“KAPITA SELEKTA”

IoT Middleware “SINA” (Sensor Information Architecture and Application)

Muhammad Azriansyah

09011281320006

SK7Pil

Jurusan Sistem Komputer Reguler

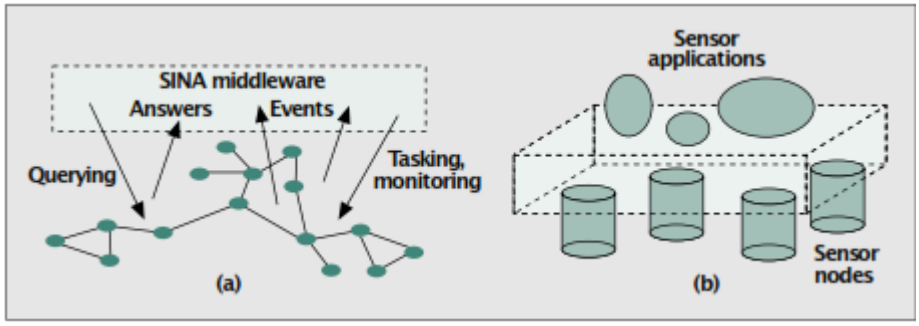
Fakultas Ilmu Komputer

Universitas Sriwijaya

2016

Pendahuluan

Munculnya teknologi telah memfasilitasi pengembangan perangkat kecil, rendah daya yang menggabungkan diprogram untuk keperluan umum komputasi dengan beberapa kemampuan komunikasi nirkabel dan penginderaan. Menyusun sensor node ini ke canggih ad hoc komputasi dan komunikasi infrastruktur untuk membentuk sensor jaringan akan memiliki dampak signifikan pada aplikasi mulai dari kesadaran situasi militer untuk proses pabrik control dan otomatisasi. Karena jumlah node sensor dan dinamika lingkungan operasi mereka (misalnya, terbatas daya baterai dan bermusuhan lingkungan fisik) menimbulkan tantangan-tantangan unik di Desain sensor jaringan dan aplikasi mereka. Isu-isu mengenai bagaimana informasi yang dikumpulkan oleh dan disimpan dalam sebuah sensor Jaringan bisa tanya dan diakses dan bagaimana bersamaan penginderaan tugas dapat dijalankan secara internal dan diprogram oleh eksternal pengguna adalah penting. Pada artikel ini kami menggambarkan sensor informasi jaringan arsitektur, disebut SINA, yang memfasilitasi query, pemantauan, dan tasking sen- Sor jaringan. Bagian berikut menjelaskan komponen dan informasi abstraksi arsitektur. Sebuah implementasi arsitektur, termasuk pemrograman sensor Bahasa disebut Sensor kueri dan Tasking bahasa (SQTL) dan lingkungannya eksekusi, digambarkan juga. Kami kemudian memperkenalkan operasi untuk informasi sensor, pengumpulan data dan menggambarkan isu-isu yang berkaitan dengan interworking antara pengguna ponsel dan node sensor stasioner. Contoh aplikasi untuk menggambarkan kemampuan pengumpulan operasi informasi dan SQTL juga disajikan bersama dengan studi simulasi mereka.



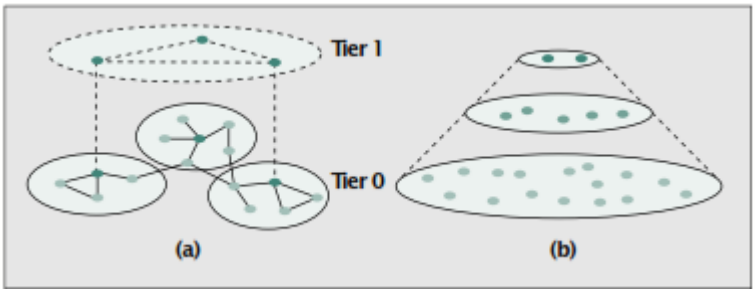
Model Sensor Network dan middleware SINA

SINA – Arsitektur Middleware

Secara konseptual, sensor jaringan dimodelkan sebagai kumpulan didistribusikan secara besar-besaran objek. SINA memainkan peran middleware, memungkinkan aplikasi sensor untuk mengeluarkan pertanyaan dan perintah tugas , mengumpulkan balasan dan hasil dari, dan memantau perubahan dalam jaringan . Modul SINA, berjalan pada setiap node sensor, menyediakan organisasi adaptif sensor informasi, dan memfasilitasi permintaan, acara pemantauan dan tasking kemampuan .

