

Meningkatkan Keandalan Model Super Peer dalam mengatasi *Single Point Failure*

I Wayan Simri Wicaksana*

*Universitas Gunadarma

E-mail: iwayan@staff.gunadarma.ac.id

ABSTRAK

Peer-to-Peer (P2P) telah menjadi media yang populer saat ini untuk melakukan berbagi-pakai data dan sumber daya. Kepopuleran P2P dimulai oleh Napster dengan melakukan pertukaran file musik. Keuntungan P2P karena memiliki sifat terbuka, otonomi, dinamis, sehingga tidak memerlukan sistem server yang powerful dan mahal. Disisi lain kelemahan utama pada P2P adalah masalah *self-organization*, *discover sources*, *load-balancing*, dan *fault tolerance*. Karena itu banyak penelitian yang difokuskan untuk mengatasi permasalahan tersebut.

Pada paper ini kami akan mengemukakan arsitektur super peer [1] untuk mengatasi kegagalan super peer. Kami merancang dengan menggunakan gabungan pendekatan dari [6], [5] dengan *redundancy super peer* model. Kami mendiskusikan konsep perancangan super peer yang memegang peranan penting dalam pembangunan super peer.

Kontribusi dari pendekatan kami dapat dimanfaatkan untuk peningkatan keandalan dalam implementasi pada pencarian sumber informasi maupun pertukaran data dengan memanfaatkan multiple super peer. Kedepan, penelitian akan kami kembangkan ke prototipe untuk menguji pada aplikasi sebenarnya.

Kata Kunci : *fault tolerance*, *discover sources*, *peer-to-peer*, *super peer*.

1. Pendahuluan

P2P network modern menunjukkan beberapa aspek khusus yang membedakan dengan sistem terdistribusi tradisional. Jaringan terdiri dari ratusan, ribuan atau bahkan jutaan peer adalah tidak umum. Sebagai konsekuensi dari skalabilitas, karakteristik P2P akan menjadi sangat dinamis dikarenakan terjadinya *joint* atau *leave peer* ke dalam network pada waktu yang tidak dapat diprediksi.

Pada sebuah skala yang besar dan dinamis akan memberikan tantangan yang rumit untuk dikembangkan sebuah model topologi. Apakah dengan adanya otoritas sentral, topologi yang tetap dapat dilakukan untuk mengatasi hal ini. Karena itu, perubahan yang dinamis dilakukan pada topology bertingkat akan dipelihara secara desentralisasi. Topologi ini akan disebut model kooperasi antara peer / node yang tergantung dari kebutuhan aplikasi khusus.

Sampai saat ini, sebagian besar aplikasi P2P menggunakan Internet, belum menghadirkan topologi khusus yang berjenjang. Konsekuensi dari pilihan ini adalah masih menggunakan pendekatan komunikasi yang belum efisien seperti *flooding*. Walaupun pada saat ini telah ada usaha untuk menyempurkan topologi

dari P2P.

Sebagaimana aplikasi file-sharing yang dimulai dengan mempertimbangkan topologi yang terstruktur. Dengan menggunakan super peer, topologi ini akan dihandle melalui hirarki dua tingkat. Dimana peer yang lebih cepat dan reliabel dari peer 'normal' akan difungsikan seperti server dengan memberikan pelayanan kepada beberapa peer. Sebagai contoh, dalam hal file-sharing, sebuah super peer akan membuat index dari file yang di sharing, walaupun pertukaran file tetap terjadi langsung antara peer.

Paradigma super peer memungkinkan jaringan desentralisasi untuk berjalan lebih efisien dengan eksploitasi keragaman dan penyebaran beban dari mesin. Disisi lain, ini bukanlah turunan dari model client-server, model ini memungkinkan secara multiple, mengurangi *point of failure*, dan meningkatkan kesehatan dari P2P.

