

DIAGRAM SITASI PAPER



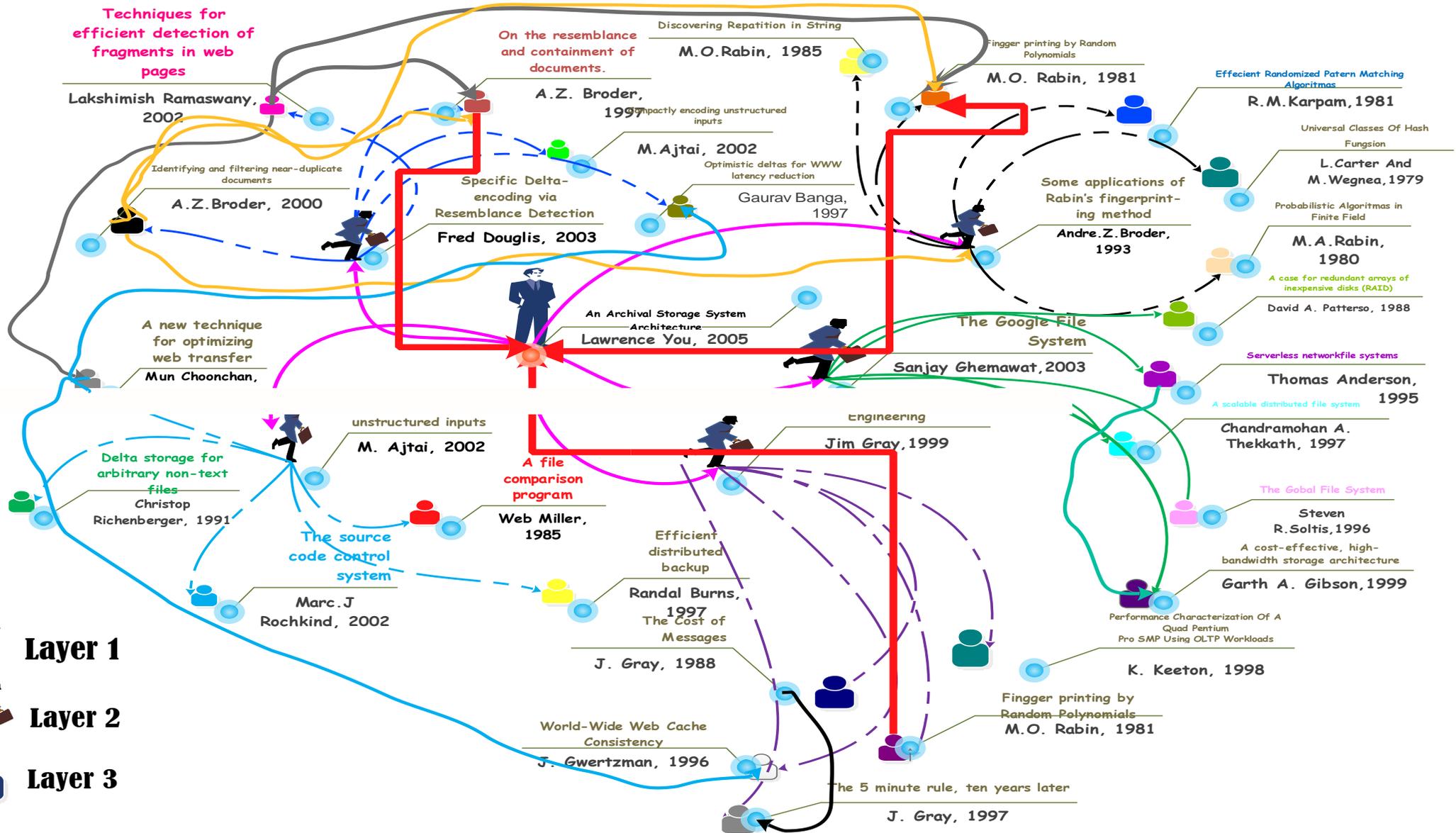
NAMA : ANASTASYA SEMBIRING

NIM : 09011181520017

KLS/JUR : SK2A/SISTEM KOMPUTER

FAKULTAS : ILMU KOMPUTER

Diagram Citasi Paper “An Archival Storage System Architecture”



