دانشگاه تبریز
دانشکده ریاضی
گروه علوم کامپیوتر

پایان‌نامه

برای دریافت درجه کارشناسی ارشد در رشته علوم کامپیوتر

ارائه‌ی روش جدیدی برای کشف سرویس در محیط محاسبات فراگیر با استفاده از پایگاه‌داده‌ی فعال

استادان راهنما
دکتر آیاز عیسی‌زاده – دکتر سعید صالحی‌پور مهر

استاد مشاور
دکتر لیلی محمدخانلی

پژوهشگر
مریم صالح محمدزاد

دی ماه ۱۳۹۰

نام خانوادگی دانشجو: صالح محمدزاد **نام:** مریم

عنوان پایان‌نامه: ارائه‌ی روش جدیدی برای کشف سرویس در محیط محاسبات فراگیر با استفاده از پایگاه‌داده‌ی فعال

استادان راهنما: دکتر آیاز عیسی‌زاده – دکتر سعید صالحی‌پور مهر

استاد مشاور: دکتر لیلی محمدخانلی

مقطع تحصیلی: کارشناسی ارشد **رشته:** علوم کامپیوتر **گرایش:** سیستم‌های کامپیوتری **دانشگاه:** تبریز

دانشکده: ریاضی **تاریخ فارغ‌التحصیلی:** ۱۳۹۰/۱۰/۱۸ **تعداد صفحه:** ۶۰

کلید واژه‌ها: محاسبات فراگیر، کشف سرویس، پایگاه‌داده‌ی فعال، قوانین ECA و ECAD

چکیده: در محیط محاسبات فراگیر، تمام وسیله‌هایی که قصد برقراری ارتباط با یکدیگر را دارند با رویدادها سر و کار دارند. به همین دلیل، کشف سرویس‌ها و مدیریت رویدادها در این محیط از حساسیت و پیچیدگی محسوسی برخوردار است. با توجه به اهمیت موضوع، هدف این رساله ارائه‌ی شیوه‌ایست که با استفاده از پایگاه‌داده‌ی فعال و به‌کارگیری قوانین ECA (Event, Condition, Action)، پیچیدگی رویدادهای بین وسیله‌ها را کاهش داده و کارآیی آن‌ها را افزایش دهد. همچنین ویژگی دیگری به نام Domain جهت مشخص کردن ناحیه‌ای که عمل مورد نظر باید در آن رخ دهد، به قوانین ECA اضافه شده است تا به وسیله‌ی ECAD (Event, Condition, Action, Domain) کنترل بیشتری روی محیط و قوانین اعمال شود.

برای رسیدن به هدف این رساله، ارائه قوانین ECAD مناسب در پایگاه‌داده‌ی فعال مد نظر قرار گرفته است. این قوانین به شکلی طراحی شده‌اند که مدیریت رویدادها به صورت موثری عملی گردند و سپس خاتمه‌پذیری، هم‌ریزی و سازگاری آن‌ها مورد بررسی قرار گرفته است.

فهرست مطالب صفحه

چکیده ۱

۱- مقدمه ۱

۱,۱ اصطلاحات ۱

۱,۱,۱ محاسبات فراگیر ۲

۲,۱,۱ کشف سرویس در محیط محاسبات فراگیر ۴

۳,۱,۱ سیستم پایگاه‌داده‌ی فعال ۵

۴,۱,۱ کانال رویدادها ۷

۵,۱,۱ دامنه و وضعیت ۹

۲,۱ بیان مساله ۹

۳,۱ اهداف پایان‌نامه ۹

۴,۱ نظریه ۱۰

۵,۱ سازمان پایان‌نامه ۱۰

۲- کارهای مرتبط ۱۱

۱,۲ پیشینه‌ی کشف سرویس در محیط محاسبات فراگیر ۱۱

۱,۱,۲ Jini ۱۲

۲,۱,۲ UPnP ۱۳

۳,۱,۲ Salutation ۱۳

۴,۱,۲ Salutation-lit ۱۴

۵,۱,۲ SARA ۱۴

۶,۱,۲ SLP ۱۵

۷,۱,۲ DHT-Based SDP ۱۵

۸,۱,۲ FSSD ۱۵

۹,۱,۲ VSD ۱۶

۱۰,۱,۲ Bluetooth ۱۶

۱۱,۱,۲ mRDP ۱۷

فهرست مطالب | صفحه

EASY ۱۲,۱,۲ ۱۷
AIDAS ۱۳,۱,۲ ۱۸
Konark ۱۴,۱,۲ ۱۹
۲,۲ پایگاه‌داده‌ی فعال ۱۹
۳,۲ خانه‌های هوشمند ۲۴
۴,۲ جمع‌بندی ۲۴

