



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

iLAND: یک میان افزار ارتقایافته برای پیکربندی دوباره زمان-واقعی

سیستم های زمان-واقعی توزیع شده خدمت-محور

چکیده- سیستم های تعبیه شده شبکه بندی شده آینده، عمیقاً پیچیده هستند و در محیطی ترکیب می شوند که تحت نظارت قرار می گیرند. آنها باید به رویدادهای کاربر و محیط واکنش نشان دهند و این کار به اصلاح ساختار آنها برای رفع و رجوع وضعیت های در حال تغییر نیاز دارد. در تعدادی از حوزه ها از جمله محیط های صنعتی، این اصلاح ساختار سیستم باید در زمان واقعی رخ دهد. این یک مسئله سخت است که به راه حل های الگودار و جدید مرتبط با دانش از حوزه ها برای ساخت یک میان افزار برای میسر نمودن تعامل زمان واقعی بین گره ها نیاز خواهد داشت که پیکربندی دوباره زمان-محدود را ایجاب می نماید. برای حمایت از زمان-واقعی، این میان افزار باید به طور عمودی به روشی مدولار از سطوح شبکه و سیستم عامل تا نرم افزار کاربردی و خط مشی های زمان واقعی برای دستیابی به رفتار محدود-در زمان معماری شود. این مقاله، یک میان افزار را ارائه می دهد که به این مشخصات حمایت کننده از پیکربندی دوباره به موقع در سیستم های زمان-واقعی توزیع شده بر اساس خدمات می پردازد. نتایج آزمایشی برای اعتبارسنجی میان افزار در یک نمونه اولیه ویدئوی مقیاس-کوچک واقعی نشان داده می شوند.

عبارت شاخص- سیستم های توزیع شده، میان افزار، سیستم های زمان-واقعی، پیکربندی دوباره، ترکیب خدمات،

SOA.

۱. مقدمه

محیط های محاسباتی کنونی در حال تبدیل شدن به زیرساخت های پیچیده ای هستند که شامل مجموعه های متعددی از گره ها با ماهیت ناهمگون می شوند که یک ابر از نقاط پردازش را تشکیل می دهند. این گره ها، حس، پردازش، تحریک و تولید اطلاعات را برای اپراتورهای انسانی و/یا برای دیگر گره ها و سیستم های فرعی انجام می دهند. اطلاعات ضبط شده در بخش های مختلف سیستم پیچیده می توانند پردازش شوند و در زمان واقعی به نقاط از

راه دور دیگر برای پردازش بیشتر آن تحویل داده شوند و نتایج مورد نیاز را به موقع تولید نمایند. برای نمونه، مورد یک سیستم نظارت ویدئویی پیچیده را با تحلیل محتوا برای نظارت بر فرآیندهای صنعتی در محیط های خصمانه در نظر بگیرید که هیچ اپراتور انسانی در آن وجود ندارد. در آینده نزدیک، انتظار می رود این زیرساخت ها، پردازش متقابل حجم زیادی از اطلاعات را در زمان واقعی انجام دهند.

برای میسر نمودن وجود محیط های محاسباتی پیچیده، یک رویکرد انعطاف پذیر برای طراحی، توسعه و قابلیت داخلی آنها مورد نیاز است. یک روش طبیعی برای این کار، میسر نمودن تعامل قسمت های فرعی به روشی تجزیه شده است. پیش بینی احتمالات محدودکننده دیگر (همانند توسعه یکنواخت) واقعاً با فناوری کنونی و در زمینه ای که احتمالات حوزه کاربرد و تقاضاهای کاربر به سرعت در بازار پدیدار می شوند سخت است.

معرفی راه حل های تعاملی گسسته، بخشی از مسئله را حل خواهد کرد، اما چالش های دیگر مرتبط با امنیت، پردازش اطلاعات توزیع شده، سنکرون سازی و دقت و غیره را به همراه خواهد آورد. به طور دقیق، عملیات زمان-واقعی چنین زیرساخت های پیچیده ای، یکی از چالش های کلیدی خواهد بود که باید از دیدگاه چندحوزه ای رفع و رجوع شود.

در حال حاضر، یک روند واضح، حمایت از پیکربندی دوباره سیستم به عنوان بخشی از دینامیک این محیط های پیچیده است. یک پیکربندی دوباره بر گذار بین دو حالت مختلف سیستم متکی است. تعدادی از رویدادها را می توان از طریق سیستمی تولید نمود که به واکنش به موقع برای رفع و رجوع وضعیت های خواسته یا ناخواسته نیاز دارد (کاربرد تعریف شده). زیرساخت (یعنی، گره ها و اتصالات آنها) باید اجرا و حتی لینک های گره را برای هدایت رویدادهای جدید تطبیق دهند و پاسخی صحیح به حالت جدید سیستم ارائه دهند.