An Archival Storage System Architecture

Penyimpanan cloud menjadi topik hangat untuk menangani masalah dan tantangan umum TI serta sembari menghadirkan peluang baru. Untuk beberapa lingkungan, tujuan utamanya adalah untuk memangkas biaya, sedangkan lainnya adalah untuk mendukung pertumbuhan. Selain itu, beberapa lingkungan perlu meningkatkan sasaran tingkat layanan (SLO) serta memenuhi perjanjian tingkat layanan (SLA) atas ketersediaan, kinerja, keamanan, dan perlindungan data. Dari pengertian di yang telah disimpulkan oleh banyak orang maka Lawrence L. You dalam papernya beliau membuat permasalahan mengenai kebutuhan sistem penyimpanan skala besar. Pada papernya yang berjudul “Deep Store: An Archival Storage System Architecture” menyatakan studi memperkirakan bahwa lebih dari lima exabyte (5×1060 byte) data diproduksi pada tahun 2002, meningkat lebih dari 30% dibandingkan tahun sebelumnya. Selanjutnya, fraksi data yang konten tetap atau data referensi terus meningkat, akuntansi untuk 37% dari semua data yang tersimpan, dan guna yang diharapkan untuk mengungguli data bisa berubah pada akhir tahun 2004. Hal ini tidak mengejutkan, terutama dalam menghadapi yang lebih dari 10.000 peraturan hukum ditempatkan di perusahaan di AS untuk kepatuhan perusahaan. Penyimpanan untuk kepatuhan meningkat 63% hanya pada tahun 2003, bahkan sebelum beberapa peraturan ing paling oleh permintaan, seperti Undang-Undang Sarbanes-Oxley, mulai berlaku.

Selanjutnya peracikan kebutuhan untuk penyimpanan arsip adalah meningkatnya volume material diubah menjadi yang Digital pada domail. Catatan permanen arsip, di mana data tidak dihapus, hanya akan terus tumbuh. Arsip Nasional dan Administrasi Records (NARA) bertujuan untuk memiliki 36 peta byte data arsip on-line pada tahun 2010. Sebagai hasil dari peningkatan tuntutan untuk penyimpanan referensi, untuk tujuan baik arsip dan kepatuhan, itu adalah daerah yang berkembang pesat bunga.

Meskipun jatuh biaya murah perangkat penyimpanan yang konsumen, biaya dikelola penyimpanan berbasis disk high-kali biaya perangkat penyimpanan itu sendiri dan lebih tinggi dari tape. Seperti baru-baru 2002, biaya untuk masukkan-storage hadiah disk yang lebih dari \$ 100 per gigabyte, dibandingkan dengan pita pada \$ 10 per gigabyte. Tren untuk dekat-line dan penyimpanan arsip adalah dengan menggunakan disk yang lebih murah, seperti perangkat ATA, bukan perangkat SCSI, dalam rangka untuk menurunkan biaya penyimpanan lebih dekat dengan tape magnetik.

Sebuah kelas baru sistem penyimpanan yang tujuannya adalah untuk re- tain volume besar data berubah sekarang berkembang. Tantangan rekayasa meliputi: meningkatkan skalabilitas, untuk mengakomodasi pertumbuhan jumlah konten arsip; im-membuktikan efisiensi ruang, untuk mengurangi biaya; meningkatkan reliabilitas, untuk mempertahankan data pada perangkat penyimpanan dengan singkat tahan ational oper- dan integritas data tidak memadai untuk penyimpanan arsip; dan mencari dan mengambil data dari dalam toko arsip. Tanda-tanda penyimpanan berbasis disk kinerja tinggi de- telah berkembang untuk menggunakan komponen biaya lebih rendah, tetapi mereka terus menjadi mahal untuk mengelola.