۳- معماری پیشنهادی ۲۷
۱,۳ معماری محیط محاسبات فراگیر پیشنهادی ۲۷
۲,۳ انواع رویدادها در یک خانه‌ی هوشمند ۳۷
۳,۳ تعریف قوانین ECAD ۳۸
۴,۳ ارزیابی و مقایسه ۴۱
۱,۴,۳ تحلیل ناسازگاری قوانین ECA ۴۱
۲,۴,۳ تحلیل هم‌ریزی قوانین ECA ۴۲
۳,۴,۳ تحلیل خاتمه‌ی اجرای قوانین ECA ۴۳
۴,۴,۳ مقایسه روش‌ها ۴۴

۴- نتیجه ۴۵
۱,۴ در اثبات نظریه ۴۵
۲,۴ در تحقق اهداف پایان‌نامه ۴۶
۳,۴ کارهای مرتبط، بحث و مقایسه ۴۶
۴,۴ دستآوردهای پایان‌نامه ۴۶
۵,۴ موضوعات پژوهشی آینده ۴۷

مراجع ۴۸
چکیده انگلیسی ۵۱

فهرست شکل‌ها صفحه

شکل ۱-۱- سیر تکامل محاسبات ۲
شکل ۱-۲- ارتباط وسیله‌ها قبل از ظهور محاسبات فراگیر ۳
شکل ۱-۳- محیط محاسبات فراگیر ۴
شکل ۱-۴- مراحل اجرای قوانین ECA ۷
شکل ۳-۱- معماری محیط محاسبات فراگیر ۲۸
شکل ۳-۲- معماری عمومی کارگزار فعال اطلاعاتی محیط محاسبات فراگیر ۳۰
شکل ۳-۳- معماری داخلی کارگزار فعال اطلاعاتی محیط محاسبات فراگیر ۳۱
شکل ۳-۴- نمودار وضعیت درخواست‌ها ۳۵
شکل ۳-۵- وضعیت سرویس‌دهنده‌ها در APIS ۳۵
شکل ۳-۶- گراف اجرای قوانین APIS ۴۲
شکل ۳-۷- گراف فعالیت قوانین APIS ۴۳

فهرست جدول‌ها صفحه

جدول ۳-۱- لیست رویدادها ۳۱

جدول ۳-۲- قوانین ECA مدیریت منابع و مدیریت اشکال ۳۳

جدول ۳-۳- طبقه‌بندی قوانین ECA ۳۶

جدول ۳-۴ لیست قوانین و رویدادهای مربوطه ۴۱

جدول ۳-۵ اولویت دادن به قوانین ۴۳

جدول ۳-۶ مقایسه ویژگی‌های روش‌های مختلف کشف سرویس ۴۴

فصل ۱

مقدمه

یک چالش مهم در محیط محاسبات فراگیر، توسعه‌ی یک پروتکل کشف سرویس می‌باشد که به کاربران و برنامه‌ها این امکان را می‌دهد تا سرویس‌های مناسبی را که توسط اجزاء نرم‌افزاری و وسیله‌ها در محیط مهیا شده‌اند، کشف کنند و به کار گیرند. تاکید این رساله بر روی بهینگی کشف سرویس و رویدادهای تبادل شده در محیط است که برای این کار از «پایگاه‌داده‌ی فعال» استفاده شده است. سیستم پایگاه‌داده‌ی فعال از سازوکاری استفاده می‌کند که قادر است به صورت خودکار به رویدادهایی که در داخل یا خارج از سیستم اتفاق می‌افتد عکس‌العمل نشان دهد.

در این فصل به تشریح اصلی‌ترین اصطلاحات به کار رفته در رساله که عبارتند از محاسبات فراگیر[۱]، کشف سرویس در محیط محاسبات فراگیر، سیستم پایگاه‌داده‌ی فعال[۲]، کانال رویدادها، دامنه و وضعیت می‌پردازیم. این فصل با تشریح ساختار رساله به پایان می‌رسد.

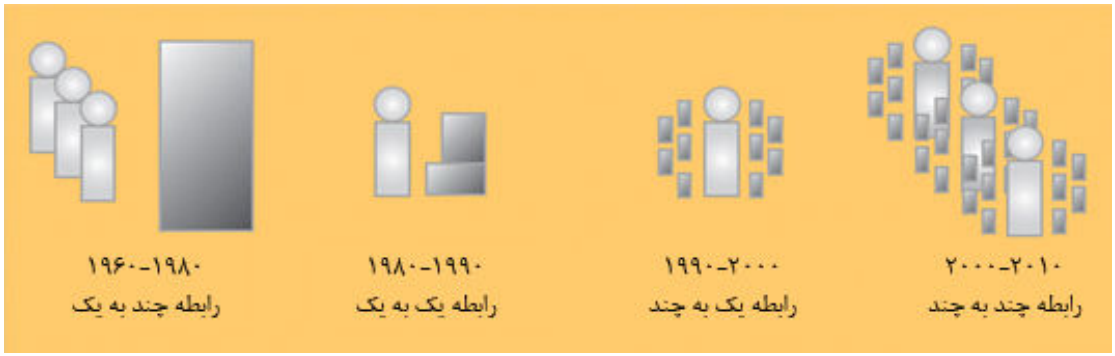
۱,۱ اصطلاحات

در این بخش به شرح اصلی‌ترین اصطلاحات به کار رفته در رساله می‌پردازیم:

