

Resumo de Artigos Seleccionados

Celina d' Ávila Samogin - RA 820157

6 de agosto de 2011

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distribuição de Dados
Referência Bibliográfica:	Byers, J. W.; Luby, M.; Mitzenmacher, M. ; Rege, A. A digital fountain approach to reliable distribution of bulk data. In: <i>Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM'98)</i> .
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.628
Número de Páginas:	12
Citações (CiteSeer):	285
Número de Referências:	27
Visão geral do assunto do artigo:	Esse artigo descreve um protocolo chamado Digital Fountain, projetado para aplicações que tem requisitos de confiabilidade na distribuição de dados e de programas. Esses requisitos foram descritos: confiabilidade, eficiência, suporte às requisições dos clientes (iniciar, interromper, continuar), tolerância (perda de dados e diferentes taxas de transmissão). São brevemente apresentadas várias soluções já propostas para garantia destes requisitos. A DF utiliza um princípio básico de <i>erasure codes</i> : os dados originais da fonte e seus dados redundantes são transmitidos para o destino e os dados redundantes podem ser usados para recuperar dados perdidos. Os autores descrevem o que eles chamam de requisitos importantes no projeto deste protocolo para obtenção de tráfego mínimo na rede e de escalabilidade. Os meios físicos considerados incluem IP <i>multicast</i> sob internet, transmissão por satélite, rede sem fio e transmissão cabeada.
Conclusões, Contribuições e Resultados Obtidos:	A contribuição desse trabalho é uma comparação entre Tornado codes e Reed-Solomon codes. Foi feito um experimento de transmissão de um vídeo de 2MB, em uma rede entre UBerkeley, CMU e UCornell entre máquinas SUN Solaris.
Comentários:	Não foram consideradas aplicações de tempo real.
Expressões-chave:	<i>Erasure Codes, Tornado Codes, Reed-Solomon Codes</i>
Observações:	O artigo foi sugerido nas notas de aula de Peer-to-Peer and Grid Computing http://www.cs.umd.edu/class/spring2007/cmssc818s/Lectures/lectures.htm acessado em 19/03/2010.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Sistema de Armazenamento Distribuído
Referência Bibliográfica:	J. Kubiawicz; D. Bindel; Y. Chen; S. Czerwinski; P. Eaton; D. Geels; R. Gummadi; S. Rhea; H. Weatherspoon; W. Weimer; C. Wells; B. Zhao. OceanStore: an architecture for global-scale persistent storage. In: <i>Proceedings of the 9th international conference on Architectural support for programming languages and operating systems (ASPLOS '00)</i> .
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.1044
Número de Páginas:	12
Citações (CiteSeer):	749
Número de Referências:	52
Visão geral do assunto do artigo:	Os autores falam dos requisitos da computação obíqua em infraestrutura persistente: conectividade, segurança e durabilidade de dados. São comentados os objetivos do OceanStore: possibilidade de ser projetado a partir de uma infraestrutura não-confiável e suportar dados que fluem livremente. O objetivo do OceanStore é propor uma arquitetura para Internet-wide archival storage utility baseado em uma federação de servidores não confiáveis. Os autores comentam o atual estado do protótipo: objetos e suas versões, objetos e suas formas (ativa e arquivada), réplicas e servidores e de que forma um objeto é encontrado (algoritmo probabilístico e algoritmo determinístico). Algumas aplicações do OceanStore são comentadas: serviço de e-mail, bibliotecas digitais, repositórios para dados científicos e agregação e difusão de dados de sensores. A arquitetura foi apresentada: identificação de objetos, controle de acesso (restrições de escrita e de leitura), mecanismos de localização de dados, modelo de atualização de dados, mecanismos de arquivamento (utiliza Tornado codes e Read-Solomon codes) e APIs para sistemas de arquivos Unix, para base de dados transacionais e <i>gateway</i> para internet. O princípio básico de <i>Erasure codes</i> utilizado pelo OceanStore é o tratamento dos dados de entrada como uma série de fragmentos (n), sua transformação desses fragmentos em um número maior ainda ($2n$ ou $4n$) e a propriedade essencial do código resultante que é que quaisquer n fragmentos codificados são suficientes para construir os dados originais. Os fragmentos são gerados em paralelo pela camada interna de servidores durante o processo de commit.
Conclusões, Contribuições e Resultados Obtidos:	A contribuição desse trabalho é uma publicação que propõe uma arquitetura para armazenamento de dados que usa <i>erasure codes</i> . Os autores calculam a confiabilidade em um dado instante através de probabilidade e através da variação do número de fragmentos, concluem que a fragmentação dos dados aumenta a confiabilidade.
Comentários:	Os autores supõem que existam apenas falhas não correlacionadas entre as máquinas. O código pode ser baixado em http://oceanstore.cs.berkeley.edu/
Expressões-chave:	<i>Ubiquitous Computing, Data Durability, Distributed Storage System, Replication, Erasure Code</i>
Observações:	O texto do artigo foi acessado em 03/04/2010. Outros artigos estão em http://oceanstore.cs.berkeley.edu/publications/index.html

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Disponibilidade de dados
Referência Bibliográfica:	Weatherspoon, H.; Kubiawicz, J. D. Erasure Coding vs. Replication: A Quantitative Comparison. In: <i>Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'01)</i> London, UK. p. 328–338, March 2002.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.5545
Número de Páginas:	10
Citações (CiteSeer):	115
Número de Referências:	14
Visão geral do assunto do artigo:	Esse artigo faz uma quantificação estatística das vantagens de <i>erasure codes</i> , comparando características de sistemas que utilizam esta técnica e também utilizam replicação de dados, com o objetivo de obter alta disponibilidade e durabilidade de dados diante de falhas de componentes destes sistemas. O artigo introduz uma definição de <i>erasure codes</i> em termos de codificação e decodificação dos fragmentos de objetos. Os autores assumem que os sistemas estudados são constituídos por uma coleção de discos distribuídos que podem falhar e que são imediatamente substituídos por outros discos novos e sem dados. Em seguida, são descritas detalhadamente três comparações utilizando tempo médio entre falhas, <i>overhead</i> de armazenamento e <i>repair epoch</i> (tempo de verificação do bloco). O artigo ainda apresenta brevemente uma arquitetura híbrida (arquivador central, fragmentos e réplicas) e conclui falando de questões abertas para pesquisa.