با در نظر داشتن این مسئله چالش آور، ما دیدگاهی را نسبت به پیکربندی دوباره زمان-واقعی از نقطه نظر سطح سیستم توصیف می کنیم که معماری میان افزار توسعه یافته در پروژه iLAND ارائه می دهد؛ این یک راه حل مستقل از سیستم عامل برای حمایت از عملیات و پیکربندی دوباره زمان-محدود در سیستم های زمان-واقعی نرم توزیع شده خدمات-گراست. این معماری، مجموعه ای از مولفه ها و منطق مرتبط را برای فراهم نمودن تضمین های معین-زمانی تعریف می کند. راه حل های مدرن با الگوریتم ها و مکانیزم های قبلی برای حمایت از پیکربندی مجدد زمان-محدود ادغام نمی شوند. آنها: 1) بر تکنیک های قابلیت درست کردن خدمات در سطح برنامه تمرکز می کنند که در مورد

مشخصات زمان واقعی کاری انجام نمی دهند یا 2) زمانی که سیستم های زمان واقعی مد نظر هستند، آنها عمدتاً با زمانبندی پردازنده سطح-پایین بدون ملاحظه واقع گرایانه از موضوعات توزیع سرو کار دارند.

این مقاله با توصیف مسئله در بخش II آغاز می شود. بخش II، کار مرتبط، بخش IV، توصیف مفهوم پیکربندی زمان واقعی و مدل برنامه؛ بخش V، معماری iLAND، بخش VI اعتباردهی آن در یک برنامه نظارت ویدئویی صنعتی مقیاس-کوچک و بخش VII نتیجه گیری این مقاله را ارائه می دهند.

A. الزامات (شرایط)

در فاز اول پروژه iLAND، موضوعات ارائه شده بالا با توجه به سیستم های تعبیه شده شبکه ای پیچیده (NES) به طور گسترده مطالعه شده است و مجموعه ای از شرایط برای معماری نرم افزار و نقش گره های سخت افزاری مختلف تعیین شده است. راه حل های مرتبط نیز با جزئیات مورد تحلیل قرار گرفته اند. آگاهی متخصصان صنعتی در مورد استقرارات مختلف مقیاس کوچک، متوسط و بزرگ ترکیب شده اند و کنترل های متقابل با آنها برای خلق سریع نمونه های اولیه نیز کلید دستیابی موفقیت آمیز به توصیف و شرایط مسئله است. سیستم های هدف پروژه در انتها، تقریباً به مفهوم سیستم های فیزیکی سایبر (CPSS) مرتبط می شوند [1] که شامل ادوات و سیستم های پزشکی با اعتماد-بالا، کنترل و ایمنی ترافیک، سیستم های پیشرفته خودرو، کنترل فرآیند و حفاظت از انرژی و غیره می شود. این مطالعات توسط جمع آوری مجموعه ای از شرایط خاص به شرح زیر نتیجه گرفته شده اند:

- عدم تجانس: راه حل ها باید برای تمام سیستم عامل ها مناسب باشند، و در نتیجه باید قابل سفارشی سازی و تعمیم باشند.
- قابل کارکرد داخلی: سیستم ها با ماهیت های مختلف از طریق واسطه های تعریف شده و ساده تعامل خواهند داشت. در کنار اینها، بیشتر NES موجود شبکه بندی خواهد شد؛ پروتکل های مناسب، کارآمد در-داده ها و سبک وزن هستند و به طور خاص برای تسهیل قابلیت عملکرد داخلی طراحی می شوند.
- عملیات با زمان-محدود: ویژگی های زمان-واقعی در بیشتر محیط های کنونی، فقط را تفاوت در سطح بحرانی بودن موقتی خود حاضر هستند. برای برخی، دقیق بودن، تنها یک ارزش افزوده واضح است، در حالیکه برای دیگران، یک الزام اجباری است. در زمینه این پروژه، ما با الزامات زمان واقعی نرم سرو کار داریم.

- رفتار دینامیک: یک مسئله زمان-اجرای پیچیده را به همراه می آورد، زیرا به سرو کار داشتن با جایگزینی تابع، توازن بار و خود-ترمیمی سیستم و غیره سرو کار دارد.
- قابلیت شمول: قابلیت ارتقا، استفاده مجدد از مولفه ها، توسعه انعطاف پذیر و تدریجی، قابلیت حفظ و مقیاس پذیری در اکثریت توسعه های کنونی نیاز می شوند.
- انسجام: انسجام سیستم را می توان در سطوح سخت افزار/ نرم افزار انجام داد. طراحی مدولار، کلید دستیابی به انسجام، بر اساس مولفه ها، خدمات یا پروتکل های ارتباطی استاندارد است.