Berbeda dengan database didistribusikan konvensional di mana informasi yang didistribusikan di beberapa situs, jumlah situs di jaringan sensor sama dengan jumlah sensor, dan informasi yang dikumpulkan oleh setiap sensor menjadi bagian tak terpisahkan (atau atribut) simpul

tersebut. Untuk mendukung hemat energi dan operasi scalable, sensor node secara otonom tersusun. Selain itu, data-sentris sifat informasi sensor membuatnya lebih efektif dapat diakses melalui berbasis atribut penamaan pendekatan bukan eksplisit alamat. SINA arsitektur terdiri dari komponen-komponen fungsional berikut.



a. Clustering, b. Hirarki clustering

Hierarki Clustering — Untuk memfasilitasi operasi scalable dalam jaringan sensor, sensor node harus dikumpulkan untuk membentuk kelompok berdasarkan tingkat daya dan kedekatan. proses penggabungan juga bisa diterapkan secara rekursif untuk membentuk hirarki cluster. Dalam sebuah cluster, kepala cluster akan dipilih untuk melakukan penyaringan informasi, gabungan, dan agregasi, seperti perhitungan periodic suhu rata-rata cluster coverage area. Sebagai tambahan proses clustering seharusnya reinitiated seharusnya gugus kepala gagal atau menjalankan rendah pada daya baterai. Dalam situasi mana hirarki cluster ini tidak berlaku, system sensor node dirasakan oleh aplikasi sebagai satu tingkat pengelompokan struktur, dimana setiap node adalah kepala cluster dengan sendirinya. The Clustering algoritma yang diperkenalkan pada memungkinkan node sensor untuk secara otomatis membentuk kelompok, memilih dan memilih kembali kepala cluster, dan mengatur ulang struktur clustering jika diperlukan.

Atribut-Based Naming — Dengan populasi besar node sensor, mungkin tidak praktis untuk memperhatikan masing-masing masing-masing node. Pengguna akan lebih tertarik pada query daerah yang memiliki suhu yang lebih tinggi dari 100˚F, atau apa adalah suhu rata-rata di kuadran Tenggara, agak daripada suhu di sensor ID #101. Untuk memfasilitasi data-sentris karakteristik query sensor, berbasis atribut penamaan adalah pilihan skema. Misalnya, nama [jenis = suhu, lokasi = N-E, suhu = 103] menggambarkan semua sensor suhu yang terletak di kuadran timur laut dengan pembacaan suhu 103˚F. sensor ini akan menjawab permintaan "yang daerah memiliki suhu yang lebih tinggi daripada 100˚F."

Location Awareness — karena fakta bahwa sensor node beroperasi dalam lingkungan fisik, pengetahuan tentang fisik mereka sendiri lokasi sangat penting. Informasi lokasi dapat diperoleh melalui beberapa metode. Global Positioning System (GPS) adalah salah satu mekanisme yang memberikan informasi lokasi absolut. Untuk ekonomis alasan, namun, hanya sebuah subset

dari sensor node mungkin dilengkapi dengan GPS receiver dan fungsi sebagai referensi lokasi oleh secara berkala transmisi sinyal beacon memberitahu lokasi mereka sendiri informasi sensor node sehingga orang lain tanpa GPS receiver dapat menentukan posisi mereka perkiraan di Medan. Lainnya teknik untuk mendapatkan informasi lokasi juga tersedia. Untuk contoh, optik pelacak memberikan presisi tinggi dan resolusi informasi lokasi tetapi hanya efektif di wilayah kecil. Dengan integrasi dari tiga komponen ini, berikut dua contoh pertanyaan dapat dilakukan secara efektif:

- **Daerah yang memiliki suhu yang lebih tinggi dari 100°F?**

Dalam teori, permintaan broadcast untuk dan dievaluasi oleh setiap node dalam jaringan. Meskipun mungkin yang terbaik kembali hasil, permintaan akan menderita waktu respon panjang. Dalam prakteknya, masing-masing kepala cluster mungkin secara berkala memperbaharui suhu bacaan anggotanya, dan query sekarang dapat multicast untuk dan dievaluasi oleh kepala cluster hanya. Hal ini mengakibatkan lebih baik waktu respon dengan mengorbankan jawaban kurang akurat. Pertanyaan di bawah ketat waktu kendala dapat dievaluasi oleh kepala cluster tingkat lebih tinggi.

