de reconfiguração e limpeza do equipamento entre os lotes.

Como a troca das ferramentas de fabricação para produzir uma peça diferente normalmente consumia muito tempo, era sensato fabricar grandes lotes de cada peça antes de trocar as ferramentas para processar a peça seguinte. Esse tempo de parada é visto como uma ineficiência da produção em lotes, mesmo assim, existem vantagens nesse processo. Pela troca de ferramentas, é possível produzir vários produtos a partir de uma única linha de produção. Esta flexibilidade permite, por exemplo, produzir um lote de produtos experimentais e testá-lo junto ao público consumidor com investimentos reduzidos. A produção em lotes também tem sido uma saída viável para empresas pequenas que não podem arcar com a produção contínua. [WOMACK&JONES, 2004]

2.10.4 Pensamento Enxuto

O Pensamento Enxuto foi originado no ambiente de produção da indústria de manufatura, mais precisamente no Sistema Toyota de Produção. Após a II Guerra Mundial, a Toyota iniciou a fabricação de automóveis de passeio, quando esbarrou em alguns problemas. Para solucioná-los desenvolveu o Sistema Toyota de Produção. Os bons resultados incentivaram a implantação da metodologia em diferentes áreas da indústria: administração, desenvolvimento de produtos e produção. Além do segmento automobilístico, sua aplicação foi estendida aos setores aeronáutico, serviços, construção civil, saúde e escritórios em geral. Publicações voltadas ao

Pensamento Enxuto divulgam ganhos de produtividade, redução de custos, redução de *lead time* e melhoria da qualidade de produtos, serviços e processos. [SACOMANO&MELO, 2004, p. 4-6]

Para Womack&Jones (2004, p. 3-9), o Pensamento Enxuto é uma forma de especificar valor, alinhar na melhor seqüência as ações que criam valor, realizar essas atividades sem interrupção toda vez que alguém as solicita e realizá-las de forma cada vez mais eficaz. Em suma, o pensamento enxuto é "enxuto" porque é uma forma de fazer cada vez mais com cada vez menos: menos esforço humano, menos equipamento, menos tempo e menos espaço, e ao mesmo tempo, aproximar-se cada vez mais de oferecer aos clientes exatamente o que eles desejam. O pensamento enxuto também é uma forma de tornar o trabalho mais satisfatório e eliminar desperdícios. É formado por cinco princípios, apresentados na Figura 2.11 e detalhados na seqüência:

1. Valor
2. Cadeia de valor
3. Fluxo contínuo
4. Produção puxada
5. Perfeição

Figura 2.11 – Princípios do pensamento enxuto

(1) O ponto inicial para o pensamento enxuto é a especificação precisa de valor. O valor só pode ser definido pelo cliente final na forma de um produto ou serviço específico, que atenda às necessidades do cliente a um preço específico em um momento específico.

(2) A cadeia de valor é o conjunto de todas as ações específicas necessárias para se levar um produto, um bem, ou um serviço, a passar pelas três tarefas gerenciais críticas em qualquer negócio: a tarefa de solução de problemas que vai da concepção até o lançamento do produto, passando pelo projeto detalhado e pela engenharia; a tarefa de gerenciamento da informação, que vai do recebimento do pedido até a entrega, seguindo um detalhado cronograma; e a tarefa de transformação física e espacial, que vai da matéria-prima ao produto acabado nas mãos do cliente.

(3) Fluxo contínuo: Uma vez que o valor tenha sido especificado com precisão, a cadeia de valor de determinado produto totalmente mapeada pela empresa enxuta e, obviamente, eliminadas as etapas que geram desperdício, chegou a hora de dar o próximo passo no

pensamento enxuto: fazer fluir as etapas restantes, que criam valor. No entanto, essa etapa exige uma mudança completa de mentalidade na empresa.

(4) Produção puxada: O primeiro efeito visível da conversão de departamentos e lotes em equipes de produção e fluxo é que o tempo necessário para se passar da concepção ao lançamento, da venda a entrega, da matéria-prima ao cliente cai drasticamente. Quando se introduz o fluxo, os produtos que consumiam anos para serem projetados são feitos em meses, os pedidos que levavam dias para serem processados estão prontos em questão de horas. As semanas ou meses de tempo de produção convencional são reduzidos a dias ou semanas. Na verdade, se não for possível reduzir rapidamente o tempo de produção à metade no desenvolvimento de produtos, 75% no processamento de pedidos e 90% na produção física, alguma coisa de errado deve estar sendo feita. Além disso, os sistemas enxutos podem fabricar qualquer produto em produção atualmente, em qualquer combinação, de modo a acomodar imediatamente as mudanças na demanda.

(5) Perfeição: À medida que as organizações começarem a especificar valor com precisão, identificarem a cadeia de valor, à medida que fizerem com que os passos para criação de valor fluam continuamente, e deixarem que os clientes puxem o valor da empresa, os desperdícios começarão a desaparecer. Ocorrerá aos envolvidos que o processo de redução de esforço, tempo, espaço, custo e erros é infinito. Ao mesmo tempo, oferecerá um produto que se aproxima muito mais do que o cliente quer. De repente, a perfeição, o quinto e último conceito do pensamento enxuto, não parece uma idéia inatingível.

2.10.5 Teoria das Filas

Filas de espera aparecem em diversos sistemas de produção, particularmente em sistemas de serviço, tais como bancos, supermercados, correios, postos de gasolinas e sistemas de manufatura, por exemplo, produtos aguardando processamento em máquinas ou estações de trabalho. Também aparecem em sistemas de transporte, como em aviões esperando para aterrissar em aeroportos, navios esperando para descarregar em portos. Nos sistemas computacionais as filas aparecem, por exemplo, nas tarefas aguardando processamento em computadores ou pacotes de dados aguardando transmissão através da rede. [ARENALES ET AL, 2007, p. 433]

A Teoria das Filas envolve o estudo matemático das filas formadas quando a demanda de um serviço excede a capacidade de fornecê-lo. As decisões referentes à capacidade de fornecimento devem ser realizadas com freqüência tanto na indústria como em outras áreas. Entretanto, as decisões sobre capacidade tornam-se complexas por causa da dificuldade em prever o momento exato da chegada dos interessados pelos serviços e o tempo de duração de cada atendimento. Oferecer muito serviço pode implicar em custos excessivos, por outro lado, fornecer serviços insuficientes pode causar filas de espera de longa duração que também terão seus custos, sejam custos sociais, custo da perda de clientes ou custo da ociosidade. Portanto, o objetivo final é encontrar o equilíbrio econômico entre o custo do fornecimento do serviço e o custo da espera. A Teoria das Filas não é capaz de solucionar diretamente esse problema, mas contribui com informações vitais para as decisões. [HILLIER&LIEBERMAN, 1967, p. 285-287; ARENALES ET AL, 2007, p. 434-435]

A Figura 2.12 representa um sistema de filas formada por clientes aguardando por um serviço. O termo cliente é usado de uma maneira genérica e pode designar tanto uma pessoa, um navio, uma transação, um processo. Quando a população é muito grande, as chegadas podem ser consideradas independentes. Nesses casos, o ritmo de chegada é uma importante variável aleatória, representada pela letra grega λ (lambda). O processo de atendimento é descrito pela taxa de atendimento μ (mi). A capacidade de atendimento é representada pela letra M, que quantifica o número de atendentes do sistema. Quando $M * \mu > \lambda$, o sistema apresenta um grau de ociosidade. Quando $\lambda > \mu$, o sistema gera fila. Em casos onde isso ocorre, devem ser utilizados alguns artifícios para melhoria do nível de serviço. É o exemplo da quantidade de atendentes numa agência bancária, onde a solução de utilizar quantidades variáveis de atendentes, aparece a partir da divisão do período global em períodos parciais. [PRADO, 1999]

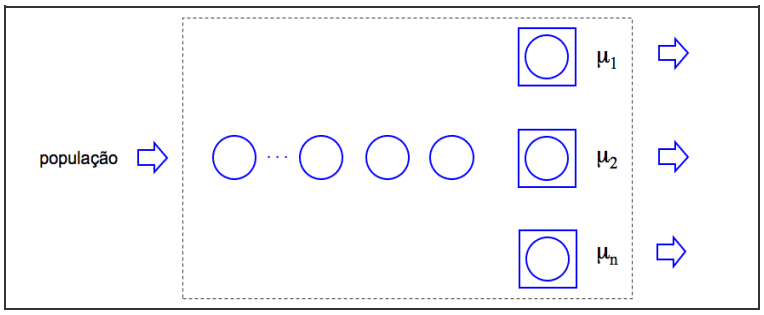

Figura 2.12 – Representação de um sistema de filas

A taxa de utilização dos atendentes é dada pela expressão [PRADO, 1999]:

1 – Quando há um único atendente: $\rho = \frac{\lambda}{\mu}$

2 – Quando há mais de um atendente: $\rho = \frac{\lambda}{M\,\mu}$

A relação entre o número de clientes na fila e a taxa de utilização dos atendentes é dada pela fórmula:

Número de clientes na fila: $NF = \frac{\rho^2}{1 - \rho}$

A Tabela 2.1 e a Figura 2.13 representam a relação entre a utilização de um recurso (U) e o tamanho da fila de espera (F).

Tabela 2.1 – Fila X Utilização de um recurso

Utilização (U)	0%	10%	20%	30%	33%	40%	50%	60%	70%	75%	80%	90%	100%
F = U / (1-U)	0	0,11	0,25	0,43	0,50	0,67	1,00	1,50	2,33	3,00	4,00	9,00	tende ao infinito

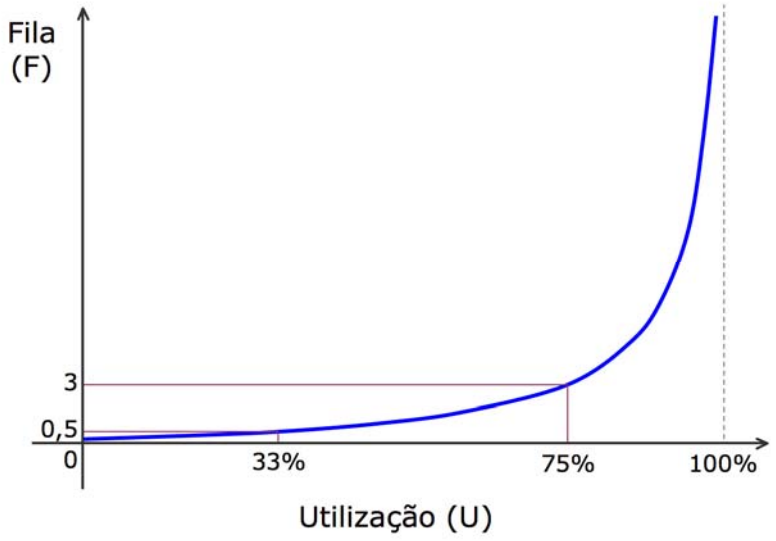

Figura 2.13 – Fila X Utilização de um Recurso

2.10.6 Teoria das Restrições

A Teoria das Restrições é um método científico idealizado pelo físico israelense Eliyahu M. Goldratt para solucionar problemas de produção. Considera que a capacidade máxima de um sistema é definida pela capacidade de uma "restrição gargalo". Ela deve ser localizada e eliminada para que a capacidade de produção do sistema melhore. [GOLDRATT&COX, 2003]

Um sistema de restrições é formado por recursos que limitam sua capacidade de atingir o mais alto desempenho em relação ao seu objetivo. Todo sistema tem restrições e todo sistema possui ao menos uma restrição. [NUNES, 2004]

O método descrito por Goldratt&Cox (2003), tem os seguintes princípios:

- Um sistema funciona como uma corrente, o elo mais fraco determina a força global, por isso deve ser encontrado e reforçado.
- O desempenho máximo do sistema total não é igual à soma do máximo de todos os elos.
- Todo sistema funciona numa relação de causa e efeito. Dentre os eventos indesejáveis observados, alguns são causadores e outros são efeitos.
- Efeitos indesejáveis não são considerados problemas e sim indicadores. Eles são resultados e causas escondidas que devem ser localizadas e tratadas.
- Uma solução é perecível. Ela se deteriora com o tempo.
- A maioria das restrições origina-se de políticas e não de fatores físicos. As restrições de políticas são mais complexas de serem identificadas e tratadas.
- Idéias não são soluções. As melhores idéias não terão potencial até que sejam implantadas. A maioria das idéias falha ainda no processo de implantação.

Goldratt&Cox (2003) comparam uma linha de produção a uma caminhada de um grupo de escoteiros em fila indiana. Cada um tem seu ritmo de caminhada, assim como os processos de produção têm sua capacidade. Caso o escoteiro mais rápido seja colocado na frente, o grupo tende a se dispersar. Uma das soluções é agrupar os escoteiros de forma que o mais lento coordene a caminhada. Outra solução seria amarrar os escoteiros com uma corda para fazê-los

andar na mesma cadência, no ritmo do mais lento, ou seja, no ritmo da restrição. O desafio é reduzir a dispersão das atividades sem aumentar o tempo total para completar o ciclo.

2.10.7 Custo de propriedade

As áreas de Engenharia de Produção e Tecnologia da Informação compartilharam alguns conceitos, um deles foi o custo de propriedade. Ele ficou conhecido do público consumidor através do método Toyota de produção, mas foi concebido pelo Gartner Group em 1987 para definir critérios de custeio de sistemas informatizados considerando aquisição inicial de equipamento, reparos, manutenções, atualizações, serviços, suporte, rede, seguros, licenças de uso de software, treinamento, além de outras despesas.

2.11 Comparação entre as teorias industriais e as de T.I

2.11.1 Filas como base dos computadores multi-usuários

Nem sempre as filas dentro de um computador foram consideradas como problema. Pelo contrário, elas foram muito bem exploradas como fator motivador de desenvolvimento de sistemas operacionais multitarefa. *A idéia nasceu no início dos anos 60, quando os pesquisadores do MIT observaram que as pessoas ficavam aguardando a liberação da máquina que atendia um único usuário por vez. Enquanto isso, havia processador ocioso esperando o término de uma operação de leitura ou gravação. Essa ociosidade poderia ser aproveitada para diminuir a fila de espera pelo equipamento. Para colocar essa idéia na prática, o sistema operacional simulava múltiplas cópias de máquinas. Cada cópia ou "máquina virtual" era controlada como se fosse uma máquina real com sistema operacional próprio*. [CREASY, 1981]

As filas que funcionavam como indicadores de problema, passaram a ser importantes no conceito do novo sistema operacional, feito para evitar o desperdício de recursos. *A capacidade do computador foi dividida em frações computacionais que eram atribuídas aos programas que estivessem na fila de espera de um determinado recurso. Alguns estavam na fila de espera pelo processador, que depois de utilizá-lo, passavam para a fila do recurso seguinte. O sistema era organizado com filas para cada recurso disponível: processador, memória, disco, fita magnética, impressão e o equipamento de tele-tipo onde era realizada a intervenção manual. Cada programa passava por essas filas quantas vezes fossem necessárias até encerrar seu ciclo de processamento.* [CORBATÓ ET AL, 1962]

O conceito de "máquinas virtuais" continuou sendo utilizado pela IBM em sua linha de *mainframes* z/9, lançada no ano de 2006. Ele tornou possível abrigar entre dezenas e milhares de servidores Linux no interior de um único *mainframe* através do sistema operacional z/VM.