B. دیدگاه عمودی

- برای برآورده سازی الزامات ذکر شده بالا، یک رویکرد معماری عمودی نیاز می شود که نه تنها سیستم عامل های اجرا را در نظر می گیرد، بلکه پشته نرم افزار را در نظر می گیرد که توانمند در سرو کار داشتن با منطق برنامه و هوش اضافی لازم برای حمایت از پیکربندی دوباره دینامیک است. در این مورد، پشته نرم افزاری شامل لایه های مورد نیاز خواهد شد که هوش اضافی را در سطوح مختلف تعبیه می کنند:
- OS: موارد ابتدائی ارتقا یافته برای مدیریت اجزای قسمت های فردی، گروه های قسمت، و برای فراهم نمودن سنکرون سازی پایه و تسهیلات زمانبندی مورد نیاز خواهد بود.
 - مدیران منبع: به عنوان ارتقایی برای OS، مدیران منابع، الگوریتم هایی را برای زمانبندی منبع اجرا خواهند نمود؛ منابع برای نهادهای فعال منسوب خواهد شد (یعنی، برنامه ها، قسمت ها یا گروه های قسمت). این الگوریتم ها باید موثر و به صورت آنلاین عملی باشند.
 - میان افزار: مینیم لایه ارتباطی است که یک الگوی نرم افزاری توزیع شده کارآمد را برای برنامه نویسی انعطاف پذیر، حمایت از مشخصات ویژگی های اجرایی مبتنی بر QoS و مشخصات موقتی فراهم می کند.

C. الزامات زمانبندی و QoS

سطوح مختلفی از تضمین های موقتی وجود دارند. برخی از مولفه ها به مشخص نمودن یک مهلت برای انجام عملیات های خود نیاز دارند، به طوری که هر کار فردی باید در این مهلت انجام شود؛ در غیراینصورت، این سیستم موفق به کار

درست نمی شود. هرچند مولفه های دیگر نیز دارای محدودیت های زمانی هستند، اما عدم موفقیت در برخی عملیات ها برای برآورده سازی برخی مهلت ها (شاید حتی تحت تاثیر اثرات شبکه) به طور کامل، عملیاتی صحیح آنها را تهدید نمی کند. مثال هایی از چنین عملیاتی های مولفه می توانند آشکارسازی و واکنش به برخی آلام ها و نمایش اطلاعات پیش زمینه باشند.

عمل کنونی برای برآورده سازی تضمین های زمانی در قسمت های بحرانی، از واحدهای HW و SW اختصاص داده شده با یک تحلیل قابلیت زمانبندی گسترده استفاده می کند. هرچند، این روند، بیشتر و بیشتر برای اجرای تمام نرم افزار در سیستم عامل های HW/SW هدف-عمومی و شبکه ها بر اساس پروتکل های TCP/IP و اترنت است. ترکیب مولفه های نرم افزار با سطوح بحرانی مختلف در یک سیستم عامل، اجرای به موقع بیشتر موارد بحرانی را تهدید می کند. این به خصوص در صورتی مشهود است که همچنین اثرات شبکه و حوادث اترنت در نظر گرفته شوند. بنابراین برای تضمین دقت، راه حل های مناسب باید در محل قرار گیرند تا از ایزولاسیون زمانی اطمینان حاصل شود؛ برای این، مشخصات سیستم عامل، زمانبندی چند-منبع و معماری های میان افزار کارآمد که به تعامل به موقع در یک محیط شبکه بندی شده کمک می کنند، باید در نظر گرفته شوند.

D. کمک

در راه حل های کنونی، یکپارچه سازی سیستم عامل های نرم افزاری جدید مبتنی بر استفاده از پروتکل های ارتباط مستقیم یا میان افزار همه منظوره است. هرچند حمایت برای رفتار دینامیک، اجرای مبتنی بر QoS و قابلیت ترکیب برنامه آنلاین به طور مشخص و با روشی ترکیبی در هر راه حل به کار برده نشده است. هرچند، برخی از تکنیک های ترکیب برنامه آنلاین وجود دارند، آنها خارج از زمینه سیستم های زمان-واقعی به کار برده می شوند. در این کار، بعد قطع-زمانی برای زمان اجرای خدمات (شامل ارتباطات آنها) و برای مدت زمانی فرآیند پیکربندی دوباره به کار می رود. ما یک میان افزار را توصیف می کنیم که به الزامات ذکر شده بالا می پردازد که حمایت برای رفتار دینامیک را با تضمین های QoS در زمینه یک الگوی نرم افزاری انعطاف پذیر در قابلیت ترکیب برنامه های توزیع شده زمان واقعی

نرم فراهم می کند. بنابراین، پیکربندی دوباره زمان-قطعی از یک دیدگاهی منسجم توسط فراهم نمودن موارد زیر مدیریت می شود.

- یک طراحی معماری عمودی برای یک میان افزار که کاربردهای مبتنی بر SOA توزیع شده را فعال می سازد و از موارد زیر حمایت می کند: 1) اجرای زمان-واقعی و 2) تبادل پیام زمان واقعی (یا مبتنی بر QoS) در میان خدمات و گره ها.
- یک معماری نرم افزار، نقاط خاص برای حمایت از اجرای زمان واقعی را نشان می دهد. این معماری از نظر دیدگاه های مولفه، ساختاری و رفتاری بیان می شود.
- یک پروتکل پیکر بندی دوباره و یک الگوریتم ترکیب مبتنی بر سرویس است که تنظیمات سیستم جدید را در زمان محدود و با خطای محدود با توجه به معیار های مشخص شده کاربر / سیستم کسب می کند. محل این مکانیزم ها در داخل میان افزار در مولفه، دیدگاه ها، و نمودار معماری توصیف می شود.