- **Apa adalah suhu rata-rata di kuadran Tenggara?**

Demikian pula, suhu rata-rata setiap cluster dapat secara berkala diperbarui dan cache oleh kepala cluster. Lebih lanjut-= lebih, permintaan harus dikirim ke node yang terletak (bernama) di Tenggara kuadran hanya.

Studi Kasus

Jaringan disimulasi terdiri dari 1024 sensor stasioner node didistribusikan dalam bentuk jaring dengan grid unit sama dengan 3 m, meliputi wilayah seluas ukuran 100 × 100 m. Masing-masing node dilengkapi dengan pemancar radio yang mampu transmisi sinyal hingga 5 m atas saluran nirkabel 2 Mb/s. Masing-masing transmisi kemudian mencakup sekitar delapan segera tetangga. Lapisan taut data menggunakan protokol 802.11, sementara lapisan jaringan mempekerjakan hanya IP fragmentasi fitur untuk komunikasi dengan segera tetangga tanpa apapun routing protokol. Semua diagnostik aplikasi yang berjalan di atas sumber UDP. Setiap node disediakan baterai dengan kekuatan yang cukup untuk setidaknya membuatnya mampu melaksanakan query lengkap operasi. Selain itu, untuk informasi tersebut metode pengumpulan yang memerlukan pengelompokan dukungan seperti APR, kami berasumsi bahwa clustering algoritma telah menyelesaikan terlebih dahulu, jadi setiap node harus memiliki pengelompokan informasi tentang induknya dan anak-anak sebelum diagnosis. Dalam simulasikami secara manual mengkonfigurasi mereka dalam kelompok dari sembilan sensor node dengan kepala cluster yang terletak di pusat setiap cluster. Sekali simulasi jaringan dimulai dan menjadi siap, satu node dalam jaringan, ditunjuk front-end, akan diminta untuk mengumpulkan informasi sensor. Node ini akan menyebarkan permintaan seluruh jaringan menurut untuk metode diagnosa yang berbeda yang digunakan.

	Total number of responses received	Fraction of expected responses received	Average response rate responses/s)	Number of MAC packets per response
Centralized approach (Prob[1] = 0.75)	229.0	29.8%	109.00	208.87
Self-orchestrated centralized (Prob = 0.75, *KH*[2] = 4 ms)	430.0	55.9%	107.50	138.17
Adaptive probabilistic response (ENRC[3] = 4)	183.8	40.4%	108.17	164.19
Diffused computation (Timeout[4] = 70 ms)	1016.0	99.2%	5080.00	14.70
Diffused computation with sampling (Timeout = 70 ms, Prob = 0.75)	767.0	99.8%	3068.00	12.78
Diffused Computation with Self-orchestration (Timeout = 70 ms, *KH* = 4 ms)	975.0	96.5%	3956.00	14.24

[1] Response probability. [2] Estimated hop delay and compensation. [3] Expected number of responses per cluster [4] Confirmation timeout