2.11.2 Filas como base dos sistemas de discos

Existe uma linha de pensamento que coloca as atividades dos discos em segundo plano argumentando que o processador sempre estará realizando tarefas enquanto aguarda o retorno de uma solicitação de leitura/gravação. Entretanto, esta afirmação pode ser contestadas de algumas maneiras. Uma delas é mostrando o erro de centralizar a administração sobre as atividades dos processadores e esquecer que os usuários se importam com o tempo de resposta. Outra é lembrando que o desempenho dos processadores vem dobrando a cada 18 meses nos últimos 15 anos e já não é mais o problema que era antes. Por fim, os discos têm sua própria teoria de desempenho: a teoria das filas, que equilibra o *throughput* em relação ao tempo de resposta. [HENNESSY&PATTERSON, 2003, p. 678-679]

A Figura 2.14 representa o diagrama de um computador de grande porte utilizando discos RAID elaborado por Menascé&Almeida (2002, p. 83). É possível observar a quantidade de filas necessárias para administrar as operações relativas ao armazenamento de dados. No modelo apresentado, que utilizou formatação de disco RAID com 5 discos, uma operação de acesso a um dado enfrenta 7 filas em seu percurso. Essa quantidade indica o grau de importância do conhecimento da Teoria das Filas para a redução dos tempos de processos da Tecnologia da Informação.

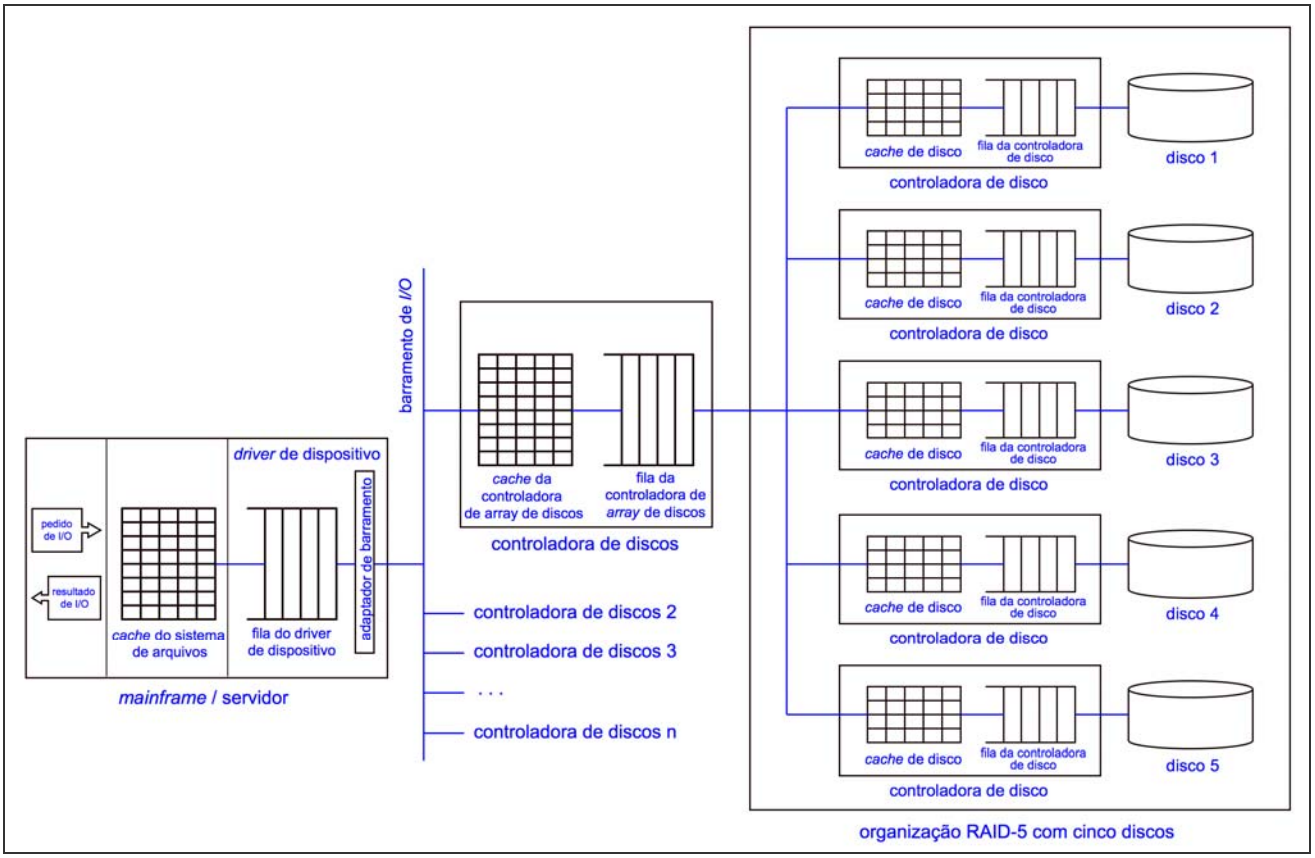

Figura 2.14 – Quantidade de filas relacionadas ao acesso a discos RAID

2.11.3 Filas como base da redução do tempo de acesso a discos

As operações de leituras e gravações ficaram mais sofisticadas e nos computadores de grande porte passaram a ser realizadas pelos sistemas de armazenamento de dados. As filas continuam sendo utilizadas como solução para reduzir os tempos de acesso aos discos que implementam esses sistemas de armazenamento de dados.

Nos modelos de discos mais antigos, as requisições eram organizadas por ordem de chegada e as operações físicas de leituras e gravações obedeciam essa ordem. Com o aumento da densidade dos discos, os dados passaram a ficar mais próximos e o modelo de organização tradicional "trilhas-cilindros", onde as trilhas ficavam uma abaixo da outra formando um cilindro, deixou de existir fisicamente.

A nova forma de organização dos discos permite melhor desempenho e o modelo de fila foi modificado para permitir o melhor aproveitamento do recurso.

Em modelos de discos que oferecem melhor desempenho, as solicitações são colocadas em uma fila e lá permanecem até que ocorra a operação física. Nesse momento, a fila é reorganizada para que seja obtido o menor trajeto entre os endereços das leituras e gravações solicitadas. Esse procedimento é semelhante ao utilizado na logística dos serviços de entregas e retiradas de mercadorias em clientes e fornecedores.

A Figura 2.15 compara duas situações onde foram atendidos 4 pedidos de leitura. Os endereços dos registros solicitados ocorreram nessa ordem: 724, 100, 9987 e 26. Foram comparados dois algoritmos capazes de localizá-los e disponibilizá-los aos solicitantes: (1) as setas externas indicam o funcionamento dos discos que utilizam um conceito tradicional de filas. As solicitações são recebidas pela ordem de chegada e no início da operação física essa fila é reorganizada pela ordem crescente de endereços no interior do disco. Nesse conceito, foram necessárias 4 operações completas de busca-latência-transferência para atender 4 solicitações de leituras ou gravações em um ambiente operacional multi-usuário, (2) a parte interna da figura representa a mesma tarefa realizada pelos discos que exploram a Teorias das Filas de modo mais eficiente em relação à tecnologia de discos que utilizam. As solicitações são recebidas pela ordem de chegada e no início da operação física essa fila é reorganizada para que seja obtido o melhor trajeto da cabeça de leitura e gravação. No exemplo, uma operação de ¾ de volta foi capaz de substituir 4 realizadas por modelos de menor desempenho. [ANDERSON, 2003], [HENNESSY&PATTERSON, 2007, p. 400-401]

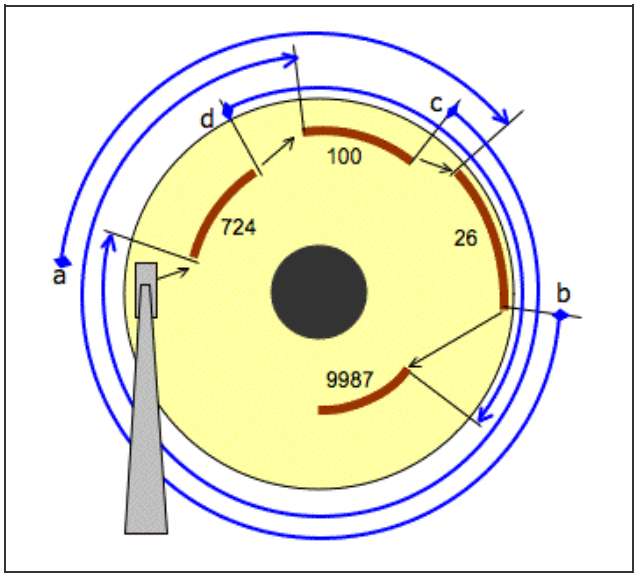

Figura 2.15 – Comparação entre modelos de filas utilizadas pelos discos

2.11.4 Conceito de processamento em lotes

O conceito de processamento de lotes utilizado neste trabalho identifica que uma série de registros serão processados em seqüência por um mesmo aplicativo, onde as decisões estão previstas em suas lógicas. Portanto, não existe interação do usuário como acontece nos processamentos *on-line*.

O custo de processamento dos dados em lotes é inferior aos processos *on-line* por permitir maior nível de compartilhamento de recursos. Os computadores utilizados para atender a demanda *on-line* precisam ser configurados para o pico de utilização, que acontece durante o horário comercial. Fora desse período existe o risco de haver ociosidade do equipamento. Nos processamentos em lote, a demanda pode ser ajustada por um profissional denominado "controlador de produção" que distribui a carga de trabalho para evitar ociosidade no *mainframe*.

Cada vez que um aplicativo tipo *batch* entra em execução, os arquivos utilizados são divididos em lotes, que recebem o nome de blocos. Esse mecanismo é utilizado para permitir o uso equilibrado de recursos computacionais entre os diversos programas em execução. Como o número de processadores é menor que o de aplicativos em execução, o sistema operacional utiliza critérios de prioridades para atender cada um deles. Um desses critérios leva em conta os momentos de leituras e gravações de um aplicativo para ceder o processador a outro. [STALLINGS, 2005, p. 52-55]

Nos processamentos *batch*, os acessos aos discos ocorrem "por pedido" como na manufatura por pedido. Nesse tipo de processo industrial, existe o preparo da linha antes que a produção seja iniciada. A eficiência é dada pela relação entre o tempo de produção e o tempo de preparação, conhecido como tempo de *setup*. O próximo passo deste trabalho foi fundamentar esse conceito.

2.11.5 Aplicação do conceito de *setup* em operações de leituras e gravações

O tempo de *setup* de um processo industrial, que prepara uma linha de produção para uma determinada tarefa, pode ser comparado às operações de busca e latência necessárias às leituras e gravações em discos magnéticos. Essas duas operações têm a finalidade de localizar um endereço no disco para que uma operação de leitura ou gravação seja efetivada. A operação de busca localiza a trilha especificada. A operação de latência aguarda a chegada do início dessa

trilha para iniciar a transferência de dados para o aplicativo. Quanto maior o lote de dados enviado ao aplicativo, menor será sua ociosidade e menor seu tempo de execução.

A Figura 2.16 exemplifica o tempo economizado em um processo computacional quando os dados são acessados em lotes maiores. No item (a) estão representados os acessos individuais aos registros, onde a cada acesso é realizada uma operação de preparo no braço mecânico que lê e grava dados no disco. A cada registro lido ou gravado, o braço mecânico sofre deslocamento de uma posição até outra onde está o endereço do próximo registro solicitado. Quando a quantidade de registros por bloco é pequena, é grande a quantidade das operações de *setup*. Em um disco magnético essas operações são compostas por: (1) tempo de busca, que significa o tempo necessário para localizar a trilha onde está o dado solicitado e (2) tempo de latência, que significa o tempo de espera pelo início da trilha onde está o dado solicitado. No item (b) estão representados os registros agrupados em blocos. Essa forma de organização dos dados permite que uma quantidade maior de registros seja tratada em uma mesma operação, exigindo uma menor quantidade de operações mecânicas do braço de leituras e gravações. Ao final, é observada a redução no tempo total do processo.

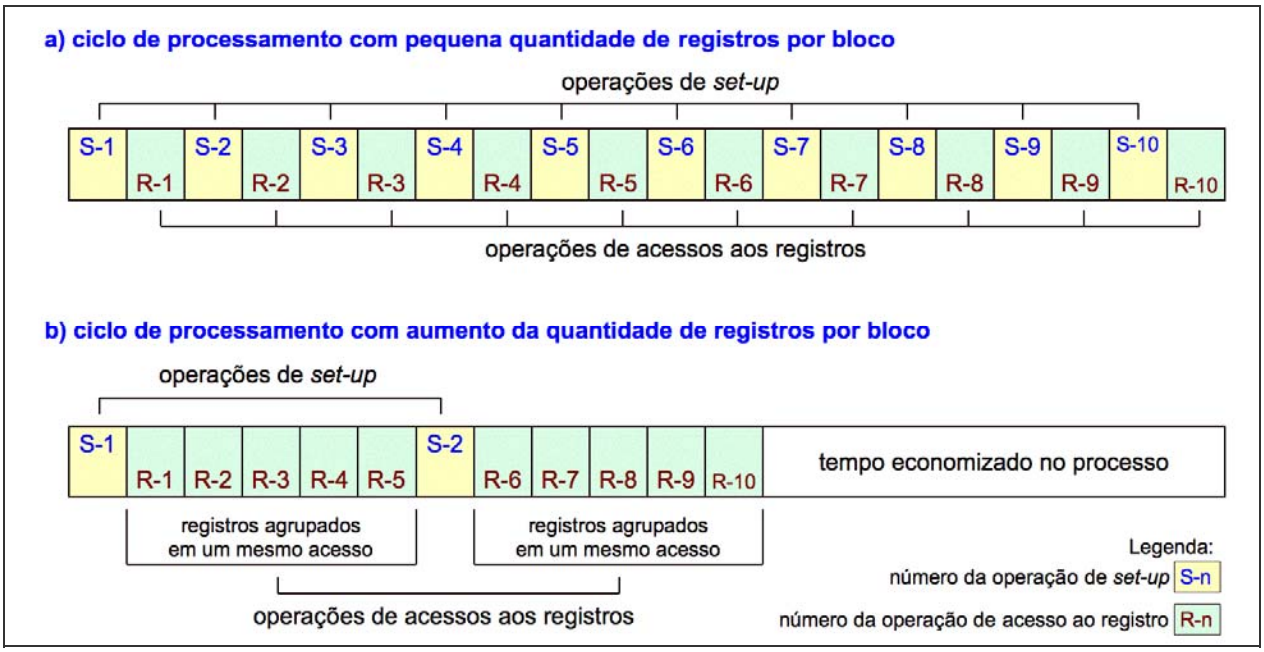

Figura 2.16 – Relação entre tempos de processos X quantidade de registros por bloco

A Tabela 2.2 apresenta os tempos de serviço de um disco magnético sob o conceito de tempos de preparação e produção. É possível observar que o tempo de preparação (12,37 ms) é

maior que o tempo da transferência dos dados lidos ou gravados (2,36 ms). Por isso, a maior contribuição para a redução do tempo de um processo computacional virá da redução do tempo de *setup* ou preparação, representado pelos tempos de busca e latência.