Metadata pasti akan melakukan peran penting dalam mengelola informasi sepanjang masa. Data dengan diri itu-adalah rapuh dalam

jangka panjang karena mungkin sulit untuk menafsirkan setelah bertahun-tahun, terutama ketika sistem yang dibuat tidak lagi ada. Interpretasi masa depan dan penyajian data membutuhkan metode yang dapat menggambarkannya. Unit utama untuk penyimpanan data pada network. Beberapa node terhubung melalui low-latency / bandwidth jaringan tinggi untuk membuat cluster storage. Setiap node berisi prosesor, memori, dan murah penyimpanan disk. Analisis isi, yang absen dari kebanyakan sistem berkas, termasuk sidik jari, kompresi, dan penyimpanan data dan pengambilan; ini operasi yang diperlukan, tetapi mereka harus memiliki throughput yang tinggi untuk menjadi praktis. Tingkat pertumbuhan kinerja untuk prosesor berbasis silikon dan kenangan historis telah lebih besar dari kapasitas dari disk magnetik. Mengingat semakin meningkat CPU-I / O kesenjangan, perbedaan ini menyiratkan manfaat kinerja potensi untuk bandwidth efektif dari pengurangan disimpan konten-kata lain, kompresi bisa mendapatkan keuntungan I / O per- kinerja.

Arsitektur software ini diwujudkan dalam proses yang mengeksekusi pada setiap node penyimpanan: layanan arsip penyimpanan, buffer penyimpanan sementara, analisa konten, dan toko beralamat isi-. Antarmuka penyimpanan arsip menerima masukan menggunakan antarmuka sistem umum seperti fungsi Interface mengambil posisi perintah in, pipa, soket, atau WebDAV, semua yang kami telah menemukan untuk menjadi cocok untuk client-server permintaan mech- anism sederhana. Buffer penyimpanan meminimalkan permintaan latency; implementasi kami menyimpan konten terkompresi di CAS a. Isi analisa menghilangkan redundansi dan juga dapat digunakan untuk mengekstraksi metadata. Toko CAS efisien konten dan mendata sama.

Dari permasalahan yang di munculkan dalam paper Lawrence L. You,2005 beliau juga mencitasi paper yang berjudul “Compactly Encoding Unstructured Inputs with Differential Compression” oleh Miklos Ajtai,2002. Pada paper Miklos Ajtai,2002 juga mencitasi paper paper terkenal yang membahas permasalahan mengenai Arsip Sistem Storage Architecture, dalam papernya kesimpulan yang dapat ditarik adalah subjek kompresi differential artikel ini, algoritmik menemukan string umum antara versi data dan menggunakan mereka untuk mengkodekan satu versi dengan menggambarkannya sebagai satu set perubahan dari pendamping. Tujuan utama dari pekerjaan ini adalah untuk menyajikan algoritma differencing baru yang beroperasi pada granularity baik (unit atom perubahan), tidak membuat asumsi tentang format atau keselarasan data input, dan dalam prakteknya menggunakan waktu linear, menggunakan ruang konstan, dan memberikan kompresi yang baik. Kami menyajikan algoritma baru, yang tidak selalu kompres optimal tetapi menggunakan lebih sedikit waktu atau ruang dari algoritma yang ada. Salah satu algoritma baru berjalan di O waktu dan $O(1)$ ruang dalam kasus terburuk (di mana setiap unit ruang berisi bit tertentu). Paper yang disitasi oleh Miklos Ajtai,2002 antara lain adalah “BURNS,R.C.,ANDLONG, D. D. E. 1997, MILLER,W.,ANDMYERS, E. W. 1985, REICHENBERGER, C. 1991, ROCHKIND, M. J. 1975, CHAN, M.,ANDWOO, T. 1999”

Pada paper lain yang disitasi oleh Lawrence L.You juga membahas tentang Arsip Sistem Storage Architecture menyatakan bahwa dalam merancang dan menerapkan Google File System, didistribusikan sistem file scalable untuk aplikasi data-intensif besar didistribusikan. Ini memberikan toleransi kesalahan sementara berjalan pada perangkat keras komoditas murah, dan itu memberikan kinerja agregat tinggi untuk sejumlah besar klien.