[1] Pervasive Computing
[2] Active Database

۱،۱،۱ محاسبات فراگیر

پیشرفت‌های تکنولوژی به سرعت راه خود را در تمام جنبه‌های زندگی بشر می‌گشایند. این پیشرفت‌ها در نحوه‌ی خرید، برقراری ارتباط، جابه‌جائی از یک مکان به مکان دیگر و سایر ابعاد زندگی انسان‌ها تاثیر گذاشته‌اند. تمام این پیشرفت‌ها، حاصل تکامل کامپیوترها و قدرت محاسباتی آن‌ها، طی سالیان نه چندان طولانی است. شکل ۱-۱ روند تکامل کامپیوترها را به خوبی نشان می‌دهد.



شکل ۱-۱- سیر تکامل محاسبات

در سال ۱۹۹۱ مارک ویزر[۳] دیدگاه جدیدی را مطرح نمود که بر اساس آن، افراد توسط منابع محاسباتی، که در اشیاء موجود در محیط نهفته شده‌اند، احاطه خواهند شد و این منابع، بدون آن‌که انسان را متوجه حضور خود نمایند، خدمات مورد نیاز او را در هر زمان و مکانی، فراهم خواهند ساخت [۳]. وی این دیدگاه جدید را که شیوه‌ی نوینی در روش‌های پردازشی ایجاد می‌کرد، محاسبات «حاضر در همه جا» نامید که در حال حاضر از آن به عنوان محاسبات فراگیر یاد می‌شود. قبل از ظهور محاسبات فراگیر، ارتباط وسیله‌ها با یکدیگر بسیار محدود بود و انسان برای انجام هر کاری باید به صورت مستقیم با وسیله‌های موجود در محیط ارتباط برقرار می‌کرد (شکل ۱-۲)، ولی پس از حضور محاسبات فراگیر در محیط، وسیله‌ها توانستند، دو به دو با یکدیگر ارتباط برقرار کنند(شکل ۱-۳).

[3] Mark Weiser

شکل ۱-۲- ارتباط وسیله‌ها قبل از ظهور محاسبات فراگیر

هدف اصلی محاسبات فراگیر، حضور فن‌آوری در هر زمان و هر جائی است [۳]. محاسبات فراگیر، کامپیوترها را در محیط اطراف به گونه‌ای تعبیه می‌کند که در عین حالی که کاملا در دسترس و قابل استفاده هستند، از دید افراد پنهان باشند. برای رسیدن به این هدف، کامپیوترها باید به شکل ابزارهای محاسباتی کوچک درآمده و در مبلمان، دیوارها، ساختمان‌ها و غیره جا گیرند. برای مثال در یک محیط محاسبات فراگیر، یک ساختمان خواهد توانست خرابی خود را گزارش دهد و یا یک جلیقه نجات قادر خواهد بود تا خطر وجود کوسه در نزدیکی را اعلام کند.

محاسبات فراگیر نحوه‌ی تفکر ما در محیط اطرافمان را به شدت تغییر خواهد داد، چرا که بسیاری از اشیاء غیرمتحرک و بی‌جان را به موجودیت‌های با قابلیت دیدن و شنیدن و نشان دادن عکس‌العمل تبدیل می‌کند که می‌توانند با یکدیگر و یا با کاربرانشان ارتباط برقرار کنند، به تبادل اطلاعات بپردازند و حتی در صورت نیاز به یکدیگر سرویس دهند [۲].

در محیط محاسبات فراگیر مجموعه‌ی اجزاء محاسباتی که در هر سیستم توزیع شده شرکت می‌کنند، در طول زمان به صورت پویا عوض می‌شوند [۴]. به عبارت دیگر، یک کاربر یا اگر دقیق‌تر بگوییم، وسیله‌ی محاسباتی که در اختیار اوست، زمانی که شخص مکان خود را تغییر می‌دهد، به

صورت خودکار با سایر وسیله‌های موجود در محیط جدید، ارتباط برقرار می‌کند. ولی این نکته به این مفهوم نیست که تمام وسیله‌های موجود در محیط محاسبات فراگیر باید سیار باشند. بلکه اجزاء خاصی که باید بستر محیط محاسبات فراگیر را در مکان خاصی فراهم کنند، باید به صورت ثابت در آن مکان موجود باشند. به عنوان مثال در یک سالن کنفرانس، حس‌گرها، لپ‌تاپ سخنران و پروژکترها باید به صورت ثابت در محیط حاضر باشند. در حالی که مثلاً موبایل شرکت‌کنندگان، لپ‌تاپ آن‌ها و ... می‌توانند به صورت پویا در سالن حاضر شوند و در طول زمان از آن محیط جدا شوند.