Tabela 2.2 – Analogia entre tempos de operações industriais e operações em discos

Tempos de uma operação industrial	Tempos de uma operação em disco magnético	Tempo da operação em discos EMC
Tempo de Preparação ou *setup*	Tempo de busca	8,2 ms
	Tempo de latência	4,17 ms
Tempo de Produção	Tempo de transferência (dados lidos ou gravados)	2,36 ms (para 65 mil *bytes*)

A Tabela 2.3 apresenta as fórmulas dos cálculos de estimativa de tempo necessário para a gravação de 1 milhão de registros. Foram considerados os seguintes valores: (1) a quantidade de registros em um mesmo bloco variou entre 1 e 128, (2) registros com tamanho de 512 caracteres, (3) disco com tempo de busca de 8,2 ms, (4) tempo de latência de 4,17 ms e (5) taxa de transferência 27,8 MB/seg. É possível observar que, conforme aumenta a quantidade de registros por bloco, aumenta o tempo da operação de gravação, entretanto diminui a quantidade de operações necessárias para realizar a gravação do arquivo completo. Apesar do aumento do tamanho do bloco causar aumento no tempo de duração das operações individuais em disco, existe ganho global com a opção de lotes maiores.

Tabela 2.3 – Tempos estimados para gravações X tamanho do lote de dados

Quantidade de registros de 512 caracteres por bloco	quantidade de operações necessárias para gravar 1 milhão de registros conforme o arranjo de blocagem		tempo individual das operações (busca + latência) + transferência (miliseg.)		tempo total das operações (busca + latência) + transferência (seg.)	tempo total (hh:mm:ss)
1	1.000.000	X	(8,2+4,17) + 0,01842	=	12.370 + 18,4	3:26:28
2	500.000	X	(8,2+4,17) + 0,03683	=	6.185 + 18,4	1:43:23
4	250.000	X	(8,2+4,17) + 0,07367	=	3.093 + 18,4	0:51:51
8	125.000	X	(8,2+4,17) + 0,14734	=	1.546 + 18,4	0:25:05
16	62.500	X	(8,2+4,17) + 0,29468	=	773 + 18,4	0:13:12
32	31.250	X	(8,2+4,17) + 0,58935	=	387 + 18,4	0:06:45
64	15.625	X	(8,2+4,17) + 1,17871	=	193 + 18,4	0:03:32
128	7.813	X	(8,2+4,17) + 2,35741	=	97 + 18,4	0:01:55

A Figura 2.17 apresenta os resultados apresentados na Tabela 2.3 utilizando o conceito de French (1982). Para sua construção, os tempos de operações foram separados em duas categorias: (1) o tempo considerado de *setup* ou de preparação, composto pelo tempo busca de um endereço de disco mais o tempo de espera pelo início da trilha e (2) o tempo das operações, onde é realizada a transferência dos dados. Esta forma de apresentação permitiu observar que, enquanto o tempo total das operações de busca e latência são inversamente proporcionais à quantidade de registros por bloco, não há variação no tempo necessário para transferência de dados entre memória e disco. Independente da quantidade de registros por bloco, o tempo das operações de transferência de dados foi uma constante de 18,4 seg., resultando em uma linha paralela ao eixo x.

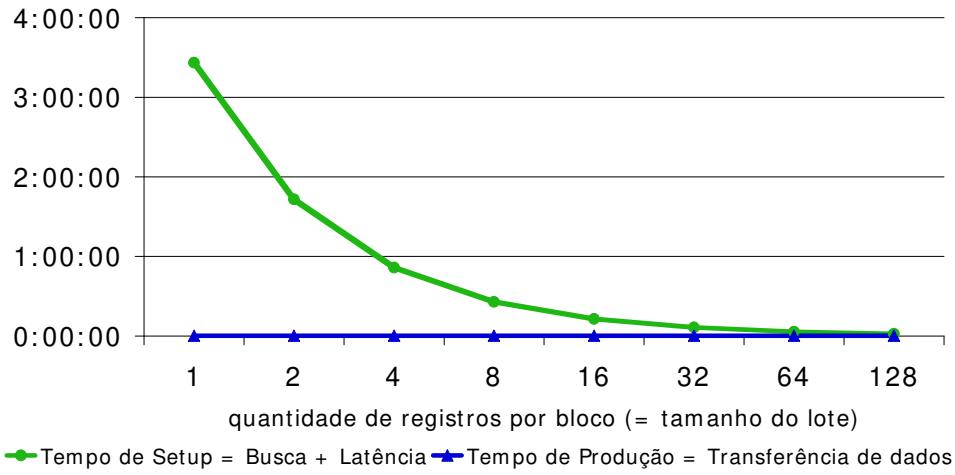

Figura 2.17 – Tempos estimados para gravação de 1 milhão de registros

2.11.6 A organização do estoque e a organização dos dados

O conceito de organização de estoque utilizado nas fábricas foi útil para a redução do tempo de resposta em computadores de grande porte. Ele tornou possível organizar os dados no interior de um disco magnético semelhante ao modo como as mercadorias são colocadas no interior de um depósito. As mercadorias com maior taxa de movimentação ficam próximas à porta, facilitando sua locomoção e reduzindo o tempo da tarefa de transporte. As mercadorias menos solicitadas são colocadas mais para o fundo, têm maior tempo de transporte mas menor número de viagens, resultando em ganho global para os processos de manufatura.

A Figura 2.18 ilustra a organização de um depósito onde os materiais de maior taxa de movimentação ficam próximos da porta, reduzindo o tempo necessário para a tarefa de transporte.

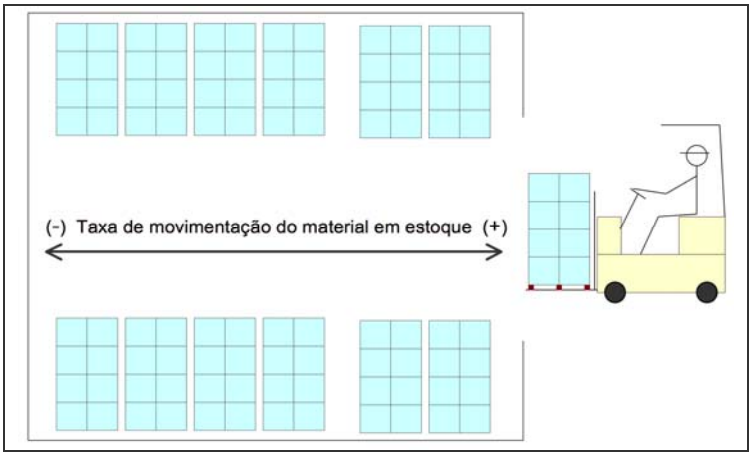

Figura 2.18 – Colocação do material em estoque em função da taxa de movimentação

Este conceito de organização utilizado pelo setor industrial é capaz de reduzir o tempo dos aplicativos em computadores de grande porte, quando aplicado à forma de organização dos dados no interior de um disco magnético. A proximidade dos dados em relação às bordas, diminui o tempo de serviço do braço de leituras e gravações, reduz o tempo de acesso aos dados e conseqüentemente, reduz o tempo de execução de um aplicativo.

2.11.7 Produção modular e tarefas paralelas

A redução do tempo de tarefas através do paralelismo de tarefas, além de ter sido utilizada na linha de produção de *mainframes* [STARR, 1965] e depois na indústria automobilística [GRAZIADIO, 2004], também foi utilizada por Amdahl (1967) na conceituação dos sistemas operacionais dos modelos IBM/360 e IBM/370. Amdahl (1967) demonstrou quanto um processamento pode ser acelerado à medida que processadores paralelos são adicionados ao computador. O conceito de paralelismo também foi utilizado na construção dos discos RAID por Patterson, Gibson e Katz [PATTERSON ET AL, 1988], permitindo melhor tempo de resposta que os discos convencionais.

O paralelismo continua sendo utilizado tanto na concepção de discos RAID como na configuração de computadores de grande porte. Entretanto, existe um ponto onde a capacidade de aceleração de tarefas proporcionadas pelo paralelismo começa a declinar.

Esse declínio é um corolário importante da Lei de Amdahl denominado Lei do Rendimento Decrescente: "A aceleração obtida por uma melhoria adicional no desempenho diminui à medida que as melhorias são adicionadas". [HENNESSY&PATTERSON, 2007, p. 40]

A Figura 2.19 representa o resultado da Lei do Rendimento Decrescente para computadores de grande porte, de acordo com o laboratório de pesquisas da IBM. "O aumento da quantidade de recursos provoca sobrecarga no gerenciamento, existe um ponto onde esse aumento deixa de trazer ganhos. Nos *mainframes* das linhas z/800 e z/9, a quantidade máxima de processadores compartilhando a mesma memória e o mesmo sistema operacional z/OS é atualmente 24. [IBM, 2005, p. 329]

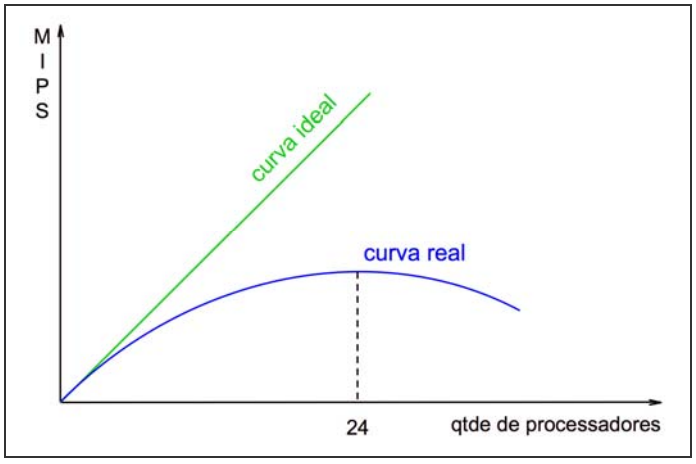

Figura 2.19 – Lei do Rendimento Decrescente em *mainframes* IBM

A revisão dos conceitos de Produção Modular e Tarefas Paralelas foi importante para este trabalho pois permitiu observar formas diferentes de aplicação de conceitos semelhantes, além de ter auxiliado concluir que o paralelismo de tarefas tem um limite, conforme demonstrado por pesquisas realizadas pela IBM (2005).

Capítulo 3

Materiais e Métodos

A verdadeira viagem do descobrimento não consiste em
buscar novas paisagens, mas novos olhares.
(Marcel Proust)[4]

3.1 Introdução

Considerações preliminares para a montagem da metodologia

Hopper [apud O'CONNOR&ROBERTSON, 1999] e Knuth (1998) enfatizaram que o desempenho de um aplicativo está relacionado à qualidade e à clareza de escrita de seu código. Entretanto, durante o desenvolvimento deste trabalho, foi observado que mesmo alguns programas com código reduzido ou que já haviam passado por processos de revisão, continuavam com tempo de resposta alto. Um exemplo dessa observação foi registrado na Figura 1.4 (Decomposição do tempo de transações observadas em um banco). Para a criação daquele gráfico foram analisadas 25 milhões de transações bancárias. O tempo médio de resposta foi de 2,69 seg., entretanto apenas 0,01 seg. foram utilizados para processamento. Apenas 0,5% do tempo de resposta foi causado pelo uso de processador, o restante foi espera por discos.

Esta constatação indica que os métodos tradicionais de melhoria de tempo de resposta, focados na redução do uso de processador, já não apresentam os resultados que apresentavam no passado.

As motivações para encontrar um método alternativo para melhorar o tempo de resposta de aplicativos são reforçadas pelas pesquisas de Denning (2005), que apontam que apenas 20% do código de um aplicativo são utilizados com freqüência e outros 80% são utilizados eventualmente. Este tema foi desenvolvido no capítulo Revisão Bibliográfica deste trabalho e concluiu que modificações realizadas na porção dos 80% menos utilizados do aplicativo colocam

em risco o negócio da empresa, pois precisam ser validadas por rotinas de testes mais precisas que as utilizadas nas situações cotidianas.

Também foi observado, nos ambientes de estudo onde este trabalho foi desenvolvido, que os computadores de grande processam milhares de programas simultaneamente, competindo por recursos com diferentes graduações de desempenho que variam de nanosegundo a milisegundo. Esses ambientes operacionais contavam com tecnologia atualizada de bancos de dados, processadores, discos e memória cache, mesmo assim havia tempo de resposta insatisfatório.

Esses fatos fundamentaram a construção de um método alternativo para localizar as causas de altos tempos de resposta que não considerasse apenas a lógica do aplicativo onde a demora foi constatada, pois ele pode estar sendo causado por fatores externos. O próximo passo deste trabalho foi fundamentar a composição do tempo de resposta para localizar os integrantes de maior influência.

3.2 Composição do tempo de resposta

A Figura 3.1 ilustra a composição do tempo percebido por um usuário para obter a resposta de uma requisição de serviços de Tecnologia da Informação. O tempo de resposta ponta a ponta pode ser dividido em dois componentes principais: tempo de rede externa e tempo no local de processamento. (1) O tempo de rede externa compreende o tempo gasto pelo tráfego das mensagens entre o local de trabalho do usuário e o local de processamento. Esse tempo pode ser decomposto em outros dois componentes: tempo de latência e tempo de transmissão. Latência indica o número de vezes de ida e volta, são as tarefas envolvidas na troca de mensagens entre a estação do usuário e o local de processamento. A latência é uma função da natureza do protocolo utilizado. O tempo de transmissão é o tempo necessário para transmitir todos os *bytes* entre a estação do usuário e o local de processamento. (2) O tempo no local de processamento pode ser decomposto em dois componentes principais: o tempo de serviço e o tempo de fila. Tempo de serviço é o período de tempo durante o qual um pedido está recebendo serviço de um recurso, como processador, disco e rede local. Tempo de fila é o tempo gasto aguardando para ter acesso a um desses recursos. [MENASCÉ&ALMEIDA, 2002, p. 68-69]

[4] apud [Fainguelernt&Nunes, 2006]

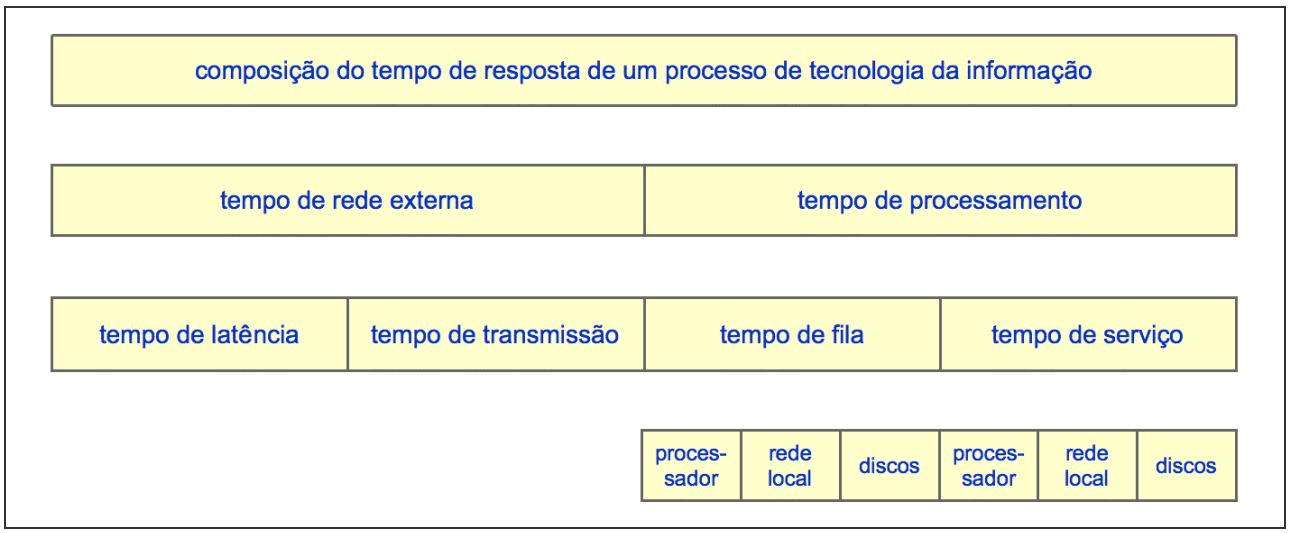

Figura 3.1 – Composição do tempo de resposta de um processo de tecnologia da informação

Uma requisição pode ter que visitar um recurso mais de uma vez antes que seja concluída, por exemplo, mais de uma leitura ou gravação em disco ou muitas visitas ao processador, resultando em uma soma de tempos entre receber o recurso e esperar por ele.