Lebih jauh, manfaat dari model super peer bukan terbatas hanya pada file-sharing. Sebagai contoh adalah mungkin melakukan pendistribusian sistem game, informasi interoperabilitas, distribusi penyimpanan dan sebagainya.

Tetapi untuk membangun dan memelihara topologi super peer adalah tidak sederhana. Seperti untuk menghadapi skalabilitas yang ekstrim, kemampuan *self-organize* dan *self-repair* dari sebuah super peer. Disisi lain adalah juga bagaimana memilih atau menentukan super peer, dan bagaimana mengurangi resiko single point of failure dari super peer. Sehingga hal ini menimbulkan pertanyaan: berapa jumlah super peer yang dibutuhkan? Bagaimana super peer saling berkomunikasi antara satu dan lainnya.

Kontribusi dari paper ini adalah akan menguji kebutuhan super peer dalam mengatasi single failure melalui simulasi, seperti melihat berapa super peer yang dibutuhkan dalam sebuah *cluster*, berapa *cost* untuk sinkronisasi antar super peer di dalam sebuah cluster.

Paper ini akan dibagi sebagai berikut, pada bagian pertama akan memberikan latar belakang dan permasalahan pada P2P dengan super peer. Bagian ke-dua akan menguraikan arsitektur dasar dari berbagai konfigurasi P2P. Bagian ke-3 akan mengevaluasi kebutuhan super peer pada sebuah cluster. Dan terakhir adalah kesimpulan dari paper ini.

2. P2P Arsitektur

2.1. Sistem Tersentral

Sistem tersentral adalah sistem topologi yang paling populer, sebagai contoh adalah client-server model. Semua fungsi dan informasi dipusatkan pada sebuah server (kadang kalau disebut 'hub') serta memiliki berbagai client (kerap disebut 'spokes') yang terhubung langsung untuk melakukan pengiriman dan penerimaan informasi. Aliran kontrol dan data semua akan melalui server pusat.

Keuntungan pertama dari sentralisasi adalah kesederhanaan. Karena semua data dikonsentrasikan pada sebuah tempat, sistem tersentralisasi akan mudah di peliharaan dan tidak ada permasalahan untuk data konsistensi dan koherensi. Sistem tersentral juga secara relatif lebih aman, karena hanya ada satu host yang perlu dilindungi.

Kerugian dari sistem tersentral adalah informasi yang hanya di hub akan merupakan sebuah *single point of failure* yang rawan. Hub juga akan merupakan sebuah *bottleneck* untuk skalabilitas dan unjuk kerja. Meskipun beberapa sistem melakukan peningkatan dan

reduksi hardware, tetapi ini tidaklah memberikan solusi yang tepat. Sehingga, meskipun sistem tersentralisasi telah banyak diterapkan pada sistem database dan web, kelemahan akan skalabilitas dan *fault-tolerance* membuat ini tidak memadai untuk penerapan aplikasi terdistribusi untuk tujuan umum.

2.2. Pure P2P

Hal utama dari pure P2P adalah kemampuan skalabilitas, karena peer dapat bergabung dengan jaringan dan segera memulai pertukaran data antar berbagai peer. Tetapi sistem terdesentralisasi juga cenderung memiliki masalah *fault-tolerance*, seperti *shutdown* pada berbagai peer, tetapi hal ini tidak secara langsung memberikan efek besar kepada sistem secara keseluruhan.

Problem utama dari pure P2P adalah efisiensi dalam mencari sumber data/informasi, Penerapan penggunaan dari mekanisme umum seperti flooding masih banyak kelemahan, seperti tingginya jalur jaringan, tergantung dari besaran *time-to-live* (TTL), tergantung dari *neighborhood*. Contoh pure P2P adalah Gnutella.

2.3. Hybrid P2P

Pada sebuah hybrid P2P, informasi kontrol akan dipertukarkan melalui sebuah server sentral, tetapi aliran data dilakukan secara langsung antar peer sebagaimana pure P2P.