شکل ۱-۳- محیط محاسبات فراگیر

۲،۱،۱ کشف سرویس در محیط محاسبات فراگیر

کشف سرویس، مسیری را برای کاربران و سرویس‌ها فراهم می‌کند تا در محیط محاسبات فراگیر با سایر سرویس‌ها تعامل داشته باشند و در آن مستقر شوند. سیستم‌های کشف سرویس از تکنیک‌های جستجوی مختلفی برای کشف یک سرویس استفاده می‌کنند. این تکنیک‌ها گاه مانند چندپخشی[۴] بسیار ساده هستند و گاه با به کارگیری تکنیک‌هایی مانند الگوریتم ژنتیک، پیچیده‌تر می‌شوند. سازوکار جستجو روی این نکته متمرکز می‌شود که چگونه پرس‌وجوها را در شبکه توزیع

[4] Multicasting

کند و چگونه سرویس مورد نظر را کشف کند. اگرچه سعی روش‌های پیشرفته بر این است که برای توزیع پرس‌وجوها هوشمندانه تصمیم‌گیری کنند، ولی مشکل اصلی، بهینه‌سازی و تحت پوشش قرار دادن ارتباطات است [۶].

ازطرف دیگر پرس‌وجوها و انطباق آن‌ها مطرح می‌شوند. این‌که چگونه پرس‌وجوهای مناسبی ایجاد شوند و این‌که پرس‌وجوهای کاربر با منابع موجود مطابقت کند. این مساله بیشتر مربوط به موضوعاتی مانند تجزیه[۵]، تعریف زبان[۶]، ساختار راهنماها[۷]، مدیریت جدول‌های هش[۸] و غیره است که موضوع بحث ما نیستند.

۳,۱,۱ سیستم پایگاه‌داده‌ی فعال

در سال‌های اخیر شاهد افزایش علاقه به استفاده از پردازش پرس‌وجوها در شبکه‌ی حس‌گرها بوده‌ایم. دلیل اصلی این امر، آگاهی زیاد در مورد پردازش پرس‌وجوها به عنوان یک روش محاسباتی مناسب برای برنامه‌های شبکه‌هائی که با حس‌گرها در تعامل‌اند، می‌باشد. برای دسترسی به این هدف می‌توان از پایگاه‌داده‌ی فعال استفاده کرد [۹].

برنامه‌های شبکه‌های حس‌گر در یکی از حالت‌های زیر می‌باشند:

- برنامه‌هائی که بر حسب رویدادها هستند. مثلا تشخیص آتش‌سوزی و مدیریت امنیت که تا زمانی که اتفاق خاصی در یکی از گره‌ها رخ نداده، سیستم غیرفعال است. سپس رویداد در کل سیستم منتشر می‌شود و در پاسخ به این رویداد، عمل خاصی رخ می‌دهد.
- برنامه‌هائی که بر اساس درخواست هستند. برای مثال مشاهده کردن محیط[۹].

با توجه به این‌که پردازش پرس‌وجوها با برنامه‌های دسته‌ی اول مطابقت دارد، بنابراین استفاده از قوانین ECA برای پاسخ به نیاز آن‌ها مناسب است.

طرح پایگاه‌داده‌ی فعال بعد از پیشنهاد چاکراوارتی [۷] برای اولین بار در سال ۱۹۹۹ ایجاد شد، که در آن یک کارگزار برای ارتباط با DBMS از قوانین ECA استفاده می‌کند. این روش به محدودیت سیستم‌های موجود مانند نبود امکان تشخیص رویدادهای مرکب غلبه کرد. یک DBMS

---

[5] Parsing
[6] Language Definition
[7] Directory Structure
[8] Hash Tables
[9] Environmental Monitoring

ارتباطی، مزایای نگهداری، کنترل هم‌زمانی و مدیریت تراکنش‌ها را دارد. توانایی فعال بودن نیز مزایایی مانند توسعه‌پذیری و قابلیت حمل را به سیستم می‌افزاید.

یک برنامه‌ی واکنشی باید قادر باشد تا وقوع رویدادهای خاص یا تغییرات در وضعیت شبکه را تشخیص دهد و به صورت اتوماتیک به آن‌ها پاسخ دهد [۹]. برای مثال در یک سناریوی مشاهده‌ی امنیتی که حس‌گرها قادر به تشخیص مواد شیمیایی خاصی در محیط هستند، علاوه بر گره‌های حس‌گر تعدادی محرک نیز با قابلیت فعال کردن هشداردهنده‌ها موجوداند. هدف این است که برنامه را طوری طراحی کنیم که هنگام اتفاق رویداد خاصی، هشداردهنده‌ی محیط فعال شود.