Sementara berbagi banyak tujuan yang sama seperti yang terdistribusi sistem file sebelumnya, desain kami telah didorong oleh variasi- dari kami beban kerja aplikasi dan teknologi ronment dari environmental, baik saat ini dan mengantisipasi, yang mencerminkan

keberangkatan ditandai dari beberapa asumsi sistem file sebelumnya. Hal ini telah menyebabkan kita untuk mempelajari kembali pilihan tradisional dan mengeksplorasi RAD poin desain ically yang berbeda.

Sistem file telah berhasil memenuhi kebutuhan penyimpanan kami. Hal ini banyak digunakan dalam Google sebagai platform penyimpanan untuk generasi dan pengolahan data yang digunakan oleh terlayannya kami serta upaya penelitian dan pengembangan yang memerlukan set data yang besar. Cluster terbesar sampai saat memberikan dreds hun- terabyte penyimpanan di ribuan disk di lebih dari seribu mesin, dan itu secara bersamaan diakses oleh ratusan klien. Pendapat ini dicetuskan oleh Sanjay Ghemawat,2003.

Dalam paper ini juga mencantumkan kesimpulan yang dapat ditarik oleh Jim Gray, 1999 bahwa Makalah ini reexamines aturan praktis untuk desain sistem penyimpanan data. Secara singkat, terlihat di penyimpanan, pengolahan, dan biaya jaringan, rasio, dan tren dengan fokus khusus pada kinerja dan harga / kinerja. Hukum rasio Amdahl untuk desain sistem hanya perlu sedikit revisi setelah 35 tahun-perubahan utama adalah peningkatan penggunaan RAM. Analisis juga menunjukkan penyimpanan harus digunakan untuk cache kedua database dan data web untuk menghemat bandwidth disk, bandwidth jaringan, dan waktu orang. Anehnya, aturan 5 menit untuk disk caching menjadi aturan cache-segalanya untuk web caching. Banyak aturan praktis adalah konsekuensi dari Hukum Moore, yang menyatakan bahwa kepadatan sirkuit meningkatkan 4x setiap tiga tahun. Itu berarti bahwa kenangan mendapatkan 4 kali lebih besar setiap tiga tahun, atau sekitar 100x per dekade. Ini juga berarti bahwa di-memori data tumbuh pada tingkat ini: menciptakan kebutuhan untuk sedikit tambahan menangani setiap 18 bulan. Pada tahun 1970 kami merasa nyaman dengan ruang alamat 16-bit: itu jarang untuk menemukan mesin dengan megaword memori. Tiga puluh tahun kemudian kita perlu 20 ekstra alamat bit untuk mengatasi GB kenangan 64 (36 alamat bit) yang ditemukan di komputer yang lebih besar di pasar. Saat ini sebagian besar komputer arsitekturmemberikan 64-bit logis (misalnya MIPS, Alpha, PowerPC, SPARC, Itanium) atau 96-bit (misalnya AS400) menangani. Fisik menangani adalah 36-bit untuk 40- bit, dan tumbuh sedikit per 18 bulan. Pada tingkat ini akan membawa dua atau tiga dekade melebihi 64-bit.

Paper ini juga terdapat pencitasion dari makalah orang yanglain yang membahs tentang masalah yang sama, dalam dalam sitasi tersebut terdapat juga paper yang yang sama persisi dengan paper yang disitasi oleh Lawrance L. You. Dalam keadaan ini dapat disimpulkan bahwa mereka berdua saling menceritakan dan mencari solusi yang sama. Masih banyak paper yang terkain antara satu dengan yang lain dalam pembuatan paper Lawrance L.You. banyak sekali paper yang bagus yang disitasi oleh dia. Dan dari semua pencitasion dan penelitian yang telah dilakukan oleh Lawrance L.You beliau dapat menjelaskan mengenai :