Conclusões, Contribuições e Resultados Obtidos:	A contribuição desse trabalho é a publicação de um trabalho que quantifica o ganho de disponibilidade de dados obtido com <i>erasure codes</i> , que <i>erasure codes</i> utilizam menos <i>bandwidth</i> e armazenamento que replicação simples em sistemas com tempo médio entre falhas semelhantes e que o uso de <i>erasure codes</i> aumenta o tempo médio entre falhas para o mesmo armazenamento e mesmo <i>repair epoch</i> , quando comparado com a replicação simples.
Comentários:	Os autores não avaliaram falhas como erros de software, erros operacionais, problemas de configuração para esse trabalho.
Expressões-chave:	<i>Availability, Distributed Storage System, Replication, Erasure Code</i>
Observações:	Esse artigo foi sugerido nas notas de aula sobre <i>erasure codes</i> do curso Peer-to-Peer and Grid Computing http://www.cs.umd.edu/class/spring2007/cmsc818s/Lectures/lectures.htm acessado em 19/03/2010. O texto do artigo também foi acessado nesta data.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distribuição de Dados
Referência Bibliográfica:	Dabek, F.; Li, J.; Sit, E.; Robertson, J.; Kaashoek, M. F.; Morris, R. Designing a DHT for low latency and high throughput. In: <i>Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation (NSDI'04)</i> San Francisco, California, USA. p. 85–98, 2004.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.8345
Número de Páginas:	14
Citações (CiteSeer):	105
Número de Referências:	47
Visão geral do assunto do artigo:	Esse artigo comenta várias técnicas como <i>iterative routing</i> , <i>recursive routing</i> , <i>proximity routing</i> e <i>neighbor selection</i> , <i>erasure coding</i> , <i>replication</i> e <i>server selection</i> e as possíveis interações e implicações destas técnicas em algoritmos de leitura e de escrita em DHTs (<i>Distributed Hash Tables</i>) utilizando testes e medição através de parâmetros como latência média. Segundo os autores, DHTs são um caminho promissor para uma infraestrutura de armazenamento global e têm sido utilizadas como base para uma variedade de arquivos <i>wide-area</i> e para sistemas de publicação de conteúdo. Os autores comentam as condições assumidas para as discussões: todos os nós da rede cooperam; os algoritmos de leitura e de escrita ficam mais custosos, pois tratam acessos mal intencionados; <i>lookups</i> são roteados usando protocolos cujos algoritmos são $O(\log N)$, DHT armazenam blocos de 8K bytes.
Conclusões, Contribuições e Resultados Obtidos:	Uma contribuição desse trabalho é a exploração de grande conjunto de decisões de projeto para DHTs no contexto de um único sistema operacional. Os autores afirmam que dados replicados permitem leituras de baixa latência porque há muitas opções para a seleção de servidores, enquanto que dados <i>erasure-coded</i> reduzem o consumo de <i>bandwidth</i> para escritas em detrimento do aumento da latência de leituras. Os autores afirmam e comentam que replicação é um caso especial de <i>erasure codes</i> . Quando se usa replicação, o objeto de dados pode ser baixado a partir da réplica que está mais próximo do cliente, enquanto que, com a codificação, a latência de download é limitada pela distância para a m-réplica mais próxima. Este problema foi ilustrado com resultados de simulação.
hline Comentários:	Os autores não avaliaram os efeitos do tamanho do bloco, o que acontece quando nós entram e saem do sistema.
Expressões-chave:	<i>Data Hash Table</i> , <i>Replication</i> , <i>Erasure Code</i>
Observações:	Esse artigo foi escolhido por comentar algoritmos de leitura e escrita de DHTs, comparando replicação e <i>erasure codes</i> e por comentar decisões de projeto de DHTs. O texto do artigo foi acessado em 03/04/2010. Outros artigos estão em http://project-iris.net/

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distribuição de Dados
Referência Bibliográfica:	Rodrigues, R.; Liskov, B. High Availability in DHTs: Erasure Coding vs. Replication. In: <i>Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05)</i> . Ithica, New York, USA. p. 226–239, February 2005.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.6053
Número de Páginas:	6
Citações (CiteSeer):	35
Número de Referências:	17
Visão geral do assunto do artigo:	Esse artigo apresenta as vantagens de se usar as técnicas <i>erasure codes</i> e replicação para implementar DHTs (<i>Distributed Hash Tables</i>) de sistemas <i>peer-to-peer</i> . Inicialmente é feita uma definição estatística da disponibilidade de um objeto em sistemas que utilizam estas técnicas. Os autores fazem muitos comentários sobre os resultados obtidos com as duas técnicas em outras pesquisas. Depois descrevem o modelo básico utilizado para as comparações: uma coleção grande e dinâmica de nós que compartilham dados. O estudo foca na avaliação do custo da <i>bandwidth</i> para manter a redundância e no entendimento dos autores sobre o que é disponibilidade do servidor na DHT.
Conclusões, Contribuições e Resultados Obtidos:	Os autores assumem que um sistema que utiliza <i>erasure codes</i> mantém uma cópia completa adicional para cada objeto armazenado no sistema, dessa forma, aumentando em uma unidade o fator de replicação, nesse caso. A tarefa de transferir apenas um subconjunto particular do objeto (um sub-bloco) é também complicado pela codificação, onde o objeto inteiro deve ser reconstruído. Com réplicas, sub-blocos podem ser baixados trivialmente. O artigo é encerrado com as conclusões e com alguns desafios na pesquisa como estudar técnicas de redundância em dados alteráveis.
Comentários:	Na análise foram considerados apenas os dados imutáveis.
Expressões-chave:	<i>Data Hash Table, Replication, Erasure Code</i>
Observações:	Esse artigo foi escolhido porque comenta algumas desvantagens de <i>erasure codes</i> e replicação. O texto do artigo foi acessado em 19/03/2010.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Redundância em Sistemas Distribuídos
Referência Bibliográfica:	Houri, Y.; Jobmann, M.; Fuhrmann, T. Self-organized Data Redundancy Management for Peer-to-Peer Storage Systems. In: <i>Proceedings of the 4th IFIP TC 6 International Workshop on Self-Organizing Systems (IWSOS '09)</i> . Zurich, Switzerland. p. 65–76, 2009.
Tipo:	Teórico
Local:	http://dx.doi.org/10.1007/978-3-642-10865-5_6
Número de Páginas:	12
Citações (CiteSeer):	
Número de Referências:	22
Visão geral do assunto do artigo:	Nesse artigo os autores fazem uma comparação entre duas técnicas para se obter redundância em sistemas de armazenagem: <i>erasure codes</i> e replicação analisando apenas o custo de <i>bandwidth</i> para manter a disponibilidade de dados em um sistema de armazenamento baseado em DHT. Os sistemas comparados são sistemas de armazenagem <i>peer-to-peer</i> , onde os nós aleatoriamente entram e saem do sistema. Os autores descrevem o conceito de disponibilidade e de durabilidade de dados. Depois descrevem como alguns estudos e protótipos implementados usam estas duas técnicas, e que métricas foram usadas por eles para medir os resultados. Depois descrevem o modelo de sistemas <i>peer-to-peer</i> que utilizaram para as avaliações.