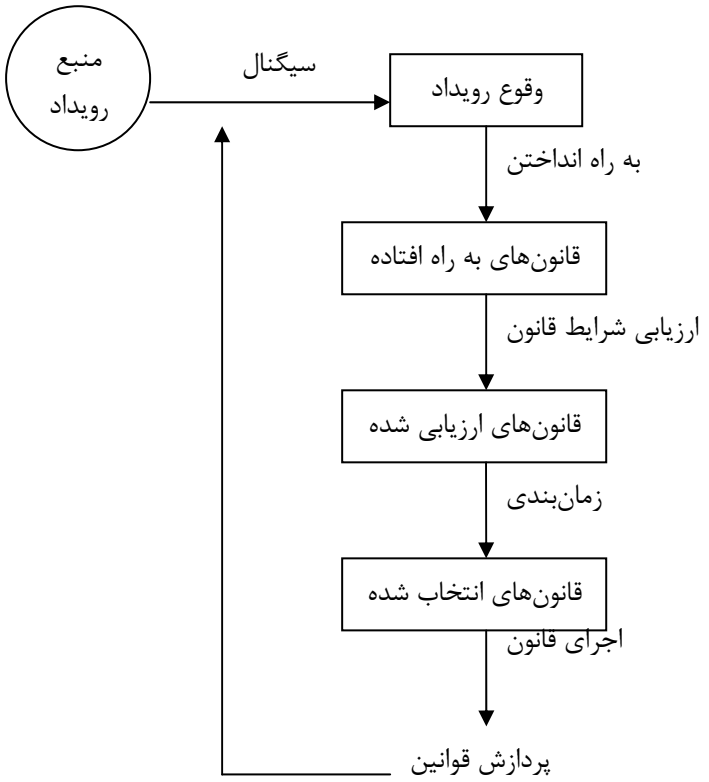
قوانین ECA یکی از روش‌ها برای رسیدن به این هدف می‌باشند که دارای شکل کلی ECA (Event, Condition, Action) هستند و از سه جزء: رویداد[۱۰]، شرط[۱۱] و عمل[۱۲] تشکیل می‌شوند. با توجه به مفهوم عبارت "on Event if Condition do Action" زمانی که پایگاه‌داده‌ی فعال وقوع رویدادی را تشخیص دهد، شرط را تست می‌کند، اگر نتیجه‌ی شرط درست باشد، عمل را اجرا می‌کند.

رویدادها یا از نوع ساده و یا از نوع مرکب هستند [۱]. رویدادهای ساده از نوع رویدادهای پایگاه‌داده، رویدادهای زمانی یا رویدادهای انتزاعی هستند. رویدادهای مرکب از ترکیب رویدادهای ساده با استفاده از عملگرهای منطقی تولید می‌شوند.

نحوه‌ی اجرای قوانین ECA که در شکل ۱-۴ نشان داده شده است، بدین صورت است که ابتدا با تشخیص وقوع رویداد، قوانینی که به این رویداد وابسته هستند، به مجموعه‌ی قوانین به راه افتاده[۱۳] اضافه می‌شوند. اگر شرط قانون درست باشد، این قانون به مجموعه‌ی قانون‌های فعال اضافه می‌شود. سپس با توجه به سیاست اتخاذ شده، یکی از قوانین از این مجموعه انتخاب شده و اجرای عملیات آن آغاز می‌شود. با اجرای عملیات، ممکن است رویدادهای جدیدی به وقوع بپیوندد و این روند تا زمانی که مجموعه‌ی منابع به راه افتاده و فعال تهی نشده است، ادامه می‌یابد [۱].

---

[10] Event
[11] Condition
[12] Action
[13] Triggered



شکل ۱-۴- مراحل اجرای قوانین ECA [۱]

۴,۱,۱ کانال رویدادها

یک کانال رویداد دو جزء دارد: مجموعه‌ای از گره‌ها و سازوکارهای الگوریتم‌وار که عملیات گره‌ها را هماهنگ می‌کنند [۹]. در یک کانال رویداد، گره‌ها، کار هماهنگی پردازش سیگنال‌ها و تجمیع داده‌ها[14] را بر عهده دارند و مسئول ذخیره‌سازی‌های درون‌شبکه‌ای و تولید رویدادها هستند. در نهایت گره‌های یک کانال رویداد، ساختارهای داخلی را که توضیحات کانال را نگهداری می‌کنند، منسجم می‌کنند. کانال رویدادها همچنین مسئول توزیع رویدادها با پیروی از قانون توزیع/تصدیق[15] (P/S)، است. سیستم‌های توزیع/تصدیق سعی می‌کنند تا منابع داده‌ها و مهیا کننده‌ی اطلاعات را توسط توزیع شفاف رویدادها یک‌جا جمع کنند. در این معماری، کانال رویداد، مسئول نگهداری لیست متقاضیان یک رویداد خاص و ارسال اطلاعات است. بنابراین گره‌های درخواست کننده ممکن است آزادانه در شبکه حرکت کنند و در جای دیگری به کانال، متصل شوند. یک کانال رویداد به

[14] Data Aggregation
[15] Publish/ Subscribe

صورت موثر به عنوان یک کارگزار رویداد نامتمرکز که از قوانین توزیع/تصدیق پیروی می‌کند، عمل می‌نماید.