III. کار مرتبط

محبوب ترین حوزه های کاربردی سیستم های توزیع شده، به شدت به زمان واقعی حساس نیستند و بنابراین، بسیاری از کارهای تحقیقاتی جنبه های زمانی را در پویایی خود و یا آن را در ابعاد مختلف مورد نیاز به اندازه کافی در نظر نمی گیرند (یعنی، شبکه، سیستم عامل، میان افزار ارتباطات و الگوی برنامه نویسی سطح برنامه). همچنین این مورد درست است که کار شدید در سطح شبکه می تواند در بهره وری بیشتر و کنترل زمانی در اجرای سیستم شبکه بندی در زمان واقعی منجر شود؛ با این حال، راه حل های مبتنی بر میان افزار به تدریج با سبک سنگین کردن برخی از کارایی ها و قابلیت پیش بینی برای انعطاف پذیری و قابلیت برنامه ریزی، در حال به دست آوردن محبوبیت هستند. بنابراین، برخی تلاش های تحقیقاتی از جزئیات شبکه در سطح پایین تا طراحی و توسعه میان افزار در حوزه های مختلف چند رسانه ای [2] و یا [3 CPS] منتقل شده اند.

در سطح شبکه، پیکر بندی دوباره زمان واقعی ترافیک برای سیستم های مبتنی بر سرویس تا حدی به کار گرفته شده است که در [4]، که در آن ترکیب سرویس که در یک نهاد اصلی مرکزی در داخل یک شبکه تحریک شده-زمانی انجام می شود. در اینجا، پیکربندی های جدید، هر چرخه ابتدایی را با استفاده از یک پیام تحریک برای حمایت از یک سطح پایه پیکر بندی دوباره ارسال می کنند.

از جانب میان افزار ارتباطات، تعدادی از راه حل ها با سطوح پذیرش مجزا، به عنوان مثال ، CORBA سابق محبوب [5] (و RT-CORBA) وجود دارند که ارتباطات توزیع شده را مبتنی بر واسطه های خوش تعریف ارائه نموده اند؛ آنها در حال حاضر عمدتاً به دلیل معماری پیچیده و سربارهای اجرایی مرتبط با آنها رها شده اند، به خصوص در مقایسه با فناوری های سبک وزن تر که موقعیتی را در بازار دارند، مانند موتور ارتباطات اینترنتی (Ice) [6]. خدمات توزیع داده ها (DDS) برای سیستم های زمان واقعی [7] یا راه حل های وابسته به زبان [8] برای زبان جاوای زمان واقعی.

روند فعلی در مهندسی سیستم های توزیع شده عمدتاً مبتنی بر الگوی معماری های سرویس گرا (SOA) و بر راه حل های میان افزار ارتقا یافته است که از زبان های تبادل مشترک به عنوان XML استفاده می کنند. اکثر سیستم های SOA و میان افزار مرتبط با آن، محیط های وب را هدف قرار می دهند مانند [10]، [11]؛ همچنین، پژوهش در مورد سیستم های مبتنی بر SOA پویا بر ترکیب سرویس های وب بر اساس چک کردن های متقابل هستی شناسی و وضوح وابستگی [12]، تمرکز دارد. جنبه زمانی معمولاً یک نگرانی در محیط های سرویس وب است که بیشتر در [13] نشان داده شده است و آنها در مورد پیدا کردن یک راه حل در زمان محدود ضعیف هستند. سنجش متقابل QoS اعمال شده در تکنیک های ترکیب سرویس مبتنی بر وب عمدتاً به درجات مربوط به نرم افزار و مفاهیم داده [14] محدود می شود. در طول چند سال، الگوهای SOA برای سیستم های تعبیه شده منبع-محدود به کار برده شده اند و اخیراً جامعه تعبیه شده زمان-واقعی نیز، معماری هایی را برای متمایز نمودن قسمت های زمان واقعی از غیر زمان واقعی در سیستم توزیع شده مبتنی بر خدمات تدبیر نموده است [16]. بنابراین، برخی از کمک ها در سطوح مختلف به عنوان خواص QoS برای اجرای خدمات و تعامل بین خدمات [17]، پشتیبانی میان افزار برای برنامه های کاربردی مبتنی بر خدمات و نظریه الگوریتم های ترکیب [18] ظاهر می شوند.

اجرای پویا نیز با استفاده از ماشین های مجازی مورد بررسی قرار گرفته است که، در میان امکانات دیگر، اجرای خدمات شفاف و کد قابل حمل را ارائه می دهد. این محیط ها معمولاً لایه های نرم افزاری ضخیم هستند که نمونه ای از سیستم عامل واقعی هستند. با این حال، خدمات در زمان واقعی به یک سیستم اجرای قطعی و یکپارچه سازی یک طرح برنامه ریزی مناسب همانند [19]، [20] نیاز دارد. ویژگی های پیشرفته همانند حرکت خدمات و دانلود کد هنوز

هم یک تهدید برای قابلیت پیش بینی است و دیگر معماری های موجود مانند [21] OSGi (و برخی از تلاش ها برای یکپارچه سازی به موقع [22]) هنوز هم به دور از زمان واقعی هستند.