A Equação 3.1 apresenta a composição numérica do tempo de resposta (R) de um processo da Tecnologia da Informação. O tempo de serviço (S) é aquele onde a requisição recebe tratamento do processador, discos e rede local ou linha de comunicação. A notação S_i^j é usada para indicar o tempo de serviço no recurso *i* durante a *j-ésima* visita ao recurso. O tempo de fila (F) é aquele onde a requisição aguarda para receber o atendimento de um recurso. A notação F_i^j representa o tempo gasto por um pedido aguardando para ter acesso ao recurso *i* durante a *j-ésima* visita. [MENASCÉ&ALMEIDA, 2002, p. 70]

Equação 3.1 – Composição do tempo de resposta de um processo de tecnologia da informação

$$R = \sum_{\substack{\text{visitas } j \\ \text{recursos } i}} \left(S_i^j + F_i^j \right)$$

Diante da definição e da fórmula de composição do tempo de resposta, o próximo passo foi identificar os pontos onde eram adicionadas as maiores proporções de tempo ao processo. Para isso foram utilizados conceitos de Pensamento Enxuto.

A revisão dos conceitos de Pensamento Enxuto permitiu fundamentar o mapeamento de processos de Tecnologia da Informação, auxiliou na localização de pontos que agregavam tempo às transações *on-line* e aos aplicativos *batch*, pontos que geravam desperdícios ou causavam atrasos no processamento. Dessa forma foi possível localizar pontos com maior margem de sucesso para a redução do tempo de um processamento.

A coleta e a organização dos dados foram realizadas com auxílio do *software Mind Map* e o resultado final foi utilizado para construir o mapeamento das etapas de um processo de T.I., desenvolvido no próximo item deste trabalho.

3.3 Mapeamento das etapas de um processo de T.I.

A partir dos conceitos de Menascé&Almeida (2002, p. 68-70) e dos conceitos de *Lean Manufacturing* de Womack&Jones (2004), foi possível criar um método para identificar os componentes do tempo de resposta de um processo de Tecnologia da Informação.

A Tabela 3.1 relaciona os itens componentes de um processo de Tecnologia da Informação. Foram agrupados em 3 etapas: entrada, processamento e saída. O objetivo desta tabela é localizar os componentes e tarefas que mais contribuem para o aumento de tempo de resposta.

Tabela 3.1 – Mapeamento das etapas de um processo de Tecnologia da Informação

	Responsável	Tarefa	Unidade	Acesso a disco
ENTRADA	Usuário	Digitação para solicitar ou fornecer informações	min.	
	Rede	Tempo de fila	ms	
		Tempo de latência	ms	
		Tempo de transmissão	ms	
		Faz acesso a discos para registrar a tarefa realizada	ms	sim
	Gerenciador de transações	Recebe a solicitação do usuário	ns	
		Administra fila de solicitações	ns	
		Administra carga de trabalho do aplicativo e pode colocar novas cópias para auxiliar	ns	
		Encaminha a solicitação ao aplicativo responsável	ns	
		Faz acesso a discos para registrar a tarefa realizada	ms	sim
PROCESSAMENTO	Memória [Stallings, 2005p, p.103]	Tempo de acesso	ns	
		Tempo de ciclo	ns	
		Taxa de transferência	ns	
		Faz acesso a disco para virtualizar memória	ms	sim
	Processador	Tempo de fila	ns	
		Tempo de latência	ns	
		Tempo de processamento	ns	
	Gerenciador de Banco de Dados	Valida regras de integridade dos dados	ns	
		Faz acesso a disco para administrar endereços	ms	sim
	Criptografia	Valida regras estabelecidas para a operação	ns	
		Utiliza processador especializado em criptografia	ns	
	Operações de leituras e gravações [Stallings, 2006, p. 177]	Tempo de espera pelo dispositivo	ms	sim
		Tempo de espera pelo canal	ms	sim
		Tempo de busca	ms	sim
		Tempo de latência rotacional	ms	sim
		Tempo de transferência de dados	ms	sim
SAÍDA	Gerenciador de transações	Recebe a solicitação do aplicativo	ns	
		Administra fila de solicitações	ns	
		Administra carga de trabalho do aplicativo e pode retirar cópias ociosas	ns	
		Encaminha a solicitação ao terminal solicitante	ns	
		Faz acesso a discos para registrar a tarefa realizada	ms	sim
	Rede	Tempo de fila	ms	
		Tempo de latência	ms	
		Tempo de transmissão	ms	
		Faz acesso a discos para registrar a tarefa realizada	ms	sim
	Usuário	Recebe o resultado de sua transação	ms	

Após a observação que os discos são solicitados em todas as atividades exceto as realizadas pelo processador, o próximo passo foi fundamentar a metodologia para que eles não representassem restrições ao desempenho dos processos de Tecnologia da Informação.

3.4 Aplicação do conceito de Teoria das Restrições

A Teoria das Restrições considera que a produção máxima de um sistema é definida pela capacidade de seu gargalo, por esse motivo deve-se buscar sua eliminação para agilizar o sistema.

Os gargalos de uma linha de produção são identificados visualmente pelo material semi-acabado acumulado ao seu redor. A eficiência, paralisação e uso desse recurso são freqüentemente monitorados para evitar perdas no restante no processo de produção. [GOLDRATT&COX, 2003]

Num processo industrial, os estudos para eliminação de gargalos são acompanhados de análises financeiras. A existência de gargalo implica em custos causados pela espera ou pelo estoque intermediário para evitar a espera. Entretanto, existe o risco de substituir os custos causados pelos gargalos por alguma outra forma de custo, não observada enquanto os gargalos eram o centro das atenções. A decisão de eliminá-los está intimamente ligada ao custo do processo. Sendo assim, é preciso definir o custo do recurso gargalo dentro da atividade para escolher a melhor forma de tratamento.

Para Michalsky apud [NUNES, 2004], a Teoria das Restrições tem grande oportunidade de aplicação tanto em produção como em serviços e novos projetos, visto que todos os casos possuem algum tipo de gargalo e a metodologia aplica-se perfeitamente nesses cenários, sejam as restrições de ordem pessoal ou material. Um computador de grande porte é um sistema de produção onde a execução das tarefas ocorre pelo processamento de aplicativos, portanto, a teoria das restrições é aplicável a este ambiente.

Conforme foi constatado no capítulo de Revisão Bibliográfica da Teoria das Restrições, Goldratt&Cox (2003) comparam uma linha de produção a uma caminhada de um grupo de escoteiros em fila indiana, onde o mais lento dita a velocidade dos demais. Nos processos de Tecnologia da Informação, os discos, por serem os mais lentos, ditarão o tempo de resposta final. Além disso, existirão discos mais lentos que outros, não pela diferença de tecnologia, mas pela diferença no tamanho da fila que ocorre em função do grau de utilização.

O próximo passo deste trabalho foi fundamentar a influência das filas em discos sobre o tempo de processo realizado por um computador de grande porte.

3.5 Descrição do ambiente de estudo 1

O primeiro objeto de estudo deste trabalho foi o conjunto de aplicativos *batch* do ambiente de processamento noturno de compensação bancária em uma instituição formada por 256 agências. Foi utilizado um *mainframe* NX-5820 da *Unisys*, com capacidade de processamento de 780 MIPS (milhões de instruções por segundo) distribuído em 8 processadores. Os dados armazenados totalizaram 1 *Terabyte* e tinham acesso realizado através de 2 processadores de leituras e gravações. A quantidade de aplicativos implantados esteve próximo de 15 mil programas escritos em sua maioria na linguagem COBOL.

Em especial, para aplicações bancárias pode-se distinguir dois tipos de tarefas: as que devem ser executadas *on-line* e as que devem ser programadas para execução posterior. Dentro do ciclo de compensação bancária existe um período conhecido pelo jargão de "processamento noturno". Ele têm início entre 20h e 22h, com prazo para término às 6 horas da manhã seguinte. Nesse prazo, todas as movimentações bancárias realizadas durante o dia precisavam estar consolidadas, quer tenham sido originadas nas agências, nos caixas automáticos ou através da internet.

A Figura 3.2 representa o ciclo do processo de compensação bancária destacando a janela noturna, período onde são realizados os reprocessamentos necessários quando há divergência no fechamento contábil, ou outro problema que afete a integridade das informações dos sistemas.

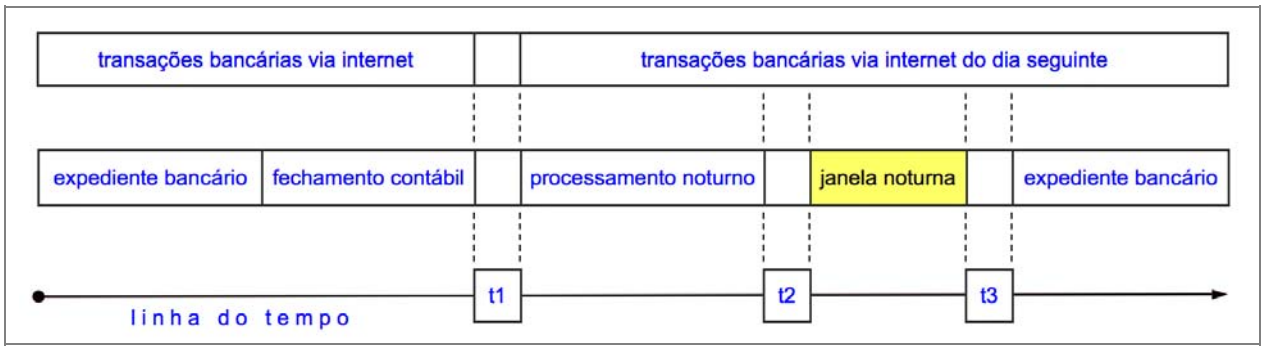

Figura 3.2 – Ciclo do processo de compensação bancária

Tarefas realizadas no período noturno:

1. recebimento dos arquivos com o movimento das agências (t1);
2. recebimento dos arquivos com o movimento das transações via internet (t1);

3. consolidação do movimento das agências (entre t1 e t2);
4. consolidação do movimento das transações dos caixas automáticos (entre t1 e t2);
5. consolidação do movimento das transações via internet (entre t1 e t2);
6. atualizações dos bancos de dados dos produtos bancários (entre t1 e t2);
7. fechamento contábil das consolidações e atualizações (entre t1 e t2);
8. aprovação dos arquivos atualizados (entre t1 e t2);
9. liberação dos dados atualizados para as transações via internet (t2);
10. liberação dos dados atualizados para as agências (t2);
11. abertura das agências (t3).

Estas tarefas devem ser realizadas com o objetivo de melhoria do nível de atendimento ao cliente e algumas características identificadas neste processo são:

1 - quanto menor o intervalo entre t1 e t2 maior o nível de serviço prestado e menor o custo operacional representado pelo tempo de processamento;

2 - quanto mais tarde for o instante t1 maior o limite para pagamentos via internet;

3 - quanto maior o intervalo entre t2 e t3, maior o limite de segurança para reprocessamentos.

3.6 Mapeamento do processo *batch*

O primeiro passo para a construção da metodologia necessária para melhorar o tempo de um processo *batch* foi elaborar o seu mapeamento.

A Figura 3.3 representa o mapeamento de um processo batch. Foi criado a partir do conceito proposto por Menascé&Almeida (2002, p. 68-69) que divide o tempo de um processo informatizado em tempo de rede externa e tempo de processamento. Para o mapeamento de processos *batch* foi desconsiderado o tempo de rede externa, uma vez que não há comunicação direta com o usuário final. Conseqüentemente, o tempo de processo de um aplicativo *batch* é composto do tempo de fila e do tempo de serviço do recurso solicitado.

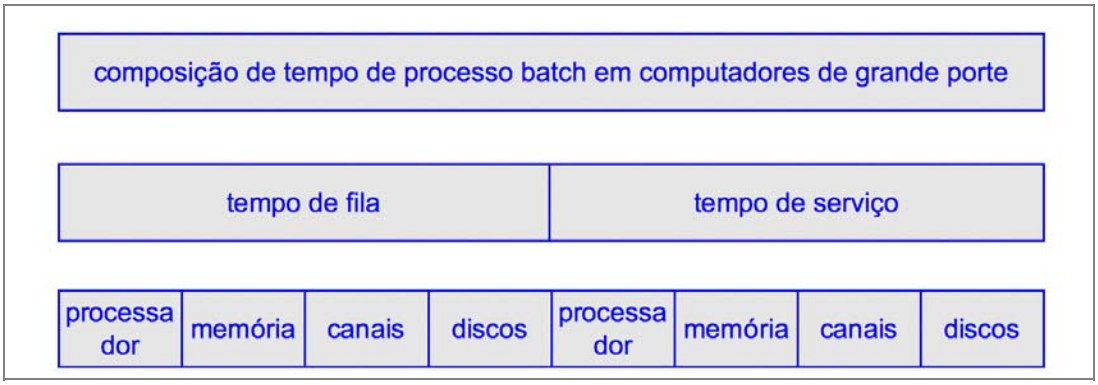

Figura 3.3 – Composição do tempo de processos *batch*

O mapeamento proposto separa os itens a serem analisados em dois grupos: os tempos de serviços realizados pelos recursos computacionais e o tempo de fila que os aplicativos aguardam pela liberação desses recursos. A partir das pesquisas realizadas por Hannessy&Patterson (2007), que apontam as diferenças nas velocidade de evolução do desempenho dos processadores e discos, as filas encontradas nos discos oferecem maior impacto nos tempos de processos computacionais por serem os recursos mais lentos.

Como o tempo de serviços dos discos é uma característica de fabricação, o próximo passo na construção da metodologia foi reduzir o tempo de permanência em fila de espera.

3.7 Aplicação do conceito de Teoria das Filas

Segundo Prado (1999) o estudo das filas em computadores surgiu como uma área de muita importância nas últimas décadas. Existem filas de programas esperando por espaço na memória ou para serem atendidos pelo processador, ou para buscar um registro de dados em um disco magnético.

Os métodos tradicionais de análise de desempenho recomendam que seja analisada a capacidade de utilização do processador, memória e discos. Quando são atingidos os limites determinados pelos fabricantes, a solução mais simples é injetar mais recursos no sistema. [LABOR&NULL, 2006, p. 328]. Este trabalho propõe que antes do incremento de novos recursos seja analisada a possibilidade de redistribuição da carga de trabalho dentre os já existentes. Essa redistribuição é viável quando existir diferentes níveis de filas entre recursos semelhantes.

A Figura 3.4 permite observar a variação do tempo de processo de um dos aplicativos do sistema Fundos Diários causado pela variação do tamanho da fila de espera por recursos. O tempos de utilização de recursos computacionais foram acompanhados durante um mês. Os dados foram coletados através do arquivo *log* e do utilitário *SYSTEM/SUMLOG* da *Unisys,* fabricante do *mainframe* utilizado. O aplicativo tinha a responsabilidade de fazer o cálculo de rentabilidade do sistema Fundos Diários de Investimentos. Suas necessidades de processador, memória e discos mantinham-se as mesmas em todos os dias do mês. Entretanto, seu desempenho era afetado pela demanda de recursos computacionais durante os fechamentos mensais dos demais sistemas da instituição financeira.