- File sistem berbasis disk tradisional, yang meliputi penyimpanan direct atau jaringan terpasang (DAS / NAS) dan penyimpanan yang jaringan area usia (SAN), tidak memiliki sifat desir- mampu untuk penyimpanan arsip. Mereka dirancang untuk memiliki kinerja tinggi bukan tingkat tinggi permanen.
- Properti tambahan untuk membedakan jauh Store dari sistem penyimpanan arsip lainnya termasuk: latency jauh lebih rendah daripada sistem tape yang menggantikan, antarmuka yang sederhana dan desain, kemampuan (penting untuk petabyte- sistem penyimpanan skala mencari), dan aksesibilitas di seluruh dekade atau abad serta seluruh sistem lokal atau didistribusikan

- Unit utama untuk penyimpanan node penyimpanan Toko Jauh. Beberapa node terhubung melalui low-latency / bandwidth jaringan tinggi untuk membuat cluster storage. Setiap node berisi prosesor, memori, dan murah penyimpanan disk. Analisis isi, yang absen dari kebanyakan sistem berkas, termasuk sidik jari, kompresi, dan penyimpanan data dan pengambilan; ini operasi yang diperlukan, tetapi mereka harus memiliki throughput yang tinggi untuk menjadi praktis.
- Arsitektur software ini diwujudkan dalam proses yang mengeksekusi pada setiap node penyimpanan: layanan arsip penyimpanan, buffer penyimpanan sementara, analisa konten, dan toko beralamat isi-. Antarmuka penyimpanan arsip menerima masukan menggunakan antarmuka sistem umum seperti fungsi Interface mengambil posisi perintah in, pipa, soket, atau WebDAV, semua yang kami telah menemukan untuk menjadi cocok untuk client-server permintaan mech- anism sederhana. Buffer penyimpanan meminimalkan permintaan latency; implementasi kami menyimpan konten terkompresi di CAS. Isi analisa menghilangkan redundansi dan juga dapat digunakan untuk mengekstraksi metadata. Toko CAS efisien konten dan metadata sama.
- Kesederhanaan dalam desain ini termotivasi oleh kebutuhan untuk pelestarian jangka panjang, tetapi tidak dengan mengorbankan fleksibilitas. Untuk penyimpanan konten-addressable dari jenis yang berbeda menggunakan metode penyimpanan yang efisien-beda ferent, kita diperlukan antarmuka penyimpanan yang seragam. Menuju tujuan pelestarian jangka panjang, spesifikasi sederhana dan kemudahan implementasi membantu en- memastikan bahwa data tertulis hari ini dari satu sistem klien masih dapat dibaca dari sebuah sistem klien yang sama sekali berbeda dalam 1, 10, atau bahkan 100 tahun. Operasi ini membentuk dasar untuk operasi penyimpanan lainnya.
- Teknik redundansi-eliminasi termasuk Kendala ini com- dikenal intra-kompresi file atau com- aliran berbasis pressors seperti gzip yang bekerja dengan baik dalam file; dan antar-berkas teknik kompresi seperti penyimpanan tunggal-contoh dalam sistem CAS, kompresi delta, dan kompresi berbasis potongan, yang semuanya bekerja dengan baik di seluruh file. Tingkat kompresi bervariasi dengan konten, dan dalam beberapa kasus antar-file dan intra-berkas com- pression saling melengkapi.
- Data lain, seperti konten digital sering dibuat dan segera Zip / gzip untuk tujuan umum isi yang, JPEG untuk konten visual, dan MPEG / MP3 untuk konten audio dan video. Bahkan ketika kompresi statis teknik coding yang digunakan, probabilitas simbol berbasis konten berbeda dari file ke file. Hal ini membuat setiap bagian sub bersebelahan representasi dikompresi lebih unik dari serupa di antara file, sehingga sulit untuk menyadari setiap kompresi bulu- ada. Dalam hal ini, menduplikasi penekanan
- Untuk menemukan konten-addressable objek tunggal, pertama dle han- disajikan dengan Virtual Object Table. Alamat konten pegangan ini digunakan sebagai kunci hash untuk mencari lokasi penyimpanan data Blok Virtual yang direferensikan oleh tabel. Blok data Virtual diambil dan pegangan dibandingkan untuk perusahaan identitas.
- Metode versi penggabungan bekerja dengan baik karena kompresi XMilluses kamus, sehingga rasio kompresi secara dramatis meningkatkan sebagai file akan lebih besar karena data ditambahkan sangat mirip, dan hanya satu kamus diciptakan untuk semua versi.