برخی از کمک ها از جانب مهندسی نرم افزار، میان افزارهای قابل پیکر بندی مجدد را با مطالعه جنبه های منعکس کننده در حمایت از قابلیت همکاری، کشف، و تحرک از طریق شبکه های پوشش نردبان [23] ساخته اند. اگر چه میان افزارهای ما را می توان در نسخه های مختلف مستقر نمود، پیکربندی آنها باید آفلاین باشد. بنابراین، تمرکز ما متفاوت است، زیرا میان افزار خود را در زمان اجرا پیکربندی مجدد نمی کند بلکه از پیکر بندی دوباره به موقع برنامه های کاربردی پشتیبانی می کند.

به طور خلاصه، تفاوت بین راه حل های موجود و میان افزار iLAND دو قسم است: (1) ما به صراحت رفتار موقتی سیستم های توزیع شده را در نظر می گیریم، و (2) ما دستیابی به پیکر بندی دوباره زمان محدود برای حمایت از تغییرات کاربردی برای تغییر ساختار سیستم توزیع شده را در زمان اجرا در زمان محدود هدف قرار می دهیم. در حال حاضر، هیچ راه حلی وجود ندارد که حمایت از ارتباط زمان واقعی و پیکر بندی دوباره را از طریق ترکیب مبتنی بر خدمات برای رسیدن به پیکر بندی دوباره زمان-محدود را ادغام نماید. در این مقاله یک جهت گیری دقیق توسط معماری یک میان افزار مبتنی بر مولفه های QoS-آگاه ارائه شده است که از اجرا و ارتباطات برنامه های کاربردی مبتنی بر خدمات و پیکربندی دوباره آنها در زمان واقعی پشتیبانی می کند. اولین مرحله از طراحی و معماری میان افزار در [24] و مدل برنامه در [25] و [26] معرفی شد.

IV. مدل سیستم

A. مدل پیکربندی دوباره و ویژگی های زمان واقعی

میان افزار پیشنهادی، سیستم های زمان واقعی نرم توزیع شده را هدف قرار می دهد که در آن عدم تحقق برخی از مهلت ها، مشکل ساز نیست و عملکرد قابل قبولی را هم می توان به دست آورد؛ ممکن است این اتفاق رخ دهد که تنها یک خروجی تنزل یافته تحویل داده شود، اما این نتیجه هنوز هم ممکن است درست باشد و از یک مطلوبیت خاص برخوردار باشد. در نتیجه، سیستم های بلادرنگ نرم مفهوم QoS را استفاده می کنند که به توانایی موازنه منابع مورد

استفاده به واسطه وظایف برای کیفیت تحویل مربوط می شود. در مدل ما، ویژگی های QoS, [25] بسته به ارتباط آنها با: 1) منابع فیزیکی و 2) قابلیت کاربرد در سطح، دو بعد دارند.

دینامیک، عدم قطعیت اجرای به طور عمده مربوط به هر دو تنوع کاربردی و یا منابع را معرفی می کند. تنوع کاربردی به جایگزینی برخی از فعالیت های خودشامل (یا واحد کاربردی) به واسطه نتایج متمایز احتمالاً ارائه شده اشاره می کند. تنوع منابع به انتساب مختلف ممکن منابع به واحدهای عملیاتی موجود مختلف اشاره دارد. هر دو نوع تنوع درون-ارتباطی می باشند. برای مثال، یک جایگزینی یک واحد فعال ویدئو توسط یک واحد صوتی به یک تنوع در انتساب منبع منجر خواهد شد زیرا پردازش صوتی، به طور متوسط، منابع کمتری را نسبت به پردازش ویدئویی مصرف می کند. با این حال، تغییر در تخصیص منابع همیشه نتیجه تغییر کاربردی نیست؛ می توان آن را یک تصمیم داخلی نظارت در پاسخ به، برای مثال، یک تجدید محاسبه توازن بار در نظر گرفت.

در این زمینه، یک پیکر بندی دوباره سیستم در هر زمان که یک تنوع عملکردی و یا یک منبع وجود دارد رخ می دهد که نشان دهنده تغییر هر دو واحد عملکردی و یا انتساب منابع آنهاست. در هر صورت، این نشاندهنده یک انتقال از حالت یک سیستم (SS_{init}) به حالت هدف (SS_{target}) است. بنابراین، یک پیکر بندی دوباره، بخشی از رفتار دینامیکی یک سیستم است. در نتیجه، پیکر بندی دوباره به موقع در هر زمان که گذار از ss_{init} به $sstarget$ انجام شده کمتر از زمان مشخص شده به tre به صورت زیر است، به دست می آید

$$f(SS_{init}, SS_{target}) < tre.$$

هدف از تکنیک های پیکر بندی دوباره در زمان واقعی توسعه مکانیسم و معماری به منظور توسعه یک چارچوب مناسب که در آن تابع می تواند زمان محدود شده است. از نقطه نظر نرم افزار-معمار، f به تعداد سطوحی بستگی دارد که باید به دقت در نظر گرفته شوند.