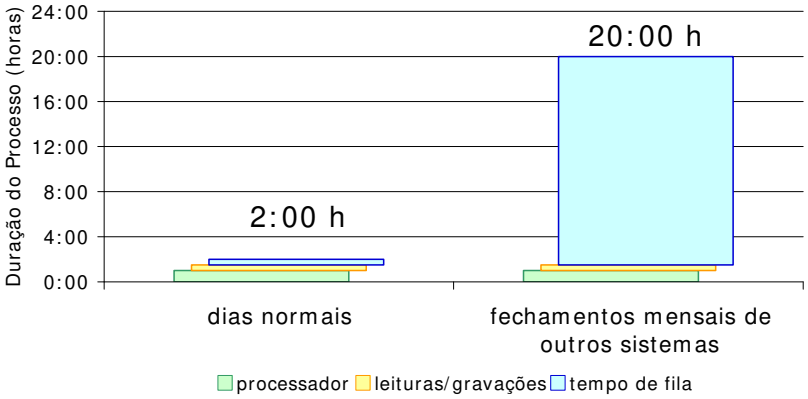

Figura 3.4 – Variação do tempo de processamento do Fundos Diários

O próximo passo foi a realização da coleta de dados no sistema de monitoramento do ambiente operacional, seguida da análise dos cenários dos dias normais e fechamentos.

A Figura 3.5 ilustra a variação dos tamanhos de filas nos discos utilizados pelo sistema Fundos Diários em dias normais e durante os fechamentos mensais dos demais sistemas. Os dados foram coletados durante um mês através do sistema de monitoramento de recursos denominado *Team Quest*. É possível observar que houve crescimento do tamanho das filas, durante os fechamentos mensais, devido ao aumento da demanda de acessos a discos.

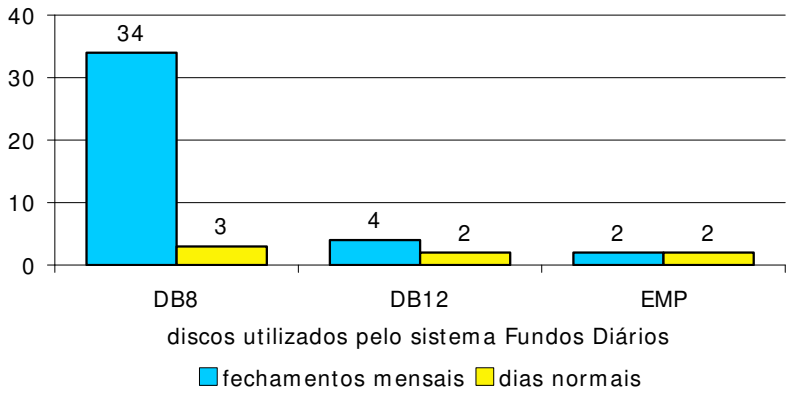

Figura 3.5 – Variação do tamanho das filas nos discos do Fundos Diários

A interpretação dada aos números obtidos foi: o sistema Fundos Diários tinha tempo de processamento igual a duas horas quando: (1) a fila no disco DB8 fosse igual a três, e (2) a fila no disco DB12 fosse igual a dois.

O próximo passo foi determinar a quantidade necessária de discos para reduzir as filas durante os fechamentos mensais aos mesmos níveis observados nos demais períodos.

Demonstração da solução dada ao disco DB8

1) Para que o disco DB8 apresentasse nível de fila igual a 3 nos fechamentos mensais, era necessário que seu nível de utilização fosse igual a 75%, de acordo com o que foi exposto no item 2.10.5 Teoria das Filas.

2) Para relacionar o nível de utilização de um disco com a quantidade de acessos por segundo (TaxaChegada), foi utilizada a fórmula dada por Menascé&Almeida (2002, p. 81-82):

- U_d = TaxaChegada * (BuscaMédia + LatênciaMédia + TransferMédia)
- Foi observado que: taxa de transferência média do ambiente operacional = 0,5 ms, então:
- 0,75 = TaxaChegada * (8,2 + 4,17 + 0,5)
- TaxaChegada = 58 I/O por segundo

3) Para que a TaxaChegada fosse a mesma dos dias normais foi realizado o seguinte cálculo:

- TaxaChegada observada nos fechamentos mensais = 170 I/O por segundo
- A quantidade de discos necessários foi dada por 170/58 = 3

4) Então: 3 discos serão suficientes para manter a fila do disco DB8 durante os fechamentos mensais no nível de 3 processos em espera, como acontece nos demais períodos.

A Figura 3.6 ilustra o método utilizado na redução das filas em discos. Para solucionar o problema do tempo de resposta alto, foi aumentada a capacidade de atendimento dos discos. A implementação dessa solução foi através da aplicação do conceito de discos de continuação do sistema operacional utilizado por esta instalação, denominado MCP (*Master Control Program*). Com esta implementação foi possível fazer com que um disco lógico fosse formado por mais de um disco físico. Na prática o disco lógico DB8 passou a ser composto por 3 discos físicos: DB8 #1, DB8 #2 e DB8 #3. O conteúdo foi igualmente distribuído para garantir níveis de serviços semelhantes entre os três "atendentes". A distribuição de arquivos nessa nova configuração de discos foi realizada através do utilitário SYSTEM/COPY da *Unisys.*

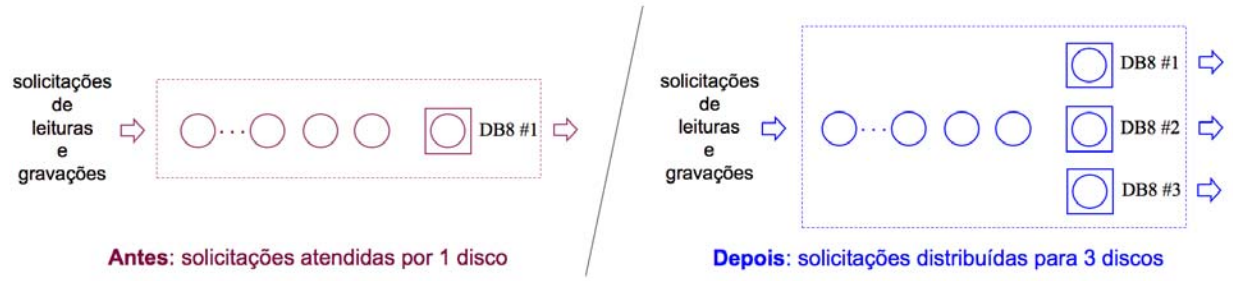

Figura 3.6 – Representação do método de redução de filas em discos

O método utilizado no disco DB8 foi aplicado a outros dois que atendiam o sistema e em seguida para todos os discos do ambiente operacional. O *software Excel* foi utilizado para auxiliar os cálculos.

A Figura 3.7 registra os resultados obtidos nos discos utilizados pelo sistema Fundos Diários. É possível observar que ao final do projeto, o nível de filas em discos durante os fechamentos mensais foram reduzidos para os mesmos valores dos dias normais anteriores ao início do projeto.

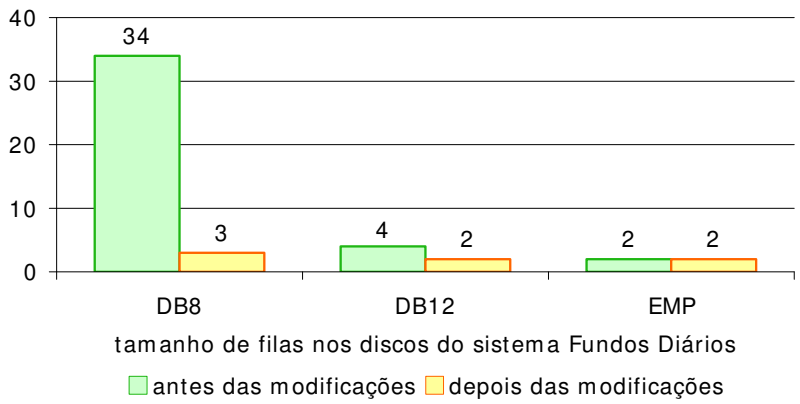

Figura 3.7 – Redução do tamanho das filas nos discos utilizados pelos Fundos Diários

A Figura 3.8 representa a diferença entre os tempos de processamento inicial e final, para o aplicativo objeto de estudo. No início havia uma variação entre 2 horas e 20 horas sem que houvesse alteração nas necessidades de recursos computacionais. O aplicativo sofria influência de outros, devido aos maiores níveis de atividade do equipamento, durante os fechamentos mensais dos demais sistemas. Com as implementações propostas por este trabalho, o tempo máximo para processamento foi reduzido de 20 horas para 2 horas.

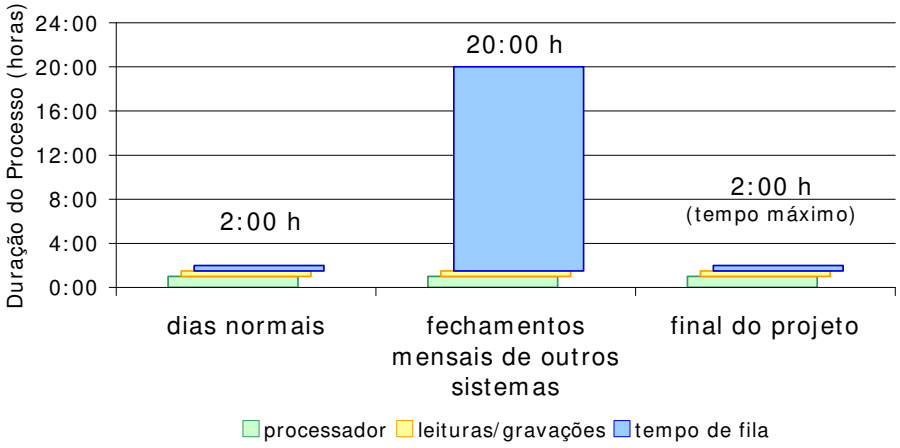

Figura 3.8 – Tempo de processamento para o Fundos Diários - antes e depois

A revisão sobre a Teoria das Filas foi importante para este trabalho pois permitiu encontrar fórmulas matemáticas para reduzir as esperas observadas nas operações de acesso a disco.

O próximo passo deste trabalho foi fundamentar a redução do nível de utilização dos discos através da aplicação do conceito industrial de *setup* nas operações de leituras e gravações.

3.8 Aplicação do conceito de *setup* em acesso a disco

O tempo de serviço de um disco magnético pode ser dividido em dois grupos, semelhantes aos conceitos aplicados às linhas de montagens industriais: tempo destinado à produção e tempo de *setup*. O tempo de *setup* é aquele tempo destinado à preparação do equipamento ou ambiente utilizado para a montagem de um determinado produto.

A contribuição do conceito de *setup* foi mensurada no *mainframe* de um dos ambientes de estudo onde este trabalho foi desenvolvido. Para isso, foi escolhido um dia onde houvesse a disponibilidade do equipamento por um período de 24 horas ininterruptas. A bateria de testes foi composta por um grupo de programas que leram e gravaram 1 milhão de registros. Foram utilizados arquivos convencionais de acesso seqüencial com registros de 512 caracteres, onde a quantidade de registros por bloco foi variado entre 1 e 128, através de parâmetros informados no programa.

Foram criados dois cenários para os testes: (1) Ambiente sem competição, onde cada programa foi executado isoladamente. As execuções foram iniciadas pelo programa que fazia o processamento de 128 registros por bloco, ao seu término era iniciado o próximo, que fazia o tratamento de 64 registros por bloco e assim até chegar ao programa que tratava 1 registro por bloco. (2) Ambiente com competição, onde as regras de execução foram similares ao ambiente sem competição, mas cada bateria de execução foi composta por 10 programas semelhantes.

A Tabela 3.2 relaciona o tempo observado no processo de criação de 1 milhão de registros em função do agrupamento de registros por bloco. As operações mecânicas do braço de leitura e gravação representam as operações de *setup* dos processos industriais, pois é uma preparação do equipamento antes de realizar a transferência dos dados. À medida que aumenta a quantidade de registros dentro de um bloco, diminui a quantidade de movimentações que o braço de leitura e gravação realiza para cumprir sua tarefa ao longo do processo. Como conseqüência, houve redução no tempo de execução de cada programa.

Tabela 3.2 – Quantidade de operações de *setup* X tempo observado de processo

Quantidade de registros por bloco	Quantidade de operações de setup	Duração em ambiente sem competição (hh:mm:ss)	Duração em ambiente competitivo (hh:mm:ss)
1	1.000.000	0:41:23	>12 h
2	500.000	0:22:20	6:14:21
4	250.000	0:12:25	2:33:00
8	125.000	0:07:07	1:48:30
16	62.500	0:04:20	0:51:29
32	31.250	0:02:47	0:26:23
64	15.625	0:02:26	0:12:11
128	7.813	0:02:10	0:07:17

Para a coleta dos dados utilizados em sua construção foi utilizado o *software* SYSTEM/SUMLOG do sistema operacional MCP.

A Figura 3.9 ilustra o impacto que o conceito de *setup* trouxe para o tempo de resposta de aplicativos *batch*. Os resultados foram colocados em duas curvas: (1) A curva inferior registra o tempo de processo para aplicativos que foram executados em ambiente sem competição, tinham o *mainframe* inteiramente disponível para a tarefa. A variação observada esteve entre 2 minutos e 41 minutos. (2) A curva superior registra o tempo de processo para os mesmos aplicativos executados em um ambiente de competição formado por 10 programas iguais executados ao mesmo tempo. O dado referente à execução do programa que tratou 1 registro por bloco não é mostrado, pois ultrapassou 12 horas de execução e precisou ser interrompido para que o tempo total do teste não ultrapassasse as 24 horas disponibilizadas para a tarefa.

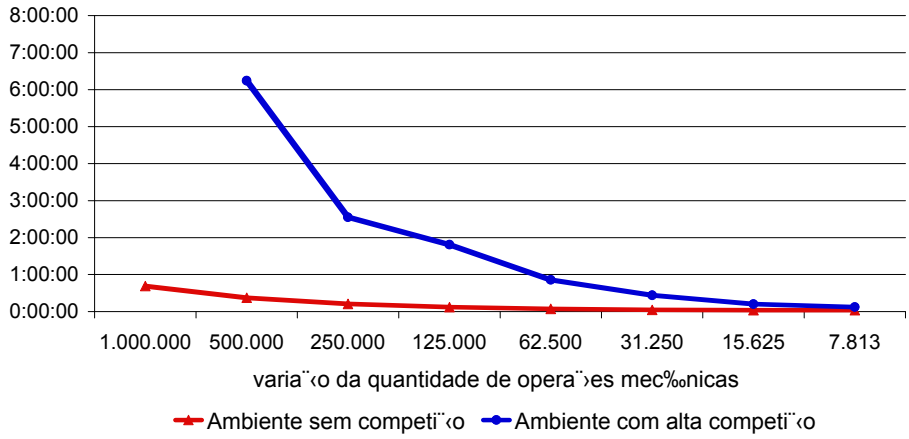

Figura 3.9 – Resultados da aplicação do conceito de *setup* em ambientes competitivos

Fazendo analogia com os conceitos de linha de produção, o tempo de fabricação pode ser reduzido de duas formas: (1) pela melhoria do processo de montagem, e (2) pela redução da quantidade ou duração das operações de *setup*. Assim, é possível analisar as seguintes alternativas para redução do tempo de processo de Tecnologia da Informação:

1. Troca de tecnologia: o tempo necessário para as operações de busca, latência e transferência diminuem a cada nova edição de disco magnético. Este fato está amparado pela Lei de Moore. [MOORE, 1965; PATTERSON&GRAY, 2003]

   Ponto contra: a proposta original deste trabalho é melhorar o tempo de resposta dos processos de T.I. do equipamento disponível, sem aquisições de novos recursos.