Conclusões, Contribuições e Resultados Obtidos:	O principal problema é que aumentando a quantidade de replicação também aumenta a probabilidade de que um dos nós envolvidos falhar. A probabilidade de perder um fragmento aumenta proporcionalmente ao número de nós que estão envolvidos no armazenamento dos fragmentos. Para cada ponto que falha, é preciso reparar a redundância e, portanto, consumir bandwidth de rede. Quanto mais redundância temos, maiores são as monitorização, manutenção e os custos de armazenamento. Os autores finalizam, descrevendo as conclusões que foram obtidas e fazem muitos comentários sobre os efeitos da variação de parâmetros como tamanho do bloco.
Comentários:	Os autores consideram apenas bandwidth pois consideram esse o único e mais importante critério para selecionar uma estratégia de redundância, pois os custos de bandwidth de hoje predominam sobre os custos de armazenamento: uma típica DSL up-link oferece até 1 Mbit/s, enquanto que um típico drive HD doméstico oferece até 1000 GB.
Expressões-chave:	<i>Data Hash Table, Distributed Storage System, Replication, Erasure Code</i>
Observações:	O texto do artigo foi acessado em 19/03/2010.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Redundância em Sistemas Distribuídos
Referência Bibliográfica:	Plank, J. S. ; Xu, L. ; Luo, J. ; Schuman, C. D. ; Wilcox-O'Hearn, Z. A Performance Evaluation and Examination of Open-Source Erasure Coding Libraries For Storage. In: <i>Proceedings of the 7th Conference on File and Storage Technologies (FAST '09)</i> . San Francisco, California, USA. p. 253–265, 2009.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.9642
Número de Páginas:	13
Citações (CiteSeer):	2
Número de Referências:	36
Visão geral do assunto do artigo:	O artigo faz uma comparação entre cinco implementações de código aberto de diferentes <i>erasure codes</i> . Os autores citam a nomenclatura de <i>erasure codes</i> que utilizaram no texto. É apresentado um sistema de armazenamento com k discos de dados e m discos de paridade utilizado nas comparações. Os autores descrevem quais são os tipos de <i>erasure codes</i> utilizados nas cinco implementações. Depois descrevem as cinco bibliotecas de código aberto de <i>erasure codes</i> que foram testadas. Das cinco bibliotecas testadas, Zfec implementou a codificação Reed-Solomon e Jerasure implementou as outras codificações (Cauchy Read-Solomon, Even-Odd, Row-Diagonal Parity ou RDP e Minimal Density RAID-6 codes). Os experimentos realizados foram de <i>encoding</i> de um grande arquivo de vídeo e de <i>decoding</i> de <i>drive</i> de dados. Depois descrevem as máquinas utilizadas nos experimentos: um mac-book com sistema operacional macosX e uma workstation dell com debian gnu linux.
Conclusões, Contribuições e Resultados Obtidos:	Os autores apresentam as conclusões que foram obtidas e as lições aprendidas.
Comentários:	
Expressões-chave:	<i>Open Source Erasure Coding Libraries</i>
Observações:	O texto do artigo foi acessado em 19/03/2010.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distributed Storage System
Referência Bibliográfica:	Fan, B. ; Tantisiriroj, W. ; Xiao, L. ; Gibson, G. DiskReduce: RAID for Data-Intensive Scalable Computing. In: <i>Proceedings of the 4th Annual Workshop on Petascale Data Storage (PDSW '09)</i> . Portland, Oregon, USA. p. 6–10, 2009.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.153.5950
Número de Páginas:	5
Citações (CiteSeer):	0
Número de Referências:	24
Visão geral do assunto do artigo:	Os autores propõem uma aplicação de <i>erasure codes</i> , o Diskreduce, uma modificação do HDFS (Hadoop <i>distributed file system</i>). O HDFS é o sistema de arquivos do Hadoop, um ambiente de programação paralela <i>open source</i> e que possui semelhanças com o GFS (<i>Google file system</i>). Para tratar falhas, o Diskreduce mantém 3 cópias de cada arquivo, assim como o Hadoop. O protótipo do Diskreduce é descrito em termos das técnicas de <i>erasure codes</i> utilizadas (RAID 6, RAID 5 e espelhamento), do número de linhas de código da implementação e da biblioteca de <i>erasure coding</i> utilizada, do número de nós da rede, do meio físico e do <i>kernel</i> das máquinas. O protótipo utilizado tem 63 nós com linux 2.6.28.10, 4TB de discos. A principal diferença entre o Diskreduce e o Hadoop é que aquele substitui várias cópias de dados de maneira assíncrona. Dessa forma, como a substituição é assíncrona, ela pode ser adiada no Diskreduce, até que a capacidade de reposição esteja disponível.
Conclusões, Contribuições e Resultados Obtidos:	Os autores afirmam que o atraso para codificar pode ser de 1 hora.
Comentários:	
Expressões-chave:	<i>Open Source Distributed Storage System, Replication, Erasure Code, RAID</i>
Observações:	Esse artigo foi escolhido porque comenta informações sobre o Hadoop, que será utilizado no meu trabalho de mestrado. O texto do artigo foi acessado em 19/03/2010.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distributed Storage System
Referência Bibliográfica:	De Camargo, R. Y.; Castor Filho, F. ; Kon, F. Efficient maintenance of distributed data in highly dynamic opportunistic grids. In: <i>Proceedings of the 2009 ACM Symposium on Applied Computing (SAC '09)</i> . Honolulu, Hawaii, USA. p. 1067–1071, 2009.
Tipo:	Teórico
Local:	http://doi.acm.org/10.1145/1529282.1529515
Número de Páginas:	5
Citações (CiteSeer):	0
Número de Referências:	15
Visão geral do assunto do artigo:	Os autores propõem um protocolo chamado OppStore que permite armazenar grande quantidade de dados utilizando técnicas para redundância em um ambiente dinâmico. Os gerentes de repósitório de dados são organizados em uma rede <i>peer-to-peer</i> com uma DHT. As outras máquinas disponibilizam espaço de armazenamento para os dados das aplicações e executam o módulo de repositório autônomo de dados. Os experimentos mediram a <i>bandwidth</i> utilizada pelo protocolo em um sistema de duas camadas.