ویژگی‌های خاص شبکه‌های حس‌گر، با توجه به نیاز ذخیره‌سازی و پردازش در شبکه، آن‌ها را با قوانین توزیع/تصدیق منطبق می‌کند:

- سیستم‌های توزیع/تصدیق با همان ویژگی‌های اساسی شبکه‌های حس‌گر توصیف می‌شوند که عبارت است از اینکه ارتباطات به صورت غیرهم‌زمان هستند. سیستم‌های توزیع/تصدیق همچنین می‌توانند خیلی زود با تغییرات به‌هم‌بندی شبکه منطبق شوند.
- سیستم‌های توزیع/تصدیق می‌توانند از عملیات نامتمرکز مدیریت رویدادها و توزیع در شبکه‌های حس‌گر پشتیبانی کنند. از آن‌جائی که محاسبات شبکه‌های حس‌گر نامتقارن است، بنابراین سازگاری محلی و کنترل محلی اهمیت زیادی دارد.
- ویژگی بی‌نامی توزیع/تصدیق به این معناست که گره‌هایی که در ارتباط‌اند، نیازی نیست که قسمتی را که می‌خواهند با آن ارتباط برقرار کنند، بشناسند (به همین دلیل است که درخواست‌کننده‌ها فقط کافی است رویدادی را که می‌خواهند دریافت کنند را شرح دهند و نیاز به ذکر نام توزیع کننده‌ی خاصی نیست) و بنابراین تجمیع داده‌ها می‌تواند بسیار شفاف پیاده‌سازی شود. به‌علاوه ویژگی بی‌نامی دلالت بر این دارد که از آن‌جائی که توزیع رویدادها برای درخواست کننده‌ها وابسته به گره حس‌گر واحدی نیست، چرخه‌های قابل تغییر فعال و غیرفعال شدن گره‌ها می‌توانند بسط داده شوند.
- سیستم‌های توزیع/تصدیق، رویدادها را به صورت همه‌پخشی ارسال می‌کنند. روشی که برای به‌روزرسانی مناسب است. ذخیره‌ی توان مصرفی برای این حالت مهم است و باید توجه کرد که با افزایش تعداد گره‌های شبکه باید توجه خاصی به این مقوله داشت.

استفاده از کانال رویدادها به عنوان هسته‌ی اصلی، این اجازه را می‌دهد که اجزای E و C و A یک قانون ECA کاملا مجزا از هم باشند. همچنین پرس و جوهائی که توسط قسمت شرط قانون ECA بیان می‌شوند، می‌توانند به صورت محلی پاسخ داده شوند و در برخی از حالات، داده‌ی لازم می‌تواند منتشر شود. همچنین با اضافه کردن گره‌های شرطی، کارایی سیستم نیز قابل تغییر است. این گره‌ها می‌توانند مسئول کنترل شرایط خاص در پاسخ به رویدادها باشند. در نهایت اگرچه اغلب تعداد گره‌های محرک کمتر از حس‌گرهاست ولی ساخت کانال‌های فعال‌سازی پیشنهاد مناسبی است.

۵،۱،۱ دامنه و وضعیت

یک خانه‌ی هوشمند می‌تواند به دامنه‌های مختلفی تقسیم شود. دامنه، یک نمایش انتزاعی از سیستم توزیع‌شده، شامل مجموعه‌ای از گره‌ها با ارتباط بین تمام یا بعضی از آن‌هاست. برای مثال تمام گره‌های یک اتاق و یا تمام گره‌های موجود روی دایره‌ای به شعاع r حول نقطه‌ای مانند x. دامنهها می‌توانند با یکدیگر در ارتباط باشند. همچنین می‌توان برای محیط، وضعیت تعریف کرد (مثلا گرم و سرد) و وضعیت‌ها را در شرایط مختلف عوض کرد.