Hasil eksperimen dari menjalankan operasi diagnosis yang berbeda

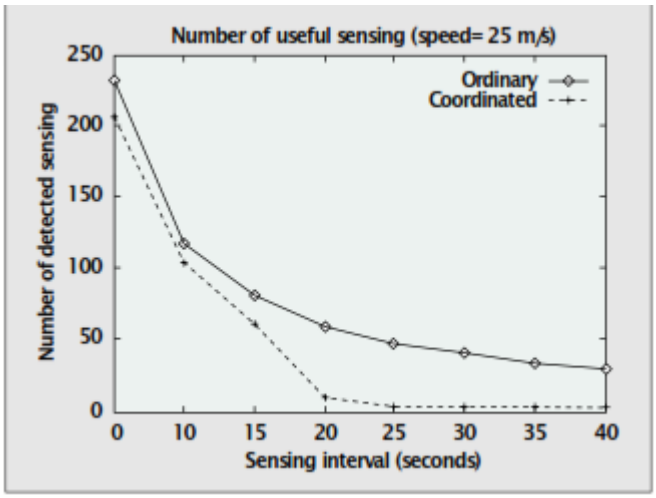
Hasil dan Analisis

Pada metrik pertama, total jumlah tanggapan yang menerima, memberi gambaran tentang berapa banyak node efektif berpartisipasi selama diagnosis. Alasan beberapa angka-angka tidak dibulatkan karena mereka diperoleh dari menjalankan percobaan beberapa waktu dan menghitung rata-rata. Berikutnya metrik ini fraksi menerima respon yang diharapkan. Ini mewakili sejumlah persentase dari tanggapan sehubungan dengan jumlah yang diharapkan dari pengaturan operasi. Sebagai contoh, kita diharapkan untuk melihat tanggapan 768 (75 persen dari 1024) diterima di frontend, tapi hanya ada tanggapan 229, yang merupakan 29.8 persen 768. Rata-rata metric tingkat respon menunjukkan respon dari masing-masing dalam jumlah tanggapan menerima per detik, diukur dari waktu tanggapan pertama tiba sampai respon terakhir menerima. Pengukuran terakhir adalah jumlah media akses kontrol (MAC) Paket dikirimkan setiap respon yang diterima, yang dimaksudkan untuk menunjukkan efisiensi masing-masing teknik dengan memberikan jumlah bandwidth jaringan yang digunakan (yang lebih rendah, lebih baik) untuk mendapatkan satu respon.

Karena jumlah besar tabrakan disebabkan oleh kurangnya respon penjadwalan, terpusat melakukan pendekatan tanpa orkestrasi diri buruk dengan segala cara. Tanggapan yang sebenarnya diterima jauh kurang dari diharapkan karena banyak paket dijatuhkan karena mengarah ke buffer overflow dan terbatasnya jumlah transmisi ulang pada lapisan MAC. Jumlah MAC paket per respon merupakan bukti baik hipotesis ini. Kinerja keseluruhan secara signifikan meningkatkan dengan bantuan diri orkestrasi. Seperti yang disajikan dalam tabel, adalah jumlah tanggapan hampir dua kali lipat, sementara jumlah paket MAC terlibat memotong hampir di setengah. Hasil berikutnya adalah dari the adaptif probabilistic pendekatan, satu-satunya metode yang digunakan pengelompokan mekanisme. Dalam konfigurasi, empat tanggapan diharapkan untuk diperoleh dari masing-masing gugusan sembilan anggota. Ini berarti 456 Tanggapan seharusnya telah menerima jika tidak ada paket kehilangan. Dari jumlah tersebut, hanya 184 (40.4 persen) benar-benar diterima.metode perhitungan disebarkan memberikan hasil yang sangat baik. Itu efektif membuat penggunaan dari

SINA aktif pemrograman (melalui SQTL skrip) untuk mendistribusikan perhitungan (menyebar) untuk setiap node. Dalam simulasi, perhitungan disebarkan melakukan rangkaian tanggapan di node sepanjang jalan kembali ke Front-end. Jumlah paket MAC bekerja berkurang jauh, mengakibatkan kurang kemungkinan tabrakan, hampir 100 persen Tanggapan, dan tingkat respons yang relatif tinggi.

Untuk mengintegrasikan sampling dan mekanisme diri diatur ke dalam perhitungan disebarkan secara teknis. Dalam gabungan disebarkan komputasi dan sampling metode, node akan menerima permintaan seperti biasa, tetapi akan menanggapi dengan probabilitas 0,75. Hasilnya adalah sedikit peningkatan oleh sebagian kecil dari respon yang diharapkan diterima, yang menjadi bahkan lebih dekat dengan 100 persen. Juga mengurangi kesempatan dari tabrakan di kanal, seperti yang ditunjukkan oleh menurun jumlah MAC paket per respon. Dengan terintegrasi disebarkan Perhitungan dan orkestrasi diri pendekatan, node yang menerima permintaan akan Jadwal diri mereka untuk mengirimkan konfirmasi kembali pesan untuk mengurangi kemungkinan tabrakan. Selain tingkat respon berkurang Disebabkan oleh tertunda konfirmasi, lainnyahasil tidak banyak berbeda dari orang-orang teknik murni disebarkan komputasi.