Conclusões, Contribuições e Resultados Obtidos:	Os experimentos utilizaram uma pequena quantidade de dados: mais de 3 000 arquivos de diferentes tamanhos ocupando um total de 65GB.
Comentários:	No artigo, os autores não mencionaram qual classe de <i>erasure codes</i> foi utilizada.
Expressões-chave:	<i>Grid computing, Open Source Distributed Storage System, Replication, Erasure Code</i>
Observações:	O texto do artigo foi acessado em 03/04/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Channel code</i>
Referência Bibliográfica:	Mitzenmacher, M. Digital Fountains: A Survey and Look Forward. In: <i>Proceedings of the IEEE Information Theory Workshop (ITW '04)</i> . Santo Antonio, Texas, USA. p. 271–276, 2004.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.126.9463
Número de Páginas:	6
Citações (CiteSeer):	0
Número de Referências:	33
Visão geral do assunto do artigo:	O artigo introduz o conceito de <i>digital fountains</i> que é uma abstração de códigos corretores para aplicações de rede que não necessitam receber mensagens na ordem correta. Os autores comparam <i>digital fountains</i> a “uma fonte de água que se usa para encher um copo de água e não se tem preocupação se cai um pouco de água, mas apenas se quer saciar a sede”. Neste esquema, o receptor pode reconstruir uma mensagem que demanda quaisquer k pacotes codificados para envio usando TCP. Quatro algoritmos de códigos corretores são comentados em termos da complexidade da codificação e da decodificação: Reed-Solomon codes, Tornado codes, LT codes e Raptor codes.
Conclusões, Contribuições e Resultados Obtidos:	Os autores comentam sobre inúmeros requisitos de aplicações de <i>digital fountains</i> : os mecanismos de controle de fluxo e controle de congestionamento do TCP estariam mantidos e aplicações de <i>streaming</i> de vídeo seriam desafios. Depois os autores comentam as barreiras para adoção de <i>digital fountains</i> como proteção a patentes e o sucesso de serviços de entrega de conteúdo como akamai. Para os autores, o uso de códigos corretores de erros cresceu muito na última década e sua larga adoção ainda é muito pretensiosa.
Comentários:	É um artigo sobre teoria e não apresenta experimentos.
Expressões-chave:	<i>Network Communication, Erasure Code</i>
Observações:	O texto do artigo foi acessado em 22/04/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Channel code</i>
Referência Bibliográfica:	Shannon, C. E. A Mathematical Theory of Communication. <i>The Bell System Technical Journal</i> , v. 35, n. 11, p. 379-423, 623-656, 1948. Murray Hill, NJ, USA.
Tipo:	Teórico
Local:	http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf
Número de Páginas:	55
Citações (CiteSeer):	0
Número de Referências:	13
Visão geral do assunto do artigo:	O artigo introduz o conceito de codificação de mensagens. O autor demonstrou que a eficiência da codificação de mensagens do emissor antes da transmissão e da decodificação das mensagens (possivelmente danificadas) que chegam no receptor, possibilita reparar os efeitos de um canal ruidoso. Um sistema de comunicação foi definido com 5 partes: a fonte de informação, o emissor, o canal, destino e destinatário. O texto está dividido em 4 partes: sistemas discretos sem ruído, canal discreto com ruído, conceitos básicos de matemática, canal contínuo.
Conclusões, Contribuições e Resultados Obtidos:	Shannon demonstrou dois importantes teoremas que são fundamentais na comunicação ponto-a-ponto: <i>source coding theorem</i> e <i>channel coding theorem</i> . Shannon introduziu dois conceitos fundamentais sobre informação que é transmitida em um sistema de comunicação: a incerteza da informação e a informação transmitida é digital. Esse trabalho do Shannon parece ser o primeiro trabalho publicado que usa o termo <i>bit</i> (<i>binary digit</i>).
Comentários:	É um artigo considerado base para a teoria de códigos.
Expressões-chave:	<i>Network Communication, Erasure Code, Coding Theory</i>
Observações:	O texto do artigo foi acessado em 21/04/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure channel</i>
Referência Bibliográfica:	Purser, M. Chapter 1: Introduction. <i>Introduction to Error-Correcting Codes</i> , p. 1-10, 1995. Norwood, MA, USA.
Tipo:	Teórico
Local:	http://books.google.com/books?id=LglhQgAACAAJ
Número de Páginas:	10
Citações (CiteSeer):	0
Número de Referências:	17
Visão geral do assunto do artigo:	O capítulo 1 é uma introdução a codificação de blocos. O texto comenta redundância em textos (ortografia, sintaxe e semântica). Também comenta a função principal da codificação que é introduzir uma redundância controlada, possibilitando que mensagens corrompidas possam ser corrigidas antes de ser processadas. O texto define mensagens válidas como <i>codewords</i> ou <i>codevectors</i> . Segundo o texto, se um erro é detectado em uma mensagem, ela pode ser corrigida, à princípio, por um de 2 métodos: retransmissão da mensagem e correção da mensagem através de <i>Error correcting-codes</i> . O texto introduz o conceito de <i>erasure channel</i> . Comenta que a ocorrência de erros pode não ser uniforme e nem aleatória. Erros tem causas e se a causa for um pico de energia, um breve curto-circuito ou uma avaria na mídia magnética, é comum que isto afete mais de um bit. O texto comenta brevemente técnicas de correção utilizadas nos algoritmos Fire codes e Reed-Solomon codes. O texto introduz o a distância de Hamming, <i>sphere-packing</i> e teorema de Shannon.
Conclusões, Contribuições e Resultados Obtidos:	O texto tem uma abrangência muito boa para quase todas as classes de códigos mais utilizadas.
Comentários:	Trata-se de um livro de 133 páginas com mais capítulos sobre códigos lineares, códigos cíclicos, códigos BCH, códigos convolucionais e alguns apêndices como um sobre corpos finitos.