۲،۱ بیان مساله

به نسبت افزایش تمایل به حضور محاسبات فراگیر در محیط، اهمیت ارتباطات بی‌سیم، کشف سرویس‌ها در محیط، کنترل امنیت و ... نیز افزایش می‌یابد. هدف اصلی برنامه‌های محیط محاسبات فراگیر این است که با به کار انداختن منابع یا سرویس‌های موجود در محیط، درخواست کاربر را اجرا کنند. برخی از درخواست‌ها نیاز به سرویس مشخصی دارند که به صورت مستقیم در محیط قابل ارائه است. درحالی که بعضی درخواست‌های دیگر به چند سرویس یا منبع اطلاعاتی نیاز دارند تا نتیجه‌ی مورد نظر را ارائه دهند. در هر صورت نیاز به یک زیرساخت کشف سرویس داریم که برای محیط محاسبات فراگیر مناسب باشد [۴].

با توجه به مطالب فوق، مساله‌ی اصلی این است که در محیط محاسبات فراگیر، چگونه می‌توان زیرساخت مناسبی برای کشف سرویس ارائه کرد تا پیچیدگی رویدادهای بین وسیله‌ها را کاهش و کارآیی آن‌ها را افزایش داد.

۳،۱ اهداف پایان‌نامه

مهمترین اهداف پایان‌نامه، پاسخ دادن به سوالات زیر است:

- چه قوانینی باید برای پایگاه‌داده‌ی فعال محیط محاسبات فراگیر تعریف شوند؟
- ضمن استفاده از پایگاه‌داده‌ی فعال، چگونه می‌توان کنترل بیشتری روی محیط داشت؟

۴٫۱ نظریه

با توجه به این‌که قبلا امکان استفاده از پایگاه‌داده‌ی فعال در محیط رایانش مشبک ثابت شده است [۱]، با در نظر گرفتن این‌که محیط محاسبات فراگیر نیز مانند محیط رایانش مشبک با پایگاه‌داده‌ها و رویدادها سر و کار دارد، می‌توان نتیجه گرفت که استفاده از قوانین پایگاه‌داده‌ی فعال در محیط محاسبات فراگیر برای کشف سرویس‌ها و مدیریت رویدادها نیز ممکن و مناسب است.

ویژگی دیگری را به نام Domain جهت مشخص کردن محیطی که عمل مورد نظر باید در آن رخ دهد، به قوانین ECA اضافه می‌کنیم تا کنترل بیشتری روی محیط و قوانین داشته باشیم.

۵٫۱ سازمان پایان‌نامه

ساختار رساله به این صورت است: در فصل دوم در مورد کارهای ارائه شده‌ی قبلی در زمینه‌های کشف سرویس در محیط محاسبات فراگیر، پایگاه‌داده‌ی فعال و خانه‌های هوشمند، بحث و بررسی شده و مزیت استفاده از قوانین پایگاه‌داده‌ی فعال، نسبت به سایر روش‌ها بیان شده است. در فصل سوم نیز ضمن شرح معماری پیشنهادی برای کارگزار فعال اطلاعاتی محیط محاسبات فراگیر، قوانین ECAD لازم تعریف شده‌اند و خاتمه‌پذیری، هم‌ریزی و سازگاری قوانین، مورد بررسی قرار گرفته است. در نهایت در فصل چهارم ضمن بررسی نتایج کار، عناوین پیشنهادی برای کارهای آتی نیز آورده شده است.

فصل ۲

کارهای مرتبط

در این فصل، به کارهای ارائه شده‌ی قبلی در زمینه‌های زیر پرداخته شده است:

- کشف سرویس در محیط محاسبات فراگیر
- پایگاه‌داده‌ی فعال
- خانه‌های هوشمند[۱۶]

۱,۲ پیشینه‌ی کشف سرویس در محیط محاسبات فراگیر

پروتکل‌های کشف سرویس فراوانی با هدف مشترک ارائه‌ی مکانیسمی برای شناساندن و کشف سرویس‌ها طراحی شده‌اند. تفاوت اصلی این پروتکل‌ها در معماری آن‌ها و محیطی است که می‌توانند در آن کار کنند. دو موضوع مهم قابل بررسی در بحث کشف سرویس، معماری کشف[۱۷] و سازوکار تطبیق سرویس[۱۸] می‌باشد [۱۰]. پروتکل‌هایی مانند Jini, Salutation, Salutation-lite, UPnP, UDDI, ... و پروتکل‌های تعیین محل سرویس، برای کمک به برنامه‌ها جهت کشف سرویس‌های راه دور در شبکه‌ها ارائه شده‌اند.

در این قسمت قصد داریم پروتکل‌ها و معماری‌های کشف سرویس را که قبلا برای محیط‌های توزیع‌شده یا فراگیر مطرح شده‌اند به صورت خلاصه مرور کنیم:

---

[16] Smart Homes
[17] Discovery Architecture
[18] Service Matching Mechanism

Jini ۱،۱،۲

Jini در سال ۱۹۹۸ بر مبنای تکنولوژی Java ارائه شد. Jini یک معماری توزیع شده‌ی مبتنی بر سرویس است که کشف، اتصال و مشاهده، ۳ پروتکل اصلی آن می‌باشند [۱۱].