Arsitektur ini adalah mengatasi masalah manajemen pada pure P2P. Server kontrol akan berfungsi sebagai agen monitor atas semua peer dan untuk memastikan bahwa informasi akan koheren.

Kelemahan dengan kontrol secara terpusat tetap terjadi, jika sentral down, maka sistem akan kehilangan kemampuan untuk melakukan pertukaran data. Akan tetapi, aplikasi saat ini tidak terpengaruh dengan kegagalan sentral sebagai aliran data antar peer.

Routing data pada hybrid dapat memberikan skalabilitas yang lebih baik dibandingkan sistem tersentral. Dimana sistem hybrid akan efektif untuk sistem aplikasi kritis, tetapi hanya dibatasi untuk skala yang kecil saja. Contoh hybrid P2P adalah Groove

2.4. Super Peer P2P

Pendekatan baru P2P adalah dengan memanfaatkan keunggulan topologi tersentral

yang dikawinkan dengan desentralisasi, ini disebut dengan Super Peer P2P. Pada super peer ada dua level, dimana level pertama adalah hubungan antar super peer yang mengikuti model pure P2P /desentralisasi, sementara setiap super peer akan memiliki anggota peer yang mengikuti model tersentral. Pemilihan peer menjadi sebuah super peer akan ada beberapa mekanisme, secara umum adalah mempertimbangkan bandwidth, *availability*, processor dan media penyimpanan. Contoh pada model super peer adalah Bibster dan Kazaa.

Beberapa keuntungan dengan super peer adalah :

- Pencarian akan menjadi lebih cepat dibandingkan yang lain, karena sistem akan dipecah menjadi set super peer (*cluster*) yang lebih kecil dengan anggota peernya. Sebagai contoh pencarian dengan *flooding* pada pure/hybrid P2P akan memerlukan $O(N)$, sementara untuk super peer akan memerlukan $O(N/M)$, dimana M adalah jumlah peer dalam sebuah super peer.
- Setiap super peer akan merupakan unit otonomi, tidak tergantung kepada server sentral.
- Super peer akan menjadi lebih *reliable* dalam memonitor anggota peernya.
- Pada pure P2P setiap peer akan memiliki kesamaan posisi dan tanggung jawab terhadap kemampuan jaringan dan komputasi. Ini akan memudahkan terjadinya penurunan unjuk kerja karena fragmentasi jaringan akibat kemampuan peer yang tidak memadai. Masalah ini diatasi pada super peer, karena hanya komputer yang *powerfull* akan berstatus menjadi super peer.

Meskipun super peer menjadi lebih efisien, skalabilitas, memudahkan manage, tetapi super peer juga merupakan *single point of failure* untuk member peernya. Salah satu pendekatannya adalah dengan membangun redundansi super peer pada cluster yang bersangkutan. Paper ini akan melihat seberapa jauh cost dan kebutuhan redundansi super peer.

Sebagari ringkasan, perbandingan dari empat model topologi jaringan dapat dilihat

pada tabel 1

Tabel 1
PERBANDINGAN MODEL JARINGAN P2P

Topologi	M	K	S	R
Tersentral	Ya	Ya	Tidak	Tidak
Pure P2P	Tidak	Tidak	Ya	Ya
Hybrid P2P	Ya	Ya	Ya	Tidak
Super Peer P2P	Ya	Ya	Ya	Ya
M=Management				
K=Koheren				
S=Skalabilitas				
R=Realibilitas				

3. Evaluasi

Evaluasi yang akan dilakukan adalah untuk mengetahui berapa banyak reduksi super peer yang dibutuhkan dalam sebuah cluster. Serta juga dilihat berapa besar biaya untuk menghandle *joint*, *live* peer maupun untuk sinkronisasi/update antara super peer.