Expressões-chave:	<i>Redundancy, Erasure Code, Coding Theory</i>
Observações:	O texto do livro foi acessado em 30/04/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	Plank, J. S. A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems. In: <i>Journal of Software - Practice & Experience</i> , p. 995-1012, 1997. New York, NY, USA.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.7089
Número de Páginas:	19
Citações (CiteSeer):	127
Número de Referências:	28
Visão geral do assunto do artigo:	<p>O artigo é uma especificação completa para a implementação do algoritmo de codificação Reed-Solomon para sistemas <i>RAID-like: disk arrays, network file systems</i> e <i>distributed checkpointing systems</i>. O autor apresenta uma especificação completa do problema e do algoritmo da codificação e detalhes de sua implementação. O modelo estudado é formado por n dispositivos de armazenamento D_1, D_2, \dots, D_n (<i>data devices</i>) e outros m dispositivos de armazenamento C_1, C_2, \dots, C_m (<i>checksum devices</i>). O conteúdo de cada um dos m <i>checksum devices</i> é calculado a partir do conteúdo dos n <i>data devices</i>. O objetivo do cálculo dos C_i para $1 \leq i \leq m$ é tal que para quaisquer m dispositivos que falhem dos $D_1, D_2, \dots, D_n, C_1, C_2, \dots, C_m$, o conteúdo dos dispositivos que falharam possa ser reconstituído a partir dos dispositivos que não falharam. Inicialmente é apresentada a especificação do problema através de n <i>data devices</i> e m <i>checksum devices</i>. O objetivo é calcular os <i>checksum devices</i> de tal maneira que para quaisquer m <i>devices</i> que falhem, os outros que não falharam podem reconstruir o conteúdo dos que falharam. O modelo assumido inclui falhas que ocorrem quando um dispositivo falha, ele cai e o sistema reconhece esta queda. Não é o caso de erros como uma falha do dispositivo que é manifestada pelo armazenamento e recuperação de valores incorretos. Depois o autor fala dos três principais aspectos do algoritmo de RS-Raid: usar a matriz de Vandermonde para calcular e manter as palavras de <i>checksum</i> (soma de verificação), utilizar eliminação de Gauss para a recuperação de falhas e usar Galois <i>fields</i> para realizar operações aritméticas. O autor descreve o algoritmo em linhas gerais e dá um exemplo de sua execução com 3 <i>data devices</i> e 3 <i>checksum devices</i>. Os detalhes de implementação são apresentados através de um controlador RAID e um sistema de <i>checkpoint</i> distribuído. O autor apresenta outros métodos de codificação utilizados em sistemas <i>RAID-like</i> como paridade. O texto tem um apêndice sobre Galois <i>fields</i>.</p>
Conclusões, Contribuições e Resultados Obtidos:	O autor afirma que a abordagem (RS-RAID) baseada em tabelas de multiplicação e divisão sob corpo de Galois é apenas uma maneira de resolver o problema para sistemas com $n + m \leq 65536$, pois, nesse caso, é uma solução software eficiente que é fácil de implementar e não consome muita memória física.
Comentários:	Trata-se de .
Expressões-chave:	<i>Reed-Solomon, Raid, Redundancy, Erasure Code</i>
Observações:	Este artigo é um relatório técnico do <i>Department of Computer Science, University of Tennessee</i> . O autor escreve o texto para leitores não treinados em <i>coding theory</i> . Contém um apêndice sobre corpos de Galois. O texto do artigo foi acessado em 04/05/2010.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	<i>Erasure Code</i>
Referência Bibliográfica:	Plank, J. S. ; Simmermann, S. ; Schuman, C. D. Jerasure: A Library in C/C++ Facilitating Erasure Coding for Storage Applications - Version 1.2. Technical report of <i>Department of Computer Science, University of Tennessee</i> . Knoxville, TN, USA. 2007.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.3116
Número de Páginas:	39
Citações (CiteSeer):	6
Número de Referências:	19
Visão geral do assunto do artigo:	O autor inicia falando que <i>erasure codes</i> para aplicações de armazenamento de dados está crescendo em importância assim como cresce o tamanho e a complexidade dos sistemas de armazenamento. O modelo de sistema de armazenagem suportado é constituído de k dispositivos de dados e m dispositivos cujo conteúdo é calculado a partir dos k dispositivos originais. O autor afirma que se o algoritmo da codificação de apagamento tem a propriedade <i>Maximum Distance Separable</i> (MDS), então o sistema como um todo será capaz de se recuperar da perda de quaisquer m dispositivos. No texto é apresentada uma figura para ilustrar as operações de codificação e a decodificação. O autor define <i>word size</i> (w) que é o tamanho de um dispositivo codificado. Como na realidade os dispositivos tem megabytes ou gigabytes de dados, o autor sugere uma de duas possibilidades para mapear w : 1º se $w \in \{8, 16, 32\}$ Considere $w = 8$. O primeiro byte de cada um dos m dispositivos de codificação é codificado com o primeiro byte de cada um dos k dispositivos de dados e assim por diante. É desta forma que Reed-Solomon <i>codes</i> funciona. 2º se cada bit de codificação $C_{i,j}$ é uma operação <i>bitwise exclusive</i> (XOR) de uma sequência de outros bits. O dispositivo é composto de w pacotes de mesmo tamanho. Cada pacote é calculado por uma operação XOR de algum conjunto de pacotes. Uma figura mostra um exemplo. Depois o autor fala da biblioteca composta de 5 módulos e cada um deles é descrito em uma seção com código e exemplos de uso: módulo aritmética de corpo de Galois, módulo rotinas do kernel, módulo código RS, módulo código Cauchy-RS, RAID-6. O autor usa os produtos matriz-vetor e bit-matriz-vetor para mostrar a codificação do sistema. Depois o autor fala da propriedade MDS.
Conclusões, Contribuições e Resultados Obtidos:	O texto é um tutorial da biblioteca e um guia do programador.
Comentários:	Uma implementação de biblioteca em C/C++ para o algoritmo RS foi apresentada. O código pode ser baixado de http://www.cs.utk.edu/plank/plank/papers/CS-08-627.html .
Expressões-chave:	<i>Open Source Erasure Codes Library, Reed-Solomon, Raid, Erasure Codes</i>
Observações:	O texto do artigo foi acessado em 04/04/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	Luby, M. ; Mitzenmacher, M. ; Shokrollahi, A. ; Spielman, D. Analysis of low density codes and improved designs using irregular graphs. In: <i>Proceedings of the 30th annual ACM Symposium on Theory of Computing</i> , p. 249-258, 1998. Dallas, Texas, USA.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.6389
Número de Páginas:	10
Citações (CiteSeer):	64
Número de Referências:	17
Visão geral do assunto do artigo:	O artigo é uma proposta de código baseado em grafos irregulares e mostrara uma técnica de como projetá-lo. Os experimentos utilizaram código com taxa 1/2, mensagens de 16 mil bits e 8 mil check bits. Foram feitas 2 mil tentativas para 3 códigos diferentes: regular, code 14 e code 22.
Conclusões, Contribuições e Resultados Obtidos:	Os autores mostraram que uma classe de códigos corretores de erros em tempo linear corrige uma grande parte dos erros com alta probabilidade.
Comentários:	Trata-se de .