Angka dari sensing yang berguna dari 2 metode ketika kecepatan kendaraan meningkat hingga 25 m/s

Data yang diperoleh dari algoritma kedua. Tiga metrik yang menarik. Pertama, kita melihat efisiensi pelacakan dan pemantauan objek bergerak dengan mengukur rasio berguna penginderaan jumlah total penginderaan. Kita mendefinisikan penginderaan berguna sebagai penginderaan yang berhasil mendeteksi kendaraan. Kami menemukan bahwa jumlah sensings yang berguna dari kedua algoritma yang tepat yang sama (209) sementara algoritma biasa dilakukan jauh lebih banyak kegiatan penginderaan (76,800 kali dibandingkan dengan 8,828 kali Bila menggunakan algoritma terkoordinasi). Ini adalah karena kurangnya koordinasi antara node sensor biasa algoritma. Berikutnya, kita menghitung jumlah paket yang dikirim keluar dari semua node untuk periode seluruh simulasi. Jelas dari kolom ketiga dalam tabel 3 yang terkoordinasi algoritma dimanfaatkan

jaringan lebih banyak bandwidth daripada biasa satu. Paket tambahan ini menyumbang semua koordinasi-terkait Paket (yaitu, penindasan dan retracking pesan). Namun, ketika kita mempertimbangkan biaya total operasi, dikoordinasikan algoritma lebih baik, seperti yang ditunjukkan dalam kolom terakhir. Di sini, kita membandingkan biaya berdasarkan biaya total penginderaan (C S) dan transmisi (C T) dengan rasio C S : C = 4:1 , dan kemudian menormalkan biaya metode biasa untuk 1. Hasilnya menunjukkan bahwa metode terkoordinasi biaya 17,9 persen metode biasa.

Pada hasil dari skenario lain, dimana kendaraan bergerak lebih cepat pada 25 m/s. Kami bervariasi interval penginderaan Sementara menjaga parameter lainnya tidak berubah. Dari angka, jumlah sensings berguna yang Diperoleh dari terkoordinasi algoritma sedikit lebih rendah daripada yang Diperoleh dari biasa metode ketika penginderaan interval lebih rendah dari 15 s. alasan adalah bahwa dalam algoritma terkoordinasi, kita mencoba untuk melestarikan Jaringan sumber daya dengan menekan penginderaan aktivitas lebih jauhnya dari lokasi kendaraan dan memperingatkan hanya node terdekat. Oleh karena itu, jumlah node sensor pemantauan kendaraan adalah jauh lebih sedikit daripada metode biasa. Namun, pada penginderaan interval 20 s dan lebih, algoritma terkoordinasi hampir tidak berhasil mendeteksi kendaraan bergerak. Ini hasil menunjukkan bahwa proses reinitiation terkoordinasi algoritma tidak bisa bersaing dengan mobilitas tinggi kendaraan dan interval panjang penginderaan.

```
< execute>
    < sender>   FRONTEND < /sender>
    < receiver>       < group> NODE[0] < /group>
                      < criteria> TRUE < /criteria>
    < /receiver>
    < application-id> 118 < /application-id>
    < rum-hop> 0 < /rum-hop>
    < language> SQTL < /language>
    < with>
        < parameter type= "clocktype"     name= "trackingTime"          value= "600" />
        < parameter type= "clocktype"     name= "reTrackingTime"        value= "40" />
        < parameter type= "clocktype"     name= "trackingFrequency"     value= " 8" />
        < parameter type= "object"        name= "target"                value= "Vehicle1">
    < /with>
    < content> < ![CDATA[
        lastSensingResult =  false;
        timerApplication =  createTimer(trackingTime);    // instantiate a timer
        timerApplication.start();                         // turn it on
        timerReTracking =  createTimer(reTrackingTime);
        execute (ALL_NODES, "TRUE", MESSAGE["content"]);    // re-broadcast
        if ((sensor1 =  getMotionSensor()).turnOn())   {    // instantiate a sensor object
          upon {                                            //      and turn it on
              receive (msg) where msg["action"] == "tell" && msg["content"] ==
              "suppress": {
                  sensor1.standby(); break;
              }
              every (trackingFrequency): {
                  if (sensor1.detect(target)) {
                      tell (ALL_NODES,                "TRUE", "suppress");
                      tell (NEIGHBORS,                "TRUE", "retrack");
                      tell (MESSAGE["sender"],        "TRUE", "found");
                      lastSensingResult =  true;
                      timerReTracking.start();
                      break;
                  }
                  else lastSensingResult =  false;
              }
              expire (timerApplication): sensor1.turnOff(); exit(0);
          }
          upon {  // After one sensor node sees the vehicle
              receive (msg) where msg["action"] == "tell" && msg["content"] ==
              "retrack": {
                  if (timerReTracking.expired()) {
                      sensor1.turnOn();
                      timerReTracking.start();
                  }
              }
              receive (msg) where msg["action"] == "tell" && msg["content"] ==
              "found":
                  tell (MESSAGE["sender"],      "TRUE",     "found");
              every (trackingFrequency): {
                  if (sensor1.detect(target)) {
                      tell (MESSAGE["sender"],  "TRUE",     "found");
                      if (! lastSensingResult)
                          tell (NEIGHBORS,      "TRUE",     "retrack");
                      lastSensingResult =  true;
                      timerReTracking.start();
                  }
                  else {
                      if (lastSensingResult)
                          timerReTracking.restart();
                      lastSensingResult =  false;
                  }
              }
              expire (timerReTracking) : sensor1.standby();
              expire (timerApplication): sensor1.turnOff(); exit(0);
          }
        }
        else exit(1);
    ]]>  < /content>
< /execute>
```