• سخت افزار: اثرات پردازش آموزش، حافظه نهان، و معماری پردازنده به طور مستقیم، به موقع بودن اجرا و قابلیت پیش بینی را تحت تاثیر قرار می دهند.

- سیستم عامل: یک سیستم عامل در زمان واقعی دارای مقدمات زمان-محدود است، بنابراین مدیران منابع بسیار کارآمد عمدتاً در داخل هسته برای ارائه مکانیسم های برنامه ریزی منابع افزایش یافته اجرا می شوند.
- میان افزار: ارتباطات انتزاعی که عدم تجانس گره و اثرات شبکه بندی را پنهان می کند، دارای پیامدهایی در پیش بینی سیستم است.

B. مدل نرم افزار

یک مدل سرویس گرا به عنوان الگوی نرم افزار انتخاب می شود، زیرا انعطاف پذیری طراحی و اجرا را افزایش می دهد. خدمات، مولفه کارکردی خود شامل هستند که ساخت برنامه های توزیع شده را به روش جدا شده میسر می سازند، یعنی، خدمات در گره های راه دور در شبکه قرار می گیرند و از طریق پیام و یا حوادث ارتباط برقرار می نمایند. در این زمینه، یک سرویس، یک کانال داده را مدیریت می کند همانطور که در شکل 1 نشان داده شده است؛ این سرویس، داده ها را از طریق برخی از واسطه های ورودی دریافت می کند، داده ها را پردازش می کند، و یک نتیجه را تولید می کند که می تواند تولید از طریق واسطه خروجی آن به سایر خدمات تحویل داده شود.

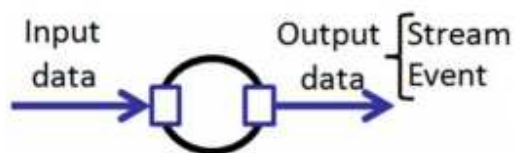
ساخت قابلیت های توسعه یافته (برنامه) توسط اتصال خدمات ممکن است. برنامه های کاربردی مبتنی بر سرویس، مجموعه ای از خدمات متصل در قالب یک نمودار هستند که در آن گره ها، خدمات، و خطوط اتصال دهنده یا فلش ها، پیام های رد و بدل شده بین آنها هستند. اتصالات خدمات در داخل یک برنامه می تواند وابستگی هایی داشته باشد که می توانند عملکردی (نوع داده های پردازش شده توسط دو خدمات متصل باید سازگار باشد) و غیر کارکردی (به عنوان مثال، به موقع بودن اجرا و شرایط منابع) باشند. در این مدل، یک سرویس توسط قابلیت ارائه داده شده آن مشخص می شود. یک سرویس (که یک مقوله وظیفه-محور مفهومی است) توسط یک یا چند پیاده سازی سرویس محقق می شود که مقوله های واقعی در حال اجرا می باشند. اجرای یک خدمت، یک نسخه خاص از یک سرویس خاص است [شکل. 2 (الف) و (ب) را ببینید].

بنابراین، برنامه در حال اجرا مجموعه ای از پیاده سازی هاس سرویس است، و در نتیجه، یک نمودار اجرا تنها شامل پیاده سازی سرویس در اجرا می شود. ما سه نوع نمودار را تعریف می کنیم:

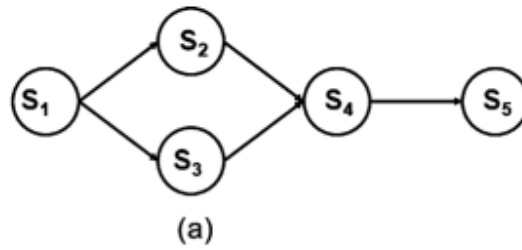
نمودار نرم افزار (برنامه) (AG) نمودار است که فقط شامل خدمات می باشد. این دیدگاهی ایستا از یک نرم افزار مورد نظر است. که در آن مجموعه ای از خدمات از نرم افزار وجود دارد. $AG^a = \{S, R, Q\}$ مجموعه ای از روابط (فلش) بین گره هاست و شامل عناصر از نوع $S_j \rightarrow S_k$ می شود که در آن خدمات S_j به S_k در نمودار متصل می شود؛ Q مجموعه ای از کیفیت پارامترهای خدمات مربوط به پردازش داده ها نرم افزاری خاص و مرتبط با نیازها برای منابع است.

نمودار گسترده (XG) شامل تمام پیاده سازی های سرویس امکان پذیر می شود. $XG^a = \{SI^a, R', Q\}$ که در آن SI^a مجموعه ای از پیاده سازی برنامه خدمات است؛ $a; SI_i$ مجموعه ای از پیاده سازی های خدمات S_i ($SI_i = \{s_i^1, \dots, s_i^n\}$) است. R' مجموعه ای از روابط (فلش) بین گره هاست و شامل عناصر از نوع $S_j^a \rightarrow S_k^d$ می شود که در آن پیاده سازی سرویس S_j^a به S_k^d در نمودار متصل می شود. Q مجموعه ای از کیفیت پارامترهای خدمات از نمودار کاربرد آن است.