Expressões-chave:	<i>Low-density parity-check, Erasure Code, Irregular Graphs</i>
Observações:	O texto do artigo foi acessado em 04/05/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	GALLAGER, G. R. Low-Density Parity-Check Codes. <i>Revised Doctoral Dissertation, Department of Electrical Engineering, MIT</i> , 1963. Cambridge, Massachusetts, USA.
Tipo:	Experimental
Local:	http://www.rle.mit.edu/rgallager/documents/ldpc.pdf
Número de Páginas:	90
Citações (CiteSeer):	0
Número de Referências:	18
Visão geral do assunto do artigo:	Este relatório é a tese de doutorado do autor de 1960 que foi revisada.
Conclusões, Contribuições e Resultados Obtidos:	O texto .
Comentários:	Trata-se de .
Expressões-chave:	<i>Low-density parity-check, Erasure Code, Regular Graphs</i>
Observações:	O texto do artigo foi acessado em 04/05/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	OLIVEIRA, C. T. ; MOREIRA, M. D. D. ; RUBINSTEIN, M. G. ; COSTA, L. H. M. K. ; DUARTE, O. C. M. B. MC05: Redes Tolerantes a Atrasos e Desconexões. In: <i>Anais do 25o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos</i> , 2007. Belém, Pará, Brasil.
Tipo:	Experimental
Local:	http://www.rle.mit.edu/rgallager/documents/ldpc.pdf
Número de Páginas:	54
Citações (CiteSeer):	0
Número de Referências:	68
Visão geral do assunto do artigo:	O texto é do mini-curso
Conclusões, Contribuições e Resultados Obtidos:	O texto .
Comentários:	Trata-se de .
Expressões-chave:	<i>Low-density parity-check, Erasure Code, Regular Graphs</i>
Observações:	O texto do artigo foi acessado em 22/04/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	WOITASZEK, M. Tornado Codes for Archival Storage. <i>PHD Thesis, Department of Computer Science of the University of Colorado</i> , 2007. Boulder, Colorado, USA.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.160.1219
Número de Páginas:	136
Citações (CiteSeer):	0
Número de Referências:	71
Visão geral do assunto do artigo:	O artigo é
Conclusões, Contribuições e Resultados Obtidos:	O texto .
Comentários:	Trata-se de .
Expressões-chave:	<i>Low-density parity-check, Erasure Code, Irregular Graphs</i>
Observações:	O texto do artigo foi acessado em 08/05/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	Cabral, M. B. GRITO: Sistema de Medições para Suporte a Preservação Digital num Ambiente Heterogêneo. <i>Master Thesis</i> , Universidade Técnica de Lisboa, Instituto Superior Técnico, 2009. Lisboa, Portugal.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.160.1219
Número de Páginas:	80
Citações (CiteSeer):	0
Número de Referências:	0
Visão geral do assunto do artigo:	O artigo é
Conclusões, Contribuições e Resultados Obtidos:	O texto .
Comentários:	Trata-se de .
Expressões-chave:	<i>Erasure Code</i>
Observações:	O texto do artigo foi acessado em 03/05/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	White, T. Chapters 1, 2 e 3. <i>Hadoop: The Definitive Guide</i> , p. 1-74. Sebastopol, CA, USA, 2009.
Tipo:	Teórico
Local:	
Número de Páginas:	73
Citações (CiteSeer):	0
Número de Referências:	0
Visão geral do assunto do artigo:	O artigo é
Conclusões, Contribuições e Resultados Obtidos:	O texto .
Comentários:	Trata-se de .
Expressões-chave:	<i>Erasure Code</i>
Observações:	O texto do artigo foi acessado em 03/05/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	Kurose, J. F. Ross, K. W. Chapter 5. <i>Redes de Computadores e a Internet: uma abordagem top-down</i> , p. 318-376, 2010. São Paulo, Brasil.
Tipo:	Teórico
Local:	
Número de Páginas:	59
Citações (CiteSeer):	0
Número de Referências:	0
Visão geral do assunto do artigo:	O artigo é
Conclusões, Contribuições e Resultados Obtidos:	O texto .
Comentários:	Trata-se de .
Expressões-chave:	<i>Erasure Code</i>
Observações:	O texto do artigo foi acessado em 03/07/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	Dean, J. Ghemawat, S. MapReduce: simplified data processing on large clusters. In: <i>Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation</i> , p. 137-150, 2004. San Francisco, CA, USA.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.163.5292
Número de Páginas:	10
Citações (CiteSeer):	459
Número de Referências:	0
Visão geral do assunto do artigo:	O artigo é uma proposta de código baseado em grafos irregulares e mostrara uma técnica de como projetá-lo. Os experimentos utilizaram código com taxa 1/2, mensagens de 16 mil bits e 8 mil check bits. Foram feitas 2 mil tentativas para 3 códigos diferentes: regular, code 14 e code 22.
Conclusões, Contribuições e Resultados Obtidos:	Os autores mostraram que uma classe de códigos corretores de erros em tempo linear corrige uma grande parte dos erros com alta probabilidade.
Comentários:	Trata-se de .
Expressões-chave:	<i>Low-density parity-check, Erasure Code, Irregular Graphs</i>
Observações:	O texto do artigo foi acessado em 04/05/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure channel</i>
Referência Bibliográfica:	Huffman, W. C. Pless, V. Chapter 1. <i>Fundamentals of Error-Correcting Codes</i> , p. 1-52, 2003. New York, USA.
Tipo:	Teórico
Local:	http://books.google.com/books?id=LglhQgAACAAJ
Número de Páginas:	52
Citações (CiteSeer):	0
Número de Referências:	0
Visão geral do assunto do artigo:	O capítulo 1 é uma .
Conclusões, Contribuições e Resultados Obtidos:	O texto tem uma abrangência muito boa para quase todas as classes de códigos mais utilizadas.
Comentários:	Trata-se de um livro de 133 páginas com mais capítulos sobre códigos lineares, códigos cíclicos, códigos BCH, códigos convolucionais e alguns apêndices como um sobre corpos finitos.
Expressões-chave:	<i>Redundancy, Erasure Code, Coding Theory</i>
Observações:	O texto do livro foi acessado em 04/07/2010.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	Patterson, D. A. Gibson, G. Katz, R. H. A case for redundant arrays of inexpensive disks (RAID). In: <i>Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data</i> , p. 109-116, 1988. Chicago, Illinois, USA.
Tipo:	Experimental
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.2597
Número de Páginas:	6
Citações (CiteSeer):	0
Número de Referências:	0
Visão geral do assunto do artigo:	O artigo é uma proposta de código baseado em grafos irregulares e mostrara uma técnica de como projetá-lo. Os experimentos utilizaram código com taxa 1/2, mensagens de 16 mil bits e 8 mil check bits. Foram feitas 2 mil tentativas para 3 códigos diferentes: regular, code 14 e code 22.