3.1. Parameter dan Skenario Evaluasi

- Untuk evaluasi berapa banyak super peer yang diperlukana, akan dilakukan simulasi dengan memberikan nilai random untuk failur dari sebuah peer (MTBF- mean time between failure) serta jika mengalami kerusakan akan memerlukan waktu berapa lama untuk perbaikan (MTBR-mean time between repair). Untuk ini simulasi akan dijalankan sebanyak 100 kali, dan setiap simulasi dimodelkan utuk 500 hari (1 hari dimodelkan dalam 1 detik).
- Untuk evaluasi *cost joint* dan *leave* dari anggota peer terhadap super peer. Bebeapa paramater dimasukkan untuk membangkitkan nilai random dari jumlah peer yang *joint* dan *leave* beserta *cost joint* dan *leave*. Simulasi akan dijalankan sebanyak sepuluh kali, dimana setiap simulasi dimodelkan utuk 500 hari. Simulasi dijalankan dengan nilai jumlah peer anggota yang berbeda, yaitu 10, 20, 50 dan 100.
- Evaluasi *cost* untuk *update* antar super peer akan dilakukan dengan menghitung total *cost* setelah 100

kali update (di atas 500 hari). Simulasi akan dijalankan sepuluh kali untuk melihat costnya secara min, max dan average.

- Evaluasi tidak memperhitungkan beban jaringan dikarenakan pertukaran data antara peer, tetapi lebih melihat kepada beban dari super peer.

Parameter awal yang digunakan untuk simulasi dapat dilihat pada tabel 2, dengan keterangan NAw : nilai awal, NBw : nilai bawah dan NAt : nilai atas. Simulasi dijalankan pada processor Opteron 2.5GHz dengan memory 2GB dengan Linux Kernel 2.6. Sehingga hasil simulasi akan sangat bergantung dari mesin dan operating sistem yang digunakan.

Tabel 2
NILAI SETTING UNTUK SIMULASI

Keterangan	NAw	NBw	NAt	Unit
MTBF SP	0	50	200	hr
MTTR SP	0	1	5	hr
No of Peer Joint	0	0	5	p/10hr
No of Peer Leave	0	0	3	p/10hr
Cost Joint	82	20	6400	bytes
Cost Leave	100	-	-	bytes
Cost Update SP	82	10	10000	bytes
Time Up Date	0	1	10	hr

3.2. Hasil Simulasi

Hasil simulasi dapat dilihat pada tabel berikut dengan perincian sebagai berikut :

- Evaluasi kebutuhan akan jumlah super peer pada tabel 3.
- Evaluasi beban akan bandwidth joint dan leave peer pada culster pada tabel 4, 5 dan 6.
- Evaluasi beban akan bandwidth untuk sinkronisasi update antar super peer pada tabel 7.

Tabel 3
KEJADIAN KETIDAK TERSEDIAAN SUPER PEER

Jumlah SP	Ketiadaan SP	Kejadian/Simulasi
1 SP	100	4x /simulasi
2 SP	6	1x /simulasi
3 SP	0	0x /simulasi

Dari tabel simulasi di atas makan dapat ditarik sebuah analisis sebagai berikut :

Tabel 4
JUMLAH PEER SETELAH SEBUAH PERIODE

Jml awal Peer	min	max	average
10	37	78	58
20	67	96	84
50	87	124	101
100	130	155	147

Tabel 5
TOTAL COST UNTUK JOINT

Jml awal Peer	min	max	average
10	405,964	521,788	446,452
20	412,912	534,958	473,681
50	410,372	537,729	468,555
100	403,760	531,508	458,151