کشف و اتصال زمانی روی می‌دهند که یک وسیله‌ی Jini به یک شبکه متصل می‌شود. کشف، زمانی روی می‌دهد که یک سرویس به دنبال کشف کننده‌ی سرویس است تا به وسیله‌ی آن اطلاعات خود را در شبکه ثبت کند و اتصال زمانی رخ می‌دهد که یک سرویس، کشف کننده‌ی سرویس را می‌یابد و از طریق آن متصل می‌شود. مشاهده زمانی روی می‌دهد که یک کاربر سرویس را طلب می‌کند.

تکنولوژی Jini از یک زیرساخت و یک مدل برنامه‌نویسی تشکیل شده است که نحوه‌ی ارتباط وسیله‌ها با یکدیگر را در نظر می‌گیرد. Jini از پروتکل RMI[۱۹] جاوا برای گردش کد در شبکه استفاده می‌کند.

سرویس مشاهده‌ی Jini را می‌توان به عنوان یک واسطه[۲۰] فرض کرد که اطلاعات پویا را در مورد سرویس‌های موجود نگهداری می‌کند. Jini سرویس‌های خود را بر پایه‌ی قرض[۲۱] واگذار می‌کند. یعنی یک کاربر می‌تواند سرویس را فقط برای مدت زمان خاصی درخواست کند تا Jini سرویس موردنظر را برای زمان درخواست شده قرض دهد. درصورتی که کاربر بخواهد سرویس موردنظر را برای زمان بیشتری از آن چه که قبلاً درخواست کرده بود، در اختیار داشته باشد، باید قبل از اتمام وقت، مجدداً استفاده از سرویس را تمدید کند. در غیر این صورت پس از اتمام مدت زمان توافق شده، Jini بلافاصله سرویس موردنظر را آزاد می‌کند. این خاصیت Jini باعث می‌شود تا هنگام از دست‌رفتن منابع، نیاز به بازبینی نباشد. در حقیقت این خصوصیت Jini قدرت تحمل خطا را در این زمینه افزایش می‌دهد.

---

[19] Remote Method Invocation
[20] Broker
[21] Lease

UPnP ۲,۱,۲

UPnP[۲۲] [۱۱] یک استاندارد مایکروسافت است که مدل PnP را توسعه می‌دهد. هدف آن فراهم کردن امکان آگهی دادن، کشف و کنترل گره‌های شبکه، سرویس‌ها و برق مصرف کننده‌ها است. در PnP یک گره می‌تواند به صورت پویا به یک شبکه متصل شود، یک آدرس IP بگیرد، هنگام درخواست، توانایی‌هایش را ارائه دهد و در مورد حضور و توانایی‌های سایر وسیله‌ها یادگیری داشته باشد. همچنین یک گره می‌تواند به راحتی و به صورت خودکار، شبکه را ترک کند بدون اینکه وضعیت ناخواسته‌ای را در شبکه ایجاد کند.

UPnP از پروتکل SSDP[۲۳] برای کشف سرویس استفاده می‌کند. این پروتکل حضور یک گره را به دیگران اطلاع می‌دهد و سرویس‌ها و وسیله‌های دیگر را کشف می‌کند. بنابراین SSDP نیز مانند پروتکل‌های سه گانه در Jini است. یک گره متصل شونده، یک اعلان را برای معرفی سرویس‌هایش به نقاط کنترلی ارسال می‌کند. نقاط کنترلی مانند سرویس مشاهده‌ی Jini کار می‌کنند. نقطه‌ی کنترلی اعلان را ثبت می‌کند. سایر وسیله‌ها نیز می‌توانند به صورت مستقیم اعلان وسیله‌ی جدید را مشاهده کنند. برخلاف Jini ، UPnP می‌تواند بدون حضور نقاط کنترلی نیز کار کند.

Salutation ۳,۱,۲

کنسرسیوم Salutation استانداردی به نام Salutation برای کشف سرویس به خصوص بین وسیله‌ها و سرویس‌ها با توانایی‌های ناهمسان ارائه داده است [۷]. معماری Salutation یک روش استاندارد برای برنامه‌ها، سرویس‌ها و وسیله‌ها جهت شرح و معرفی توانایی‌هایشان به سایر برنامه‌ها، سرویس‌ها و وسیله‌ها مهیا می‌کند. معماری Salutation این امکان را فراهم می‌کند که جستجو و کشف بر اساس توانایی‌های خاص انجام گیرد. این معماری از دو جزء اصلی تشکیل شده است:

۱- مدیر Salutation[۲۴]: که هسته‌ی معماری است و مانند سرویس مشاهده در Jini یا نقطه‌ی کنترلی در UPnP است و بیشتر به عنوان واسط سرویس شناخته می‌شود. یک

---

[22] Universal Plug And Play
[23] Simple Service Discovery Protocol
[24] Salutation Manager