Conclusões, Contribuições e Resultados Obtidos:	Os autores mostraram que uma classe de códigos corretores de erros em tempo linear corrige uma grande parte dos erros com alta probabilidade.
Comentários:	Trata-se de .
Expressões-chave:	<i>Low-density parity-check, Erasure Code, Irregular Graphs</i>
Observações:	O texto do artigo foi acessado em 04/05/2010.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Redundância em Sistemas Distribuídos
Referência Bibliográfica:	Chiola, G. An Empirical Study of Data Redundancy for High Availability in Large Overlay Networks. In: <i>Proceedings of the 2nd International Workshop on Hot Topics in Peer-to-Peer Systems (HOTP2 '05)</i> . Washington, DC, USA. p. 43–51, 2005.
Tipo:	Teórico
Local:	http://portal.acm.org/citation.cfm?id=1090948.1091374
Número de Páginas:	8
Citações (CiteSeer):	0
Número de Referências:	10
Visão geral do assunto do artigo:	O artigo apresenta um estudo que analisa a relação da redundância, implementada por replicação e por codificação por apagamento, com a disponibilidade de dados. Foi feita uma caracterização da replicação e da codificação, utilizando a sobrecarga de armazenamento (n cópias para replicação e n pedaços codificados e k pedaços originais para codificação), em termos das operações nos dados: criar, ler, nova versão, atualizar e eliminar. Os sistemas simulados tinham 60 peers, 1 000 peers e 500 000 peers. Os sistemas foram simulados através da técnica de simulação Montecarlo. Os experimentos utilizaram sistemas com diferentes replicações e codificações. Os gráficos mostraram a fração de peers que falharam e disponibilidade de dados.
Conclusões, Contribuições e Resultados Obtidos:	Os autores concluíram que usar apenas a replicação tem sentido apenas em poucos casos. O parâmetro k deve ser escolhido com base num equilíbrio entre eficiência da comunicação e a disponibilidade desejada. Os autores concluíram que o acesso somente de leitura pode ser suportado tanto por replicação de dados simples como por codificação, para privilegiar atualização consistente, uma codificação de alta disponibilidade é necessária, que se caracteriza por fracionamento do original dados em k pedaços e adicionando exatamente $k - 1$ pedaços e que se ler e a disponibilidade de atualização consistente são de igual importância, isso requer codificação $(k, 2k - 1)$.
Comentários:	É um estudo muito interessante.
Expressões-chave:	<i>Redundancy, Availability, DHT</i>
Observações:	O texto do artigo foi acessado em 25/06/2011.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distributed Storage System
Referência Bibliográfica:	Storer, M. W. Greenan, K. M. Miller, E. L. Voruganti, K. Pergamum: replacing tape with energy efficient, reliable, disk-based archival storage. In: <i>Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST'08)</i> . San Jose, California, USA. p. 1:1–1:16, 2008.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.130.1267
Número de Páginas:	16
Citações (CiteSeer):	0
Número de Referências:	24
Visão geral do assunto do artigo:	Os autores propõem um sistema MAID (<i>massive array of idle disks</i>) utilizando milhares de discos de dados. O sistema é projetado para aplicações do tipo uma escrita, várias leituras. O sistema substitue sistemas de armazenamento em fitas. O sistema é composto por milhares de <i>tomes</i> conectados por uma rede. Cada <i>tome</i> tem 4 componentes: <i>commodity hardware</i> para armazenamento, memória flash com metadados, CPU e porta de rede e executa seu daemon. Não existe um controlador central. As assinaturas de dados dos conteúdos de disco estão na NVRAM o que diminui o consumo de energia caso estivessem no disco. A codificação gera blocos redundantes de um conjunto de blocos de dados (<i>intra-disk</i>) e também gera redundância para os discos (<i>inter-disk</i>). No texto aparece uma seleção de alguns sistemas de armazenamento, comparando-se algumas de suas características. A estratégia mais utilizada nos sistemas avaliados é a codificação. Uma instalação de Pergamum pode conter apenas um tome ou até milhares deles. Os clientes submetem pedidos de arquivos de dados a um Pergamum tome através de uma interface. Entre os Pergamum tomes, os pedidos são de blocos de dados. Através da console, cada Pergamum tome reporta seu estado. Os blocos tem tamanho de 128KB até 1MB ou mais. Segmentos em um único disco são agrupados em regiões e um grupo de redundância é construído a partir de regiões de tamanhos idênticos em vários discos. Uma tabela com vários sistemas de armazenamento compara o custo de 10 PB. A sobrecarga de armazenamento para um sistema Pergamum com segmentos que utilizem 62 blocos de dados e dois blocos de paridade e os grupos de redundância de dados com 13 discos e três discos de paridade é $64/62 \times 16/13 - 1 = 0,27$ vezes a capacidade de dados. Foi utilizada simulação para estimar a confiabilidade dos dados pela métrica tempo médio de perda de dados através de fatores como taxa de falha do disco, taxa de reparação do disco, taxa de falha de setor, taxa de correção pela codificação. O protótipo do Pergamum é descrito em termos das técnicas de <i>erasure codes</i> utilizadas (XOR parity, Reed-Solomon), do número de linhas de código da implementação, das linguagens de implementação (Python e C), do <i>hardware</i> utilizado (ARM para Pergamum tome e intel duo core para cliente), do meio físico (rede ethernet), do <i>kernel</i> das máquinas com Pergamum tome(linux), do sistema operacional dos clientes (OS X) e do armazenamento (drive 7200 RPM SATA formatado com XFS).
Conclusões, Contribuições e Resultados Obtidos:	O custo de manutenção foi calculado em \$0.50 por gigabyte. Pelos experimentos, os autores mostraram que o Pergamum pode ter uso de energia de baixa, custo de hardware baixo, alta confiabilidade e gerenciamento mais simples.
Comentários:	Esse artigo tem interessantes referências.