2. Disco de uso exclusivo: elimina o tempo de busca, pois sendo dedicado exclusivamente a um único aplicativo de acesso seqüencial, a cabeça de leitura e gravação, após atender a uma solicitação, permanece em repouso sobre a última trilha utilizada. Não havendo solicitações de outros aplicativos, a cabeça de leitura e gravações permanece em repouso no local do acesso anterior, ponto de partida da próxima tarefa.

   Ponto contra: com o crescimento constante da capacidade dos discos, os *mainframes* utilizam mais de um disco virtual dentro de um disco real, como conseqüência as cabeças de leitura e gravação de um disco físico são compartilhadas por mais de um disco lógico. Para essa alternativa funcionar é necessário que todos esses discos fiquem ociosos, o que é inviável operacional e financeiramente.

3. Redução da quantidade de *setup*: é a alternativa mais prática de ser implantada, pois é necessário e suficiente que seja aumentada a quantidade de registros lidos ou gravados em cada operação de processos batch.

A aplicação dos conceitos de *setup* utilizados pela indústria permitiu observar que: (1) o tempo total de acesso aos discos pode ser reduzido através da maximização de registros por bloco e (2) o aumento de registros por bloco diminui o nível de atividade do disco, diminui as filas de espera e diminui a variação do tempo de um processo batch em ambientes competitivos.

3.9 Descrição do ambiente de estudo 2

O segundo ambiente de estudo deste trabalho foram as transações *on-line* do Sistema Educação utilizado para apoiar o atendimento de alunos e professores e instalado em uma organização do setor público que realiza o processamento de dados para as secretarias e autarquias de um dos estados brasileiros. O equipamento utilizado foi um *mainframe* IBM modelo 2064, com capacidade de processamento de 575 MIPS (milhões de instruções por segundo) distribuídos em 2 processadores. O sistema operacional foi o z/OS. O gerenciador de banco de dados foi o *Adabas*. O gerenciador de transações foi o CICS (*Customer Information Control System*). O gerenciador de recursos foi o RMF (*Resource Measurement Facility*). Os discos utilizados foram da marca *Fujitsu* modelo *Spectris* 400, com capacidade para abrigar 6 discos lógicos em cada disco físico.

3.10 Mapeamento de discos lógicos

A Figura 3.10 esquematiza um disco físico contendo 6 discos lógicos. Os discos físicos, em computadores de grande porte, são similares aos utilizados pelos micro-computadores, conforme proposta de Patterson (1988). À medida que a capacidade de armazenamento aumenta, com taxas de crescimento demonstradas por Gray&Shenoy (2000), os discos físicos são capazes de abrigar cada vez mais discos lógicos. Os discos lógicos são aqueles referenciados pelo sistema operacional e pelos aplicativos hospedados em um *mainframe*.

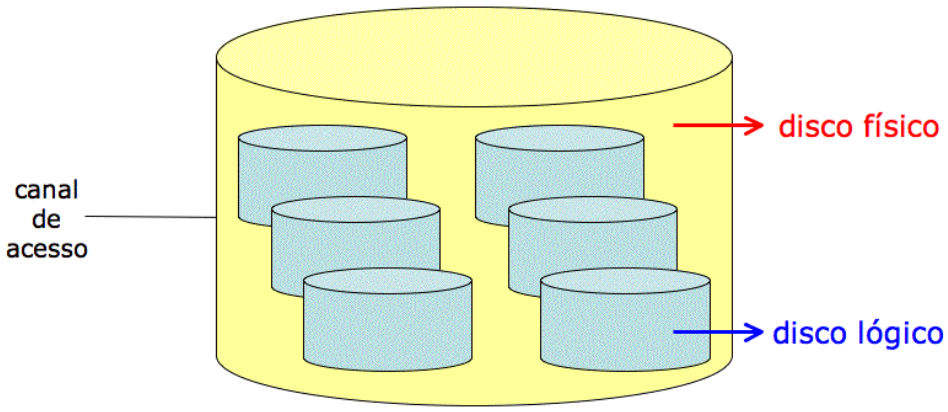

Figura 3.10 – Representação do arranjo de 1 disco físico contendo 6 discos lógicos

Além do mapeamento de processo proposto por este trabalho para o ambiente operacional 1, também foram mapeados os discos físicos e lógicos utilizados pelo ambiente operacional 2. Esse mapeamento auxiliou no balanceamento da carga de trabalho dos discos. O banco de dados do Sistema Educação utilizava 21 discos lógicos, que por sua vez ocupavam 6 discos físicos compartilhados por outros bancos de dados.

3.11 Aplicação do conceito de organização do estoque

A analogia do conceito de organização dos estoques nas fábricas permitiu que as tabelas de bancos de dados com maior quantidade de acesso fossem colocadas nas proximidades das bordas dos discos que as hospedavam.

A Figura 3.11 representa um prato de disco magnético utilizado pelo computador onde o resultado foi observado.

A Figura 3.12 representa a variação da taxa de transferência de dados em função da distância do disco lógico em relação à borda do disco físico. Para os dois primeiros discos lógicos próximos da borda, a taxa de transferência era de 25 MB/s, conforme a localização se aproximava do centro do disco, o valor caia para 18 MB/s. Isso acontecia pois à medida que se caminhava para o interior do disco o comprimento da trilha diminuía, diminuindo a quantidade de dados gravados por trilha.

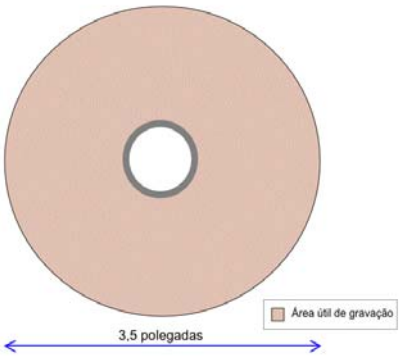

Figura 3.11 – Representação de um disco magnético

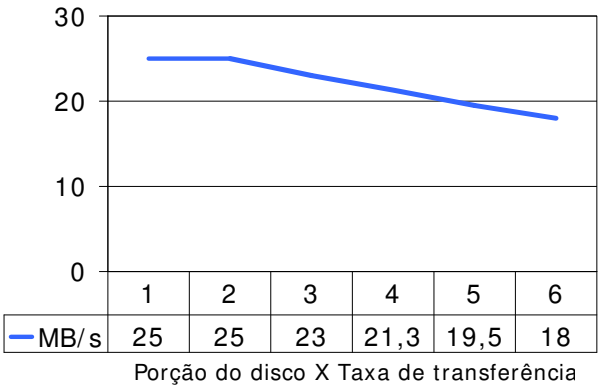

Figura 3.12 – Taxa de transferência do dado em função da proximidade da borda

A Figura 3.13 registra os tempos de resposta antes e depois da redistribuição dos 21 discos utilizados pelo Sistema Educação, instalado em um dos ambientes de estudo onde este trabalho

foi desenvolvido. É possível observar que os maiores tempos de respostas foram reduzidos enquanto que os menores foram aumentados. Essa troca permitiu que os discos com maior taxa de utilização tivessem condições de apresentar respostas mais rápidas, reduzindo o tempo de resposta das transações sem modificação de aplicativos e sem aquisição de equipamento.

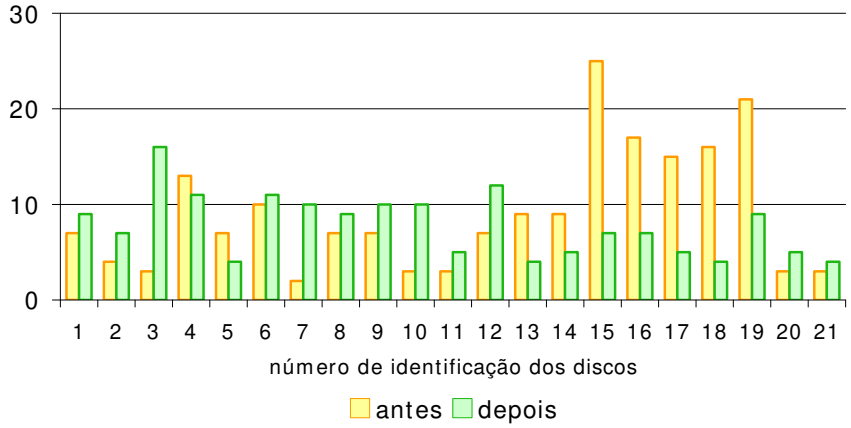

Figura 3.13 – Comparação do tempo de resposta em 21 discos redistribuídos

Capítulo 4

Resultados e Discussões

Não é bom proceder sem refletir...
(Provérbios 19.2)

4.1 Análise da problemática do ambiente operacional 1

No primeiro ambiente operacional analisado existiam aplicativos que apresentavam tempos de duração muito diferentes para processamento de volumes de dados similares.

O tempo de processamento de um dos aplicativos, que formavam o sistema Fundos Diários de Investimentos, variava entre 2 horas e 20 horas. Uma das características de sistemas de investimentos diários é possuir quantidades semelhantes de registros processados diariamente, que não justificava a variação no tempo de processamento. Diante dessa diferença, as equipes de desenvolvimento, suporte e produção, da organização bancária onde foi desenvolvido o primeiro ambiente de estudos, buscavam uma solução. Entretanto, a resposta não foi localizada utilizando apenas os conhecimentos tradicionais de análise de desempenho de sistemas.

A Figura 4.1 registra o histórico das situações antes e após a implantação da metodologia descrita neste trabalho. Em primeiro lugar estão os valores de 2 h e 20 h, que caracterizaram a variação e em seguida está o valor estabilizado de 2 h, resultado da implantação da metodologia.

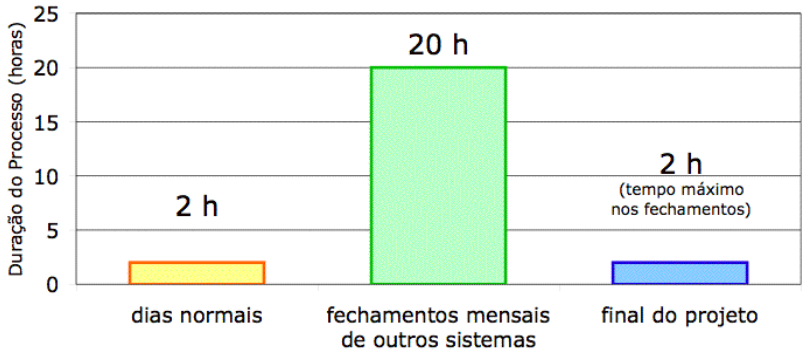

Figura 4.1 – Comparação entre os tempos de processos do Fundo Diário

4.2 Resultados da solução aplicada ao ambiente operacional 1

Resultados obtidos com redução de filas em discos

Com o aprendizado adquirido no sistema Fundos Diários, apresentado no capítulo Materiais e Métodos, foi possível obter ganhos nos demais sistemas da instalação.

A Figura 4.2 apresenta os resultados obtidos quando o método proposto foi aplicado aos outros sistemas instalados nesse *mainframe*. Esta etapa do trabalho focou a diminuição de filas em discos, através do aumento da capacidade de atendimento, assimilada durante a revisão dos conceitos de Teoria das Filas.

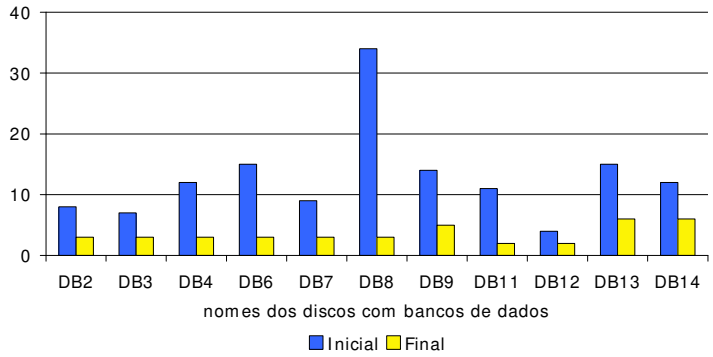

Figura 4.2 – Redução das filas nos discos que continham bancos de dados

Resultados obtidos com redução do tempo de *setup*

A Figura 4.3 registra os resultados obtidos com a implantação do conceito de *setup* nas operações de acesso a arquivos convencionais. Foram efetivados os resultados que estavam previstos teoricamente no item 2.11.5 Aplicação do conceito de *setup* em operações de leituras e gravações.

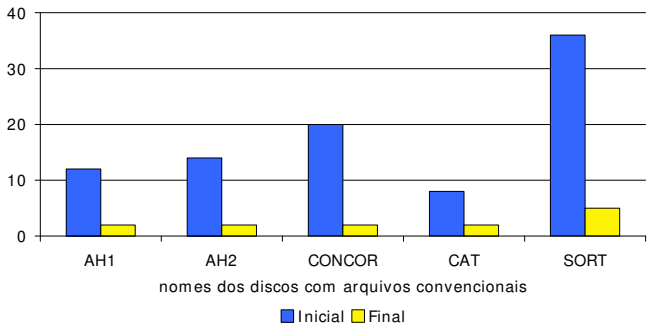

Figura 4.3 – Redução das filas nos discos que continham arquivos convencionais

Redução no tempo de processamento

A Figura 4.4 registra as redução dos tempos médios de processamentos dos aplicativos no ambiente de estudo 1. Elas foram reflexo da diminuição dos tamanhos das filas em discos onde residiam seus arquivos convencionais e seus bancos de dados. As implementações, além de reduzirem o tempo de processamento e tornarem possível a estabilização dos tempos dos processos executados durante a janela noturna, também aumentaram a garantia da abertura das agências pelas manhãs com a totalidade das informações atualizadas.

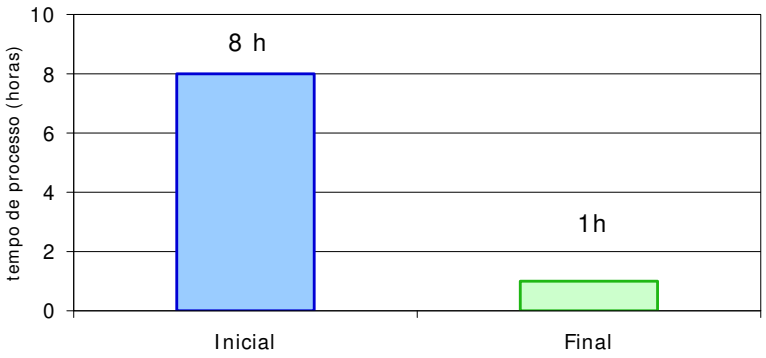

Figura 4.4 – Redução dos tempos de processamento dos aplicativos

Redução no uso de processador

A Figura 4.5 registra a redução na taxa de uso de processador. No início dos trabalhos, o equipamento estava operando em 80% da capacidade, no final do projeto, a média de utilização do processador baixou para 56%. Esse índice permitiu que os tempos de processos fossem estabilizados.