- Mengacu kepada tabel 3, ternyata dengan 2 super peer sudah memiliki kemampuan yang memadai, dengan 3 super peer dari hasil simulasi 100 kali, bahwa ternyata hasil dari simulasi super peer akan selalu tersedia.
- Mengacu kepada tabel 4, 5 dan 6, bahwa cost joint dan leave tidak terlalu berbeda terhadap perbedaan jumlah peer awal yang berbeda. Walaupun simulasi sudah disempurnakan dengan peningkatan jumlah *joint* dan *leave* akan sejalan dengan pertumbuhan jumlah anggota peer dengan model linear. Walaupun model linear tidak terlalu sesuai dengan kondisi real pertumbuhan anggota peer yang sulit ditebak. Biaya maksimal bandwidth untuk *joint* dan *leave* hanyalah sekitar 550 kilo bytes relatif kecil untuk kecepatan jaringan saat ini.
- Beban super peer untuk update antara super peer relatif rendah juga yaitu hanya sebesar 875 kilo byte, ini juga relatif kecil dibandingkan kecepatan jaringan saat ini.
- Melihat dari hasil di atas dengan redudansi 3 super peer adalah sangat memadai untuk mengatasi kegagalan super peer. Dan beban bandwidth untuk sinkronisasi antara super peer adalah relatif kecil untuk kondisi kecepatan jaringan saat ini.

Tabel 6
TOTAL COST UNTUK LEAVE

Jml awal Peer	min	max	average
10	6,266	7,588	7,061
20	6,107	7,378	6,796
50	6,043	7,492	6,906
100	6,038	7,478	6,621

Tabel 7
TOTAL COST UNTUK UPDATE ANTAR SUPER PEER

Simulasi ke	Hari ke	Cost 2 SP	Cost 3 SP
1	576	537,354	817,590
2	569	551,972	804,626
3	537	545,297	689,119
4	551	487,032	823,716
5	547	516,387	646,670
6	546	540,471	823,135
7	533	470,109	840,463
8	534	543,556	873,193
9	548	490,854	721,749
10	576	515,625	816,484
min	533	470,109	646,670
max	576	551,972	873,193
average	551.7	519,866	785,675

4. Penutup

Super peer merupakan jaringan yang efektif karena menggabungkan model terpusat dan pure P2P untuk mendapatkan keunggulan dari kedua model topologi dan saling meniadakan kelemahan yang dimilikinya.

Permasalahan *single point of failure* dari super peer dapat dikurangi dengan melakukan redundansi super peer. Dari evaluasi berdasarkan simulasi, didapatkan bahwa dengan 3 super peer sudah sangat memadai untuk mengatasi problem ini. Disisi lain, beban jaringan juga tidak meningkat secara drastis untuk kebutuhan sinkronisasi antara beberapa super peer dalam sebuah *cluster*.

Walaupun dari simulasi telah didapatkan hal yang menarik, tetapi perlu di uji coba dalam sebuah prototipe dan mendekati kondisi nyata di jaringan terbuka. Untuk itu untuk penelitian kami kedepan, akan mengembangkan prototipe untuk pengujian redundansi super peer model.

DAFTAR PUSTAKA

[1] Fiorano Software Inc. Whitepaper: Super-Peer Architectures for Distributed Computing. www.fiorano.com, 2003.

[2] L. Garces-Erice, E.W. Biersack, P.A. Felber, K.W. Ross, and G. Urvoy-Keller. Hierarchical Peer-to-Peer Systems. In *Proc of Parallel Processing Letters (PPL)*, volume 13, pages 643–657, December 2003.

[3] Michael Kleis. Hierarchical Peer-to-Peer Networks using Lightweight SuperPeer Topology. In *Proc. of 10th IEEE Symposium on Computers and Communications (ISCC'05)*, pages 143–148, Spain, 27 June 2005.

[4] Alper Tugay Mizrak, Yuchung Cheng, Vineet Kumar, and Stefan Savage. Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup. In *Proc. of the IEEE Workshop on Internet Application*, pages 104–111, SanJose, CA, USA, June 2003.

[5] Alberto Montresor. A Robust Protocol for Building Superpeer Overlay Topologies. Technical Report UBLCS-2004-8, Dept of Computer Science, Univ. of Bologna, Mura Anteo Zambony 7, 40127 Bologna, Italy, May 2004.

[6] Beverly Yang and Hector Garcia-Molina. Designing a Super-Peer Network. In *Proc. of 19th International Conference on Data Engineering (ICDE'3)*, pages 49–74, 2003.