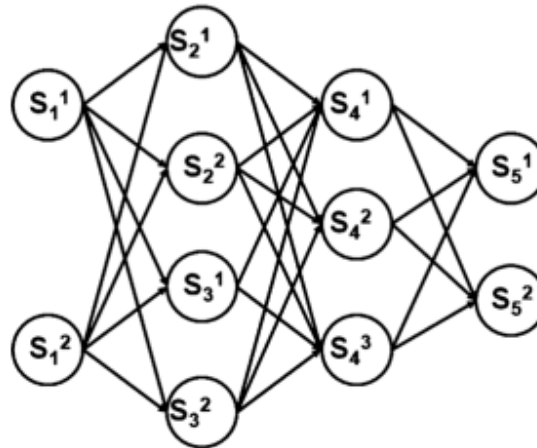
نمودار اجرا (EG) برنامه ی در اجرا است؛ که شامل پیاده سازی سرویس در حال اجرا می شود $EG^a = \{SI^a, R'', Q\}$ ، که در آن مجموع پیاده سازی های خدمات گنجانده شده در SI^a تنها پیاده سازی هایست که توسط منطق ترکیب انتخاب شده اند (که بعداً در بخش V توضیح داده می شود)؛ R'' زیرمجموعه ای از R' است و Q دارای ساختار یکسان برای XG^a است.



شکل 1: کانال داده خدمات



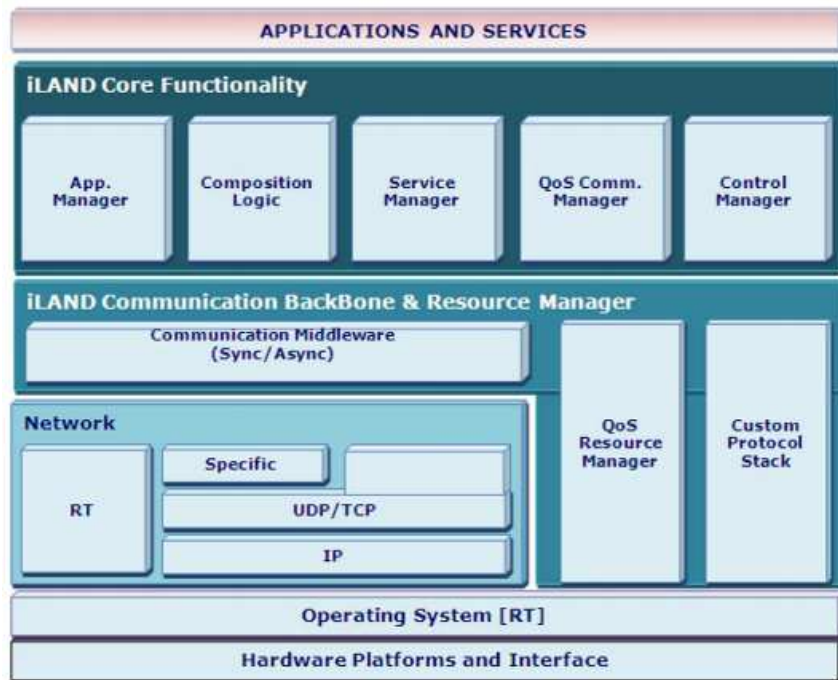
(a)



(b)

شکل 2 - (a) برنامه کاربردی به عنوان یک ساختار گراف (AG یا گراف برنامه). (ب) نمودار گسترده (XG) حاوی پیاده

سازی سرویس های مختلف هر یک از خدمات از AG.



شکل 3. معماری میان افزار iLAND

رویکرد مبتنی بر خدمات مناسب برای رفتار سیستم پویا است (به عنوان مثال، جایی که در آن پیکر بندی دوباره باید توسط اصلاح نمودار اجرا رخ دهد). بنابراین یک پیکر بندی دوباره، انتقال از یک نمودار اجرا به دیگری است. محرک های پیکر بندی دوباره می توانند متنوع باشند: 1) عمل کاربر به عنوان یک درخواست برای تغییر از یک قطعه قابلیت در نرم افزار یا درخواست صریح و روشن برای تغییر در کیفیت خروجی تحویل داده شده توسط برنامه. 2) تصمیم سیستم داخلی معمول شناسایی شده از طریق نظارت بر سیستم. این مورد زمانی می تواند رخ دهد که بار سیستم باید متعادل باشد و مکانیزم های مورد نیاز برای این منظور راه اندازی شوند. و 3) تصمیم برنامه با توجه به برخی شرایط نرم افزار خاص و یا برنامه ریزی رویداد. این محرک منجر به دو نوع پیکر بندی دوباره می شود: 1) کاربردی، که در آن یک سرویس باید آغاز، متوقف، و یا جایگزین شود و 2) داخلی، که در آن یک پیاده سازی سرویس جایگزین می شود.