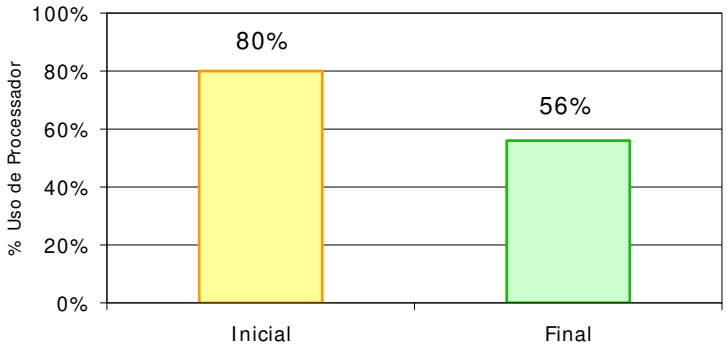

Figura 4.5 – Redução do uso de processador no final do projeto

Comprovação da Lei de Amdahl para sistemas equilibrados

Ao término dos trabalhos foi possível constatar a validade da Lei de Amdahl para os dias atuais. Essa lei, conforme mencionado no item 2.4.3 Lei dos Sistemas Equilibrados, foi desenvolvida no ano de 1967 para direcionar o balanceamento de recursos computacionais.

A Figura 4.6 representa os valores definidos pela Lei de Amdhal para sistemas equilibrados: deve existir uma relação de 1 MB/s para 8 MIPS. [AMDAHL, 1967; GRAY & SHENOY, 2000; GRAY ET AL, 2006]

A Figura 4.7 representa os valores medidos ao final do projeto. É possível observar a proporção de 1 para 8 entre quantidade de dados lidos ou gravados (medidos em *megabytes* por segundo) e o consumo de processador (medido em milhões de instruções por segundo). A existência dessa relação ao final do trabalho permite concluir que o método proposto levou o ambiente ao equilíbrio.

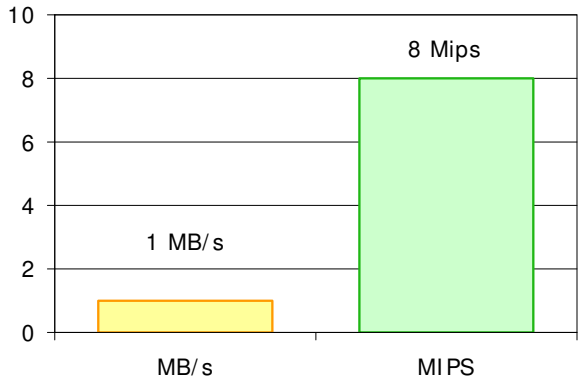

Figura 4.6 – Lei de Amdahl prevista

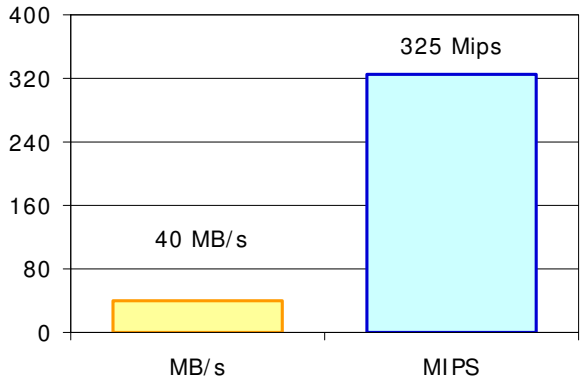

Figura 4.7 – Lei de Amdahl observada

Comparação com o método tradicional

O método tradicional de melhora de desempenho de aplicativos consiste em analisar a lógica do programa e localizar sentenças que possam ser substituídas por outra de melhor desempenho. É verdade que existem comandos de programação que são mais velozes que outros e por isso, é possível localizar alguns deles e substituí-los por outros mais rápidos. Entretanto, esse método exige grande quantidade de horas de analistas e programadores para obter reduções na razão de nanosegundos, que é a velocidade dos processadores. Como as expectativas para redução do tempo de processos apoiados por computadores de grande porte estão em torno de

minutos ou mesmo em torno de horas. Hoje, essas reduções não são satisfatórias, pois já não trazem os resultados do passado, quando as velocidades de discos e processadores eram mais próximas.

Além disso, o consumo de processador para efetuar os testes necessários e os procedimentos de homologação de cada programa modificado, pode ser superior ao que será economizado e os riscos introduzidos pelas modificações de lógicas também precisam ser ponderados.

4.3 Análise da problemática do ambiente operacional 2

O Sistema Educação tinha a responsabilidade de matricular e transferir alunos e professores, atribuir aulas aos professores e relacionar as atividades e seus executores. Sua capacidade de atendimento, no início dos trabalhos, era de 1.000 usuários simultâneos em todo o estado. O tempo para atender cada aluno ou professor chegou a atingir picos de 20 minutos, sem considerar o tempo que permaneceram na fila de espera.

4.4 Resultados da solução aplicada ao ambiente operacional 2

Este ambiente operacional estava composto por modelos de discos que apresentavam desempenhos diferentes em função da proximidade do dado em relação à borda do prato magnético.

A solução adotada foi sub-dividida em 3 etapas: (1) transferir as tabelas de bancos de dados com maior número de solicitações de acessos para as bordas, (2) transferir as tabelas com menor utilização para o centro do prato magnético, (3) distribuir a carga de trabalho entre os discos para reduzir a concorrência nos canais de acesso.

Os valores envolvidos nessas operações tinham pouca variação, estavam entre 25 MB/s a 18 MB/s, conforme foi registrado no item 3.11 Aplicação do conceito de organização do estoque, entretanto o efeito multiplicador de 1000 usuários simultâneos trouxe resultados significativos.

Resultados obtidos com o conceito de organização de estoques

A Figura 4.8 apresenta os resultados obtidos com a aplicação do conceito de organização de estoques utilizados pela indústria. Através da figura é possível comparar os tempos de resposta anteriores e posteriores às implementações, nas 9 principais transações do Sistema Educação.

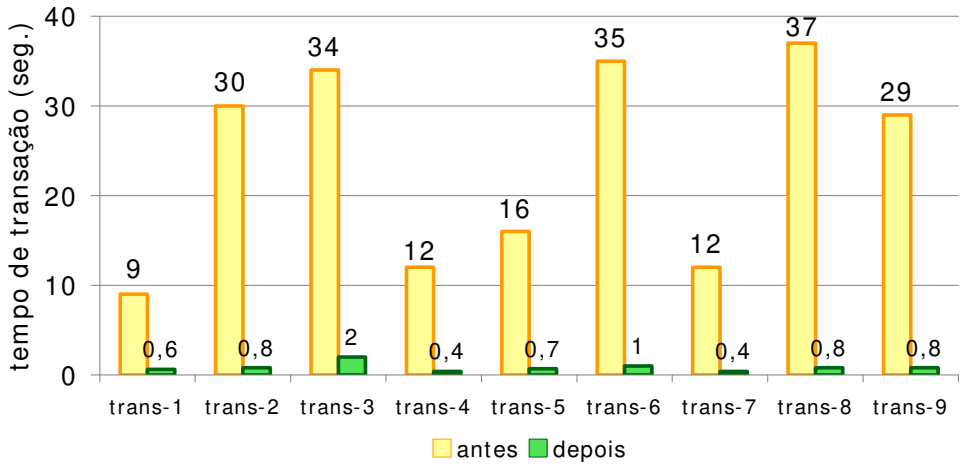

Figura 4.8 – Comparação entre os tempos de transações - antes e depois

Resultados observados pelos usuários

A Figura 4.9 registra a diferença entre os tempos de atendimentos aos clientes internos e externos. Antes do início do projeto, cada cliente, depois de passar pela fila de espera, precisava de 20 minutos para que todas as transações necessárias ao seu atendimento fossem processadas. Ao final do projeto, esse tempo foi reduzido para 2 minutos.

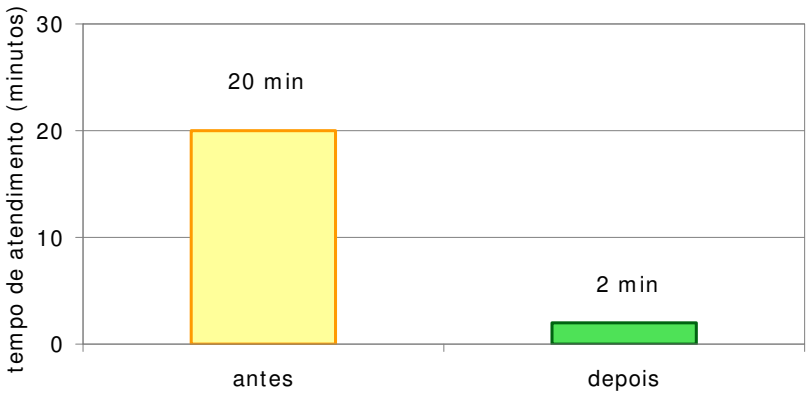

Figura 4.9 – Comparação entre os tempos de atendimentos - antes e depois

A Figura 4.10 registra o aumento na quantidade de usuários atendidos. Depois da implantação do projeto, o número de usuários simultâneos no sistema passou de 1000 para 1200 pessoas.

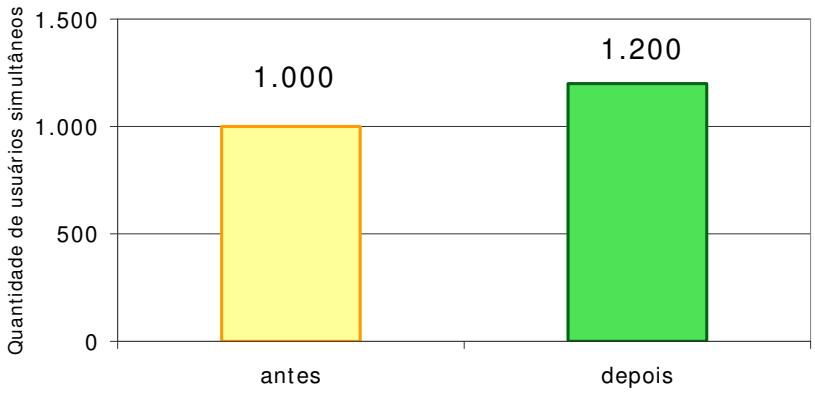

Figura 4.10 – Comparação entre a quantidade de usuários simultâneos - antes e depois

Aumento no nível de serviço com mesmo uso de processador

Através da Figura 4.11 é possível observar que o nível de utilização de processador não foi alterado ao longo do projeto. Foi possível oferecer maiores níveis de serviço sem sobrecarregar o processador disponível.

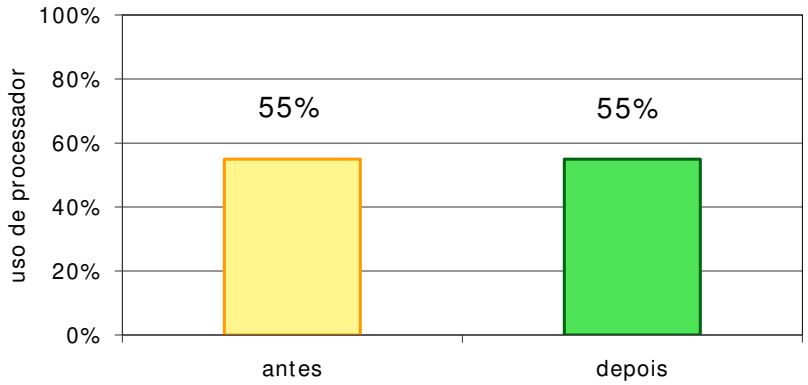

Figura 4.11 – Comparação entre o uso de processador - antes e depois

4.5 Considerações sobre modificações de aplicativos

A produção ideal acontece quando se solicita um recurso e ele está livre, sem fila. Porém o custo para manter um ambiente computacional ideal é economicamente inviável.

Visando os custos finais de processos de Tecnologia da Informação, este trabalho procurou reduzir o tamanho das filas em disco através da melhor utilização dos recursos disponíveis, sem propor aquisição de novos *hardwares* como processador, memória e discos.

A viabilidade econômica dos processos foi a maior preocupação ao se sugerir uma fila pequena. Pois, quando há espera por um componente lento, como os discos, recursos mais nobres, como memória e processador, ficam ociosos, sub-utilizando investimentos e encarecendo o produto final.

Em termos de Tecnologia da Informação, foi observado que existem resultados mais consistentes quando as melhorias são iniciadas pelas operações de leituras e gravações ao invés de ações que investem na redução de consumo de processador. Um processador tem suas operações realizadas em nanosegundos (10^{-9}), as operações de leituras e gravações acontecem em milisegundos (10^{-3}), o que dá a seguinte relação:

$$\frac{10^{-3}}{10^{-9}} = 1.000.000$$

Portanto, melhorar a forma de uso dos arquivos trouxe resultados maiores que aqueles trazidos pelo método tradicional de modificações de aplicativos.

Quanto às filas em discos o ideal é que elas tenham tamanho igual a zero, entretanto isso implica na aquisição de uma quantidade de discos tendendo a infinito, por isso este trabalho propôs duas alternativas: (1) reduzir as filas através da utilização de discos ociosos ou menos utilizados e (2) reduzir a utilização dos recursos de discos a partir do conceito de *setup* observado nos processos industriais de fabricação.

4.6 Considerações sobre o termo "periférico"

As operações de leituras e gravações sempre foram as órfãs da arquitetura de computadores. Historicamente negligenciadas por entusiastas de processadores, o preconceito contra essas operações é institucionalizado na medida de desempenho mais comum, que é o tempo de processador. O desempenho das operações de leituras e gravações de um computador não pode ser medido pelo tempo do processador que, por definição, as ignora. A condição de cidadão de segunda classe dessas operações fica mais aparente no título de "periféricos" aplicados aos dispositivos que realizam as operações de leituras e gravações. Embora o tempo de processador seja interessante, o tempo de resposta é melhor como indicador de desempenho. Ele compreende o tempo entre o momento que o usuário digita um comando e o momento que os resultados aparecem. Algumas pessoas sugerem que esse preconceito tem fundamento. A velocidade das operações de leituras e gravações não importa, argumentam, pois sempre existe um processo que pode ser executado enquanto outro espera por um periférico. Certamente, se os usuários não se importassem com o tempo de resposta, o *software* interativo nunca teria sido criado, e hoje não existiria nenhuma estação de trabalho ou computador pessoal. [HENNESSY&PATTERSON, 2003, p. 678]

4.7 Considerações sobre Custo de Propriedade

Ainda que o objetivo deste trabalho não fosse formalizar custos, a revisão dos conceitos de Custos de Propriedade permitiu observar que eles estão além do valores destinados para aquisição de tecnologia, desenvolvimento de sistemas e treinamentos em *softwares*.