Expressões-chave:	<i>Distributed Storage System, Erasure Code</i>
Observações:	O texto do artigo foi acessado em 25/06/2011

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distributed Storage System
Referência Bibliográfica:	Saito, Y. Frølund, S. and Veitch, A. Merchant, A. Spence, S. FAB: building distributed enterprise disk arrays from commodity components. In: <i>Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems</i> . Boston, MA, USA. p. 48–58, 2004.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.2631
Número de Páginas:	11
Citações (CiteSeer):	77
Número de Referências:	36
Visão geral do assunto do artigo:	Os autores propõem um sistema MAID (<i>massive array of idle disks</i>) utilizando milhares de discos de dados. O sistema é projetado para aplicações do tipo uma escrita, várias leituras. O sistema substitue sistemas de armazenamento em fitas. O sistema é composto por milhares de <i>bricks</i> conectados por uma rede. Cada <i>brick</i> tem <i>commodity hardware</i> para armazenamento, CPU, and NVRAM conectados por uma rede ethernet. Não existe um controlador central. Um algoritmo assíncrono baseado em voto Paxos é usado para manter a consistência de metadados entre os <i>bricks</i> . Uma instalação de Pergamum pode conter apenas um tome ou até milhares deles. Os clientes submetem pedidos de arquivos de dados a um <i>brick</i> coordenador (qualquer <i>brick</i>). O coordenador envia um pedido para os <i>bricks</i> de armazenamento desses blocos.
Conclusões, Contribuições e Resultados Obtidos:	O custo de manutenção foi calculado em \$0.50 por gigabyte. Pelos experimentos, os autores mostraram que o Pergamum pode ter uso de energia de baixa, custo de hardware baixo, alta confiabilidade e gerenciamento mais simples.
Comentários:	Esse artigo tem interessantes referências.
Expressões-chave:	<i>Distributed Storage System, Erasure Code</i>
Observações:	O texto do artigo foi acessado em 26/06/2011.

Categoria Genérica:	<i>Information theory</i>
Categoria Específica:	<i>Erasure code</i>
Referência Bibliográfica:	Duminoco, A. Data Redundancy and Maintenance for Peer-to-Peer File Backup Systems. <i>PHD Thesis, École Doctorale d'Informatique, Télécommunications et Électronique de Paris</i> , 2009. Paris, France.
Tipo:	Experimental
Local:	http://www.eurecom.fr/people/duminuco.en.htm
Número de Páginas:	175
Citações (CiteSeer):	0
Número de Referências:	111
Visão geral do assunto do artigo:	O artigo é uma tese de doutorado que propõe uma codificação que combina a eficiência da largura de banda da replicação com a eficiência do armazenamento dos clássicos <i>Erasure codes</i> . As seções avaliadas foram as 2 e 3 que tratam de trabalhos relacionados e desafios e custos.
Conclusões, Contribuições e Resultados Obtidos:	
Comentários:	Trata-se de .
Expressões-chave:	<i>Erasure Code, Peer-to-Peer, Storage</i>
Observações:	O texto do artigo foi acessado em 28/06/2011.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Reliability
Referência Bibliográfica:	Baker, M. Shah, M. Rosenthal, D. S. H. Roussopoulos, M. Maniatis, P. Giuli, TJ Bungale, P. A fresh look at the reliability of long-term digital storage. In: <i>Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys '06)</i> . Leuven, Belgium. p. 221–234, 2006.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.4521
Número de Páginas:	14
Citações (CiteSeer):	31
Número de Referências:	58
Visão geral do assunto do artigo:	O artigo comenta a importância da preservação de dados e as ameaças a ela. Os autores apresentam várias estratégias para reduzir a probabilidade de perda de dados.
Conclusões, Contribuições e Resultados Obtidos:	
Comentários:	
Expressões-chave:	<i>Distributed Storage System, Erasure Code</i>
Observações:	O texto do artigo foi acessado em 28/06/2011.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Redundância em Sistemas Distribuídos
Referência Bibliográfica:	Weatherspoon, H. Moscovitz, T. Kubiatowicz, J. Introspective Failure Analysis: Avoiding Correlated Failures in Peer-to-Peer Systems. In: <i>Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems (SRDS '02)</i> . Los Alamitos, CA, USA. p. 362–367, 2002.
Tipo:	Teórico
Local:	http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.2434
Número de Páginas:	6
Citações (CiteSeer):	27
Número de Referências:	19
Visão geral do assunto do artigo:	O artigo apresenta um estudo que analisa técnicas de tolerância a falhas: falhas bizantinas, codificação por apagamento, independência de falhas.
Conclusões, Contribuições e Resultados Obtidos:	Os autores concluíram que
Comentários:	
Expressões-chave:	<i>Redundancy, Fault tolerance, Peer-to-Peer</i>
Observações:	O texto do artigo foi acessado em 28/06/2011.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distributed Storage System
Referência Bibliográfica:	Ghemawat, S. Gobioff, H. Leung, S. The Google file system. In: <i>Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03)</i> . New York, NY, USA. p. 29–43, 2003.
Tipo:	Teórico
Local:	http://labs.google.com/papers/gfs.html
Número de Páginas:	15
Citações (CiteSeer):	77
Número de Referências:	12
Visão geral do assunto do artigo:	Os autores propõem
Conclusões, Contribuições e Resultados Obtidos:	
Comentários:	
Expressões-chave:	<i>Distributed Storage System, clustered storage</i>
Observações:	O texto do artigo foi acessado em 29/06/2011.

Categoria Genérica:	Sistemas Distribuídos
Categoria Específica:	Distributed Storage System
Referência Bibliográfica:	Weil, S. A. Brandt, S. A. Miller, E. L. Long, D. D. E. Maltzahn, C. Ceph: a scalable, high-performance distributed file system. In: <i>Proceedings of the 7th symposium on Operating systems design and implementation (OSDI '06)</i> . Seattle, Washington, USA. p. 307–320, 2006.
Tipo:	Teórico
Local:	http://portal.acm.org/citation.cfm?id=1298455.1298485
Número de Páginas:	14
Citações (CiteSeer):	81
Número de Referências:	37
Visão geral do assunto do artigo:	Os autores propõem o Ceph, um sistema de arquivos <i>open source</i> que possui três principais componentes: um <i>cluster</i> de servidores de metadados (que gerencia o namespace, nomes de arquivos e diretórios), um <i>cluster</i> de OSDs (dispositivos de armazenamento de objetos) que armazenam dados e metadados e os clientes que utilizam uma interface do sistema de arquivos. O Ceph agrupa dados em PGs (grupos de colocação) e usa uma função <i>hash</i> para distribuir os PGs nos OSDs, cujo algoritmo CRUSH é $O(\log n)$ e usa uma árvore-B para indexar os PGs. Existe um módulo em desenvolvimento que permite usar o Ceph como armazenamento para uma instância do Hadoop. O Ceph utiliza apenas replicação e está disponível para linux sob LGPL.
Conclusões, Contribuições e Resultados Obtidos:	
Comentários:	
Expressões-chave:	<i>Distributed Storage System, clustered storage</i>
Observações:	O texto do artigo foi acessado em 30/06/2011.