Dengan cerdas menggabungkan versi kita harus mampu mencapai efisiensi ruang yang lebih baik, serta metode sederhana untuk "perjalanan waktu" antara versi. Salah satu kelemahan metode ini yang mungkin membuatnya tidak layak dalam beberapa kasus adalah bahwa metadata harus disimpan pada penyimpanan bisa berubah untuk menjaga semua versi dalam satu file XML. Hal ini mungkin tidak dapat diterima untuk beberapa sistem tergantung pada media penyimpanan dan tingkat keamanan yang diperlukan; Namun, metode differencing dapat digunakan dalam kasus ini.

- Ada beberapa keuntungan untuk metode ini. Komputerisasi ini digest cepat pada 83 MB / s untuk SHA-1 dan 227 MB / s untuk MD5 pada perangkat keras uji kami (Intel Pentium 4, non-HT 2.66 GHz, 533 MHz FSB, memory PC2100 ECC). Hanya satu fitur yang perlu dihitung, dan kunci hash tunggal dapat digunakan untuk mencari file dalam waktu yang konstan atau hampir konstan. Banyak file yang identik, dan jadi ketika mereka terdeteksi, jumlah tabungan adalah luar biasa.

Dari semua paper yang telah saya cantumkan didalam ini maka dapat ditarik dua kesimpulan bahwa sistem penyimpanan skala besar yang menyimpan data berubah secara efisien dan andal untuk jangka waktu yang lama. Data arsip disimpan di cluster node dan dicatat ke hard disk. Desain membedakan dirinya dari sistem file tradisional dengan menghilangkan redundansi dalam dan di file, distribusi konten untuk skalabilitas, bergaul metadata yang kaya dengan konten, dan menggunakan variabel tingkat replikasi berdasarkan pentingnya atau tingkat ketergantungan setiap potongan data yang disimpan. Kami mengevaluasi dasar-dasar desain kami, termasuk PRESIDIO, virtual konten-addressable storage kerangka kerja dengan beberapa metode untuk antar-file dan intra-berkas kompresi yang efektif membahas kemampuan variabel data-dependent kompresi data. Kami mengukur konten dan efisiensi penyimpanan data meta, menunjukkan kebutuhan untuk model replikasi gelar Variable-, dan memberikan hasil awal untuk kinerja storage.

Penjelasan yang dapat saya paparkan dari bagan yang telah saya buat adalah sebagai berikut. Paper Lawrence L.You,2005 telah mencitasi paper dari M.Ajtai,2002, Andre.Z.Broder,1993, Fred Douglass,2003, Sanjay,Ghemawat,2003, Dan paper dari Jim Gray 1999. Dalam paper yang telah di sitasi oleh Lawrence L.You terdapat juga paper yang sama yang telah disitasi oleh orang pembuat paper yang disitasi kembali oleh Lawrence.L.You,2005 yaitu paper yang disitasi oleh Andre.Z.Broder, yang bernama M.O.Rabin 1981, dan paper milik M.o.rabin ini juga telah disitasi oleh Jim Gray,1999. Dalam paper Fred Douglass,2003 terdapat juga paper yang telah disitasi oleh beliau yang juga disitasi oleh Lawrence.L.You,2005, paper tersebut diatas namakan oleh Andre.Z.Broder,1997. Paper yang dibuat oleh Andre.Z.Broder,1997 juga telah disitasi kembali oleh Lakshimish Ramaswamy,2002 dan disitasi kembali oleh beliau dalam judul paper yang berbeda pada tahun 2000. Dalam paper Andre.Z.Broder,1997 juga mencitasi paper milik M.O.Rabin 1981.