A Figura 4.12 ilustra uma curva de custo total em função do nível de serviço do sistema. É possível observar que, enquanto os custos operacionais de oferecer o serviço aumentam com o aumento do nível de serviço, os custos devido aos atrasos sofridos pelos usuários diminuem. [ARENALES ET AL, 2007, p. 434-435]

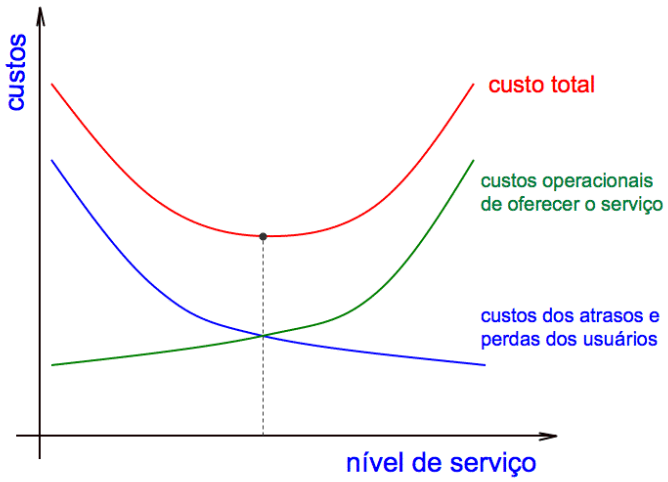

Figura 4.12 – Curva de custo total em função do nível de serviço

Durante o desenvolvimento deste trabalho, foram identificados os seguintes fatores como agregadores de custos em sistemas instalados em um computadores de grande porte:

1) O custo da fila de espera. Foi possível refletir, através dos estudos de Arenales (2007), que a economia obtida com a instalação de um equipamento com capacidade inferior às necessidades de seus usuários é anulada ou superada pelos custos de espera pelo atendimento.

2) Custo de manutenções em aplicativos visando a melhoria do desempenho a partir do aprimoramento da lógica de programação.

3) Custo de administração de riscos inseridos no negócio por falha nos procedimentos de testes ou homologações de sistemas modificados pelos processos de manutenções.

Existem outras contribuições que os conceitos de Custo de Propriedade podem agregar aos processos de Tecnologia da Informação, e por isso, o aprofundamento destas pesquisas está mencionado no capítulo Sugestões para Trabalhos Futuros.

Capítulo 5

Conclusões e Sugestões para Trabalhos Futuros

5.1 Conclusões

Mesmo com o uso de tecnologia moderna, como memória *cache* e discos RAID, o tempo de resposta nos ambientes de estudo estavam, antes do início deste trabalho, acima das expectativas dos usuários.

Observando a similaridade entre os processos industriais e os da Tecnologia da Informação, foi possível utilizar conceitos de administração da produção para desenvolver uma metodologia alternativa que reduziu o tempo de resposta em computadores de grande porte com resultados superiores aos obtidos pelo método tradicional de modificação de código de aplicativos. Por causa da diferença de velocidade entre discos e processadores, o ambiente operacional causa influência no desempenho de um aplicativo em maior grau que a qualidade do código de programação utilizado em sua lógica. Isso ocorre devido aos milhares de outros aplicativos que disputam recursos em no interior de um computador de grande porte e causam filas de espera durante o processo.

Através do compartilhamento de conceitos observados em linha de produção, foi possível organizar o raciocínio, localizar os pontos críticos, priorizar as atividades e conduzir a implantação de melhorias sem impacto no ambiente operacional.

Os resultados foram obtidos em duas arquiteturas de sistemas operacionais distintos: (1) No MCP (*Master Control Program*) da Unisys, projetado sob o conceito de ortogonalidade vetorial ou independência de vetores no espaço euclidiano e (2) no sistema operacional z/OS da IBM, projetado sob a arquitetura de von Neumann. A metodologia proposta alcançou os seguintes resultados:

Ambiente operacional 1 – MCP Unisys – processamento noturno de sistemas bancários

- Estabilidade no tempo de processamento de um aplicativo que apresentava duração variando entre 2h e 20h. O tempo final ficou estabelecido em 2 horas, independente da carga de trabalho do equipamento. O aprendizado foi levado a outros sistemas e os resultados foram reproduzidos.
- Redução de 8 h para 1 h no tempo de processamento de aplicativos do período noturno.
- Aumento da margem de segurança para eventuais reprocessamentos no período noturno.
- Garantia da aberturas das agências no horário estabelecido.
- Validação da Lei de Amdahl para sistemas atuais.
- Redução no uso de processadores na ordem de 24%.
- Sobrevida do equipamento por 2 anos sem aquisição de maior capacidade de processador.

Ambiente operacional 2 – z/OS IBM – Sistema Educação

- Redução do tempo de transação na ordem de até 40 seg. para próximo de 1 seg.
- Redução no tempo de atendimento aos usuários de 20 minutos para 2 minutos.
- Aumento da capacidade de atendimento de 1000 para 1200 pessoas simultaneamente, sem aumento de uso de processador.

Considerações sobre outros ambientes operacionais

A metodologia proposta neste trabalho foi reproduzida em outros tipos de ambientes operacionais, além do bancário e educacional. Com algumas adaptações, foi aplicada em ambientes Windows e Linux e obteve resultados positivos.

5.2 Sugestões para Trabalhos Futuros

... assim também andemos nós em novidade de vida.
(Romanos 6.4)

A partir das pesquisas realizadas e do material coletado durante o desenvolvimento deste trabalho foram observadas as seguintes propostas de trabalhos futuros:

1. Desenvolver um método capaz de: (1) conceituar a diferença entre melhoria e otimização, (2) estabelecer o ponto de máxima eficiência de um *mainframe*, (3) localizar as causas que o afasta desse ponto e (4) propor alternativas para a aproximar o desempenho do *mainframe* de seu ponto de máxima eficiência.

2. Desenvolver um método capaz de estabelecer, com maior precisão, a relação entre redução de filas em discos e redução do uso de processador. Esse trabalho permitirá mensurar a quantidade de processador utilizado para administrar filas em disco e conseqüentemente, não utilizada para atender o usuário.

3. Desenvolver um método capaz de validar a robustez dos modelos matemáticos utilizados nos *softwares* e serviços de planejamento de capacidade (*capacity planning*) de *mainframes*. Durante o desenvolvimento deste trabalho, foi observado que: (1) os desperdícios de processadores para administrar filas em discos são projetados para as próximas configurações e aquisições de equipamentos e (2) as simulações utilizadas por esses *softwares* projetam seus resultados considerando que o desempenho do equipamento está em seu ponto ótimo.

Referências Bibliográficas

Anderson, Dave. You don't know jack about disks. ACM Queue (June) 2003, v.1, n.4, pp20-30. Disponível em: www.acmqueue.org /modules.php?name =content&pa=showpage&pid=46. Acesso em 09/Maio/2006.

Arenales, Marcos N., et al. Pesquisa operacional para cursos de engenharia. Rio de Janeiro: Campus, 2007, 523p.

Botelho, Adriano. *Do fordismo à produção flexível: a produção do espaço num contexto de mudança das estratégias de acumulação do capital.* São Paulo: Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo, 2000, 148 p. Dissertação (Mestrado).

Corbató, Fernando J.; Daggett, Marjorie M.; Daley, Robert C. An Experimental Time-Sharing System. Computation Center MIT, Cambridge, MA. 1962. [Scanned and transcribed by F. J. Corbató from the original SJCC Paper of May 3, 1962]. Disponível em: www.eecs.harvard.edu/cs261/papers/corbato62.pdf. Acesso em 09/Jun/06.

Creasy, R. J. The Origin of the VM/370 Time-Sharing System. IBM J. Res. Develop. Vol. 25 no. 5. pp 483-490. September 1981. Disponível em: www.cis.upenn.edu/~cis700-6/04f/papers/creasy-vm-370.pdf. Acesso em 09/Jun/06.

Denning, Peter J. The Locality Principle: locality of reference is a fundamental principle of computing with many applications. Communications of the ACM. Vol. 48 no. 7. pp 19-24. July 2005. Disponível em: portal.acm.org/citation.cfm?id=1070856&coll=&dl= ACM&CFID= 15151515&CFTOKEN=6184618#references. Acesso em 24/Out/06.

Dictionary.com Unabridged (v 1.1). Random House, Inc. Disponível em: http://dictionary.reference.com.

Fainguelernt, Estela K.; Nunes, Kátia K. R. Fazendo arte com a matemática. Porto Alegre: Artmed, 2006, 126p.

Fitzsimmons, James A.; Fitzsimmons, Mona J. Service Management: operations, strategy, information technology. 5th ed. New York: McGrawn-Hill, 2005, 638p.

French, Simon. Sequencing and Scheduling: an introduction to the mathematics of the Job-Shop. New York: John Wiley & Sons Inc., 1982, 245p.

Godinho, Rogério; Saito Ana C. Mainframe sobrevive e cresce no País. Gazeta Mercantil, C-1, 06/Jun/2007.

Goldratt, Eliyahu M.; Cox, James. A Meta: um processo de melhoria contínua. 2ª. edição. Tradução de Thomas Corbett Neto. São Paulo: Nobel, 2003, 366p.

Gray, Jim; Putzolu, Franco. The 5 Minute Rule for Trading Memory for Disc Accesses and The 5 Byte Rule for Trading Memory for CPU Time. Technical Report TR86.1 Tandem Computers, Cupertino, CA. May, 1985. Disponível em: research.microsoft.com /~gray/papers/TandemTR86.1_FiveMinuteRule.doc. Acesso em 09/Maio/2006.

Gray, Jim; Graefe, Goetz. The Five-Minute Rule Ten Years Later, and Other Computer Storage Rules of Thumb. SIGMOD Record, vol. 26, no. 4, 1997, pp 63-68. Disponível em: research.microsoft.com /~gray/5_min_rule_SIGMOD.doc. Acesso em 09/Maio/2006.

Gray, Jim; Shenoy, Prashant. Rules of Thumb in Data Engineering. 16th International Conference on Data Engineering, San Diego, 1/Mar/2000, pp 3-12. Disponível em: ftp://ftp.research.microsoft.com/pub/tr/tr-99-100.pdf. Acesso em 09/Maio/2006.

Gray, Jim. The Revolution in Database Architecture. Paris, France. Association for Computing Machinery, Inc. 2004. Disponível em research.microsoft.com/research/pubs/view.aspx?tr_id=735. Acesso em 22/Jun/06.

Graziadio, Thaise. *Estudo comparativo entre os fornecedores de componentes automotivos de plantas convencionais e modulares.* São Paulo: Escola Politécnica da Universidade de São Paulo, 2004, 185 p. Tese (Doutorado).

Hennessy, John L.; Patterson, David A. Computer architecture: a quantitative approach. 3rd ed. San Francisco: Morgan Kaufmann, 2003, 1093p.

Hennessy, John L.; Patterson, David A. Computer Architecture: a quantitative approach. 4th ed. San Francisco: Morgan Kaufmann, 2007, 688p.

Hillier, Frederick S.; Lieberman, Gerald J. Introduction to operations research. San Francisco: Holden-Day, Inc., 1967, 639p.

IBM Redbook Effective zSeries Performance Monitoring Using Resource Measurement Facility. Manual técnico IBM SG24-6645-00. 2005, 358p.

James, Andrea. The Amazing Grace Hopper. Oct 30, 2001. Disponível em: www.computinghistorymuseum.org/teaching/papers/biography/james.pdf. Acesso em 22/Jun/06.

Knuth, Donald E. The Art of Computer Programming: sorting and searching. 3rd ed. Reading: Addison-Wesley, 1998, 650p.

Menascé, Daniel A.; Almeida, Virgílio A. F. Capacity Planning for Web Services: Metrics, Models and Methods. Upper Saddle River: Prentice Hall PTR, 2002, 572p.

Menascé, Daniel A.; Almeida, Virgílio A. F. Planejamento de Capacidade para Serviços na WEB: métricas, modelos e métodos. Tradução: Daniel Vieira e Virgílio A. F. Almeida. Rio de Janeiro: Campus, 2003, 445p.

Moore, Gordon. Cramming More Components Onto Integrated Circuits. Electronics, April 19, 1965. Disponível em: ftp://download.intel.com/museum/ Moores_Law/Articles-Press_Releases /Gordon_Moore_1965_Article.pdf. Acesso em 09/Maio/2006.

Mueller, John P.; Chaudhry, Irfan. Microsoft Windows 2000 Performance Tuning Technical Reference. Redmond: Microsoft Press, 2000, 540p.

Nunes, Gustavo A. *Desenvolvimento de um método de melhoria do processo logístico de uma*

empresa prestadora de serviços de distribuição de energia elétrica pela identificação de gargalos e avaliação dos custos das atividades desenvolvidas. Porto Alegre: UFRGS, 2004, 120p. Dissertação (Mestrado). Disponível em: www.producao.ufrs.br/arquivos/publicacoes/gustavo_nunes.pdf. Acesso em 09/Maio/2006.

O'Connor, J. J.; Robertson E. F. Biografia de Grace Hooper. Jul/1999. Disponível em: http://www-history.mcs.st-andrews.ac.uk/Biographies/ Hopper.html. Acesso em 09/Maio/2006.

O'Connor, J. J.; Robertson E. F. Biografia de Donald Ervin Knuth. Abr/2002. Disponível em: http://www-history.mcs.st-andrews.ac.uk/ history/Mathematicians/Knuth.html. Acesso em 09/Maio/2006.

Patterson, David A.; Gibson, Garth; Katz, Randy H. A Case for Redundant Arrays of Inexpensive Disks (RAID). Proceedings of the 1988 ACM SIGMOD Conference on the Management of Data, pp. 109-116.

Patterson, David A.; Chen, Peter M. Storage Performance: Metrics and Benchmarks. Proceedings of the IEEE Volume 81, Issue 8, Aug 1993, pp 1151-1165.

Patterson, David A.; Gray, Jim. A Conversation with Jim Gray. Entrevista realizada por David Patterson em Jun/03. Disponível em: www.acmqueue.org/modules.php?name=Content&pa=showpage& pid=43. Alternativa em: research.microsoft.com /~gray/papers/QueueAConversationWithJimGray.pdf. Acesso em 04/Abr/06.

Patterson, David A; Keeton, Kimberly K. Hardware Technologies Trends and Database Opportunities. [2000]. Disponível em: cs.berkeley.edu/~patterson/talks. Acesso em 29/09/2006.

Porter, James. How computer storage became a modern business. Palestra apresentada no Computer History Museum. Mountain View, CA em 09/Mar/2005. Disponível em: archive.computerhistory.org /lectures//how_computer_storage_became_ a_modern_business.wmv. Acesso em 09/Maio/2006.

Prado, Darci Santos do. Teoria das filas e da simulação. Belo Horizonte: Desenvolvimento Gerencial, 1999, 122p.

Sacomano, J. B; Melo, J. G. *Estudo comparativo do seis sigma e do pensamento enxuto.* In: XI SIMPEP, Bauru. 2004. Disponível em: www.feb.unesp.br. Acesso em 30/Jan/07.

Sebesta, Robert W. Concepts of Programming Languages. 8th ed. Boston: Addison-Wesley, 2007, 696p.

Severino, Antonio J. Metodologia do trabalho científico. 22ª ed. São Paulo: Cortez, 2002, 334p.

Slack, Nigel; Chambers, Stuart; Johnston, Robert. Administração da Produção. Revisão técnica de Henrique Corrêa. 2ª ed. São Paulo: Editora Atlas, 2002, 754p.

SPC Benchmark executive summary IBM system storage DS4800 disk storage system. 10/Abr/2007. Disponível em: www.storageperformance.org /results/a00050_IBM-DS4800_ SPC1_executive-summary.pdf. Acesso em 08/Jun/2007.

Stallings, William. Operating systems: internals and design principles. 5th ed. Upper Saddle

River: Prentice-Hall, 2005, 818p.

Stallings, William. Computer organization and architecture: designing for performance. 7th ed. Upper Saddle River: Prentice Hall, 2006, 778p.

Womack, James; Jones, Daniel. A mentalidade enxuta nas empresas: elimine o desperdício e crie riqueza. Tradução: Ana Beatriz Rodrigues, Priscila Martins Celeste. Rio de Janeiro: Campus, 2004, 458p.