Kode komplit STQL untuk algoritma tracking kendaraan terkoordinasi

Kesimpulan

Munculnya teknologi telah memfasilitasi pengembangan jaringan sistem kecil, rendah daya perangkat yang menggabungkan deprogram komputasi dengan beberapa sensing dan nirkabel kemampuan komunikasi. Segera lingkungan fisik kita akan tertanam dengan node sensor yang memungkinkan pengumpulan informasi baru dan kemampuan pengolahan. banyaknya sensor node dan dinamika lingkungan operasi mereka menimbulkanunik tantangan dalam bagaimana informasi yang dikumpulkan oleh dan disimpan dalam jaringan sensor dapat bertanya dan diakses, dan bagaimana serentak penginderaan tugas dapat dijalankan secara internal dan deprogram oleh klien eksternal. Artikel ini menjelaskan SINA sensor informasi jaringan arsitektur yang melayani peran middleware untuk memfasilitasi query, pemantauan, dan tasking sensor jaringan. Dengan mengintegrasikan hirarkis pengelompokan sensor node dan berdasarkan atribut berbasis mekanisme penamaan pada asosiatif siaran, SINA menyajikan spreadsheet asosiatifabstraksi yang memungkinkan informasi untuk diatur dan diakse menurut untuk aplikasi spesifik kebutuhan. Kernel SINA, mewakili oleh kumpulan melihat, menerapkan tiga komunikasi-tion paradigma — sampling, diri diatur, dan disebarkan komputasi operasi- untuk memfasilitasi pengumpulan informasi dan penyebaran. Di atas SINA kernel adalah substrat deprogram difasilitasi oleh bahasa SQTL untuk program penginderaan tugas. Sensor Jaringan query dan tasking aplikasi juga disajikan bersama-sama dengan studi simulasinya.

Referensi

[1] D. Estrin, R. Govindan, and J. Heidemann, “Embedding the Internet,” Commun. ACM, vol. 43, May, 2000, pp. 38–41.

[2] T. Imielinski and S. Goel, “World,” IEEE Pers. Commun., vol. 7, Oct. 2000, pp. 4–9.

[3] A. Ward, A. Jones, and A. Hopper, “A New Location Technique for the