۷. میان افزار برای پیکر بندی دوباره زمان واقعی

برای رسیدن به پیکر بندی دوباره زمان واقعی در iLAND، مطالعه ای از سیستم باید در ارتباط با تجزیه و تحلیل سیستم انجام شود. یک مطالعه قبلی از سیستم، اجرای غیر قطعی و پیچیده ساختار نمودار که به یک انفجار ترکیبی از مجموعه حالت سیستم منجر خواهد شد را حذف می کند. در سیستم های زمان واقعی، رسیدن به پیکر بندی دوباره زمان-محدود در غیاب کامل دانش در مورد ساختار سیستم، نیاز زمانی آن، و ویژگی های بخش های تشکیل دهنده آن ممکن نیست. برای این دلیل، iLAND مراحل مختلف را مشخص نموده است.

• فاز اولیه متشکل از مشخصات سیستم از طریق ایجاد ساختار نرم افزار، مجموعه خدمات آنها، پیاده سازی سرویس هر یک از خدمات، و غیره است. در این فاز، مطالعه آفلاین سیستم و خواص زمان آن (نمودار و هزینه زمانی از انتقال) انجام می شود. این مرحله، آگاهی در مورد رفتار زمان-اجرا آینده و امکانات سیستم را به ارمغان می آورد.

• فاز عملیات. پس از عبور از یک مرحله اولیه، سیستم آماده برای اجرا است. در این مرحله، ترکیبی

از خدمات و اجرای آنها رخ می دهد.

در فاز اولیه، تنظیمات پیش فرض توسط میان افزار ذخیره شده است؛ این تنظیمات پیش فرض، یک حالت سیستم است که دارای حداقل کیفیت قابل قبول است و در نتیجه، ثابت می شود که همیشه به صورت آفلاین قابل زمانبندی

می باشد. هدف از این تنظیمات پیش فرض، داشتن یک حالت سیستم است که در صورتی می تواند اجرا شود که شکاف زمان پیکر بندی دوباره توسط میان افزار در برخی از نقاط اجرا برآورده نشود.

A. معماری

معماری میان افزار iLAND از اصول کلاسیک میان افزار لایه لایه با اضافه کردن تعدادی از برنامه های افزودنی برای موارد خاص پیروی می کند:

- پشتیبانی از برنامه های کاربردی سرویس گرا؛
- ادغام تکنیک های پیکر بندی دوباره زمان قطعی و الگوریتم های خدمات ترکیب؛
- ارتباطات زمان واقعی با تعریف پشته پروتکل شبکه کامل (برای مثال، کنترل دسترسی سطح -2- زمانی برای فعال کردن تجزیه و تحلیل قابلیت زمان بندی)؛
- قابلیت حمل به قفسه های مختلف ارتباطات خارج از ستون فقرات میان افزار از جمله DDS یا Ice، و همچنین یکپارچه سازی با DSA [9]؛ این مورد توسط تعریف یک پل ارتباطی مشترک برای مدل تعامل همزمان و ناهمزمان به دست می آید.

ویژگی های ذکر شده در بالا در شکل 3 منعکس شده است که کلیدی از معماری میان افزار را ارائه می نماید.

لایه کارکرد هسته (CFL) شامل بسیاری از قابلیت های ارزش اضافه شده کلیدی مربوط به مدیریت برنامه و کنترل پیکر بندی دوباره می شود. اجزای آن در اینجا داده شده است.

• مدیر خدمات (SM) حاوی شکل های هندسی اولیه برای اعلام، حذف، و تغییر در خواص (از جمله منابع مورد نیاز آ»ها) از خدمات منحصر به فرد و پیاده سازی خاص آنهاست.

• مولفه مدیر برنامه (AM) شامل منطق برای تعریف ساختار یک برنامه کاربردی (نمودار خدماتی)، به عنوان مثال، خدمات و ارتباط آنها می شود. نرم افزار سطح پارامترهای QoS (خواص به عنوان مثال، پایان به پایان برای تمام نمودار خدماتی) با استفاده از این بخش مشخص شده است.

• منطق ترکیب (CL) شامل منطق برای ترکیب مبتنی بر خدمات متشکل از پیدا کردن مجموعه ای از خدمات برای ساخت یک برنامه مشخص مطابق با پارامترهای QoS مشخص شده می شود. نمونه هایی از معیارهای ترکیب، به حداقل رساندن مصرف حافظه در تمام خدمات و یا به حداقل رساندن / جلسه مهلت پایان به پایان هستند. از آنجا که هر خدمات می تواند نسخه های مختلف داشته باشد که آن را تحقق بخشد، یک نمودار کامل برای جستجوی یک راه حل معتبر می تواند دارای پیچیدگی بالا باشد؛ این مورد در iLAND با تعریف فازهای مختلف در طراحی، توسعه، و اجرای سیستم حذف شده است. الگوریتم ها باید کارآمد و محدودشده-زمانی باشند، زیرا آنها به صورت آنلاین اجرا می شوند.

• جزء مدیر کنترل (CM) شامل منطق هماهنگی پیکر بندی دوباره سیستم می شود؛ که شامل محرک های پیکر بندی دوباره هستند و فرآیندی را آغاز می کند که دارای مراحل مختلف است. مدت زمان هر مرحله است نیز برای انجام یک انتقال زمان-محدود کنترل می شود.

• مولفه مدیر ارتباطات QoS (QS)، پیکربندی پارامترهای ارتباطی QoS جهانی را صورت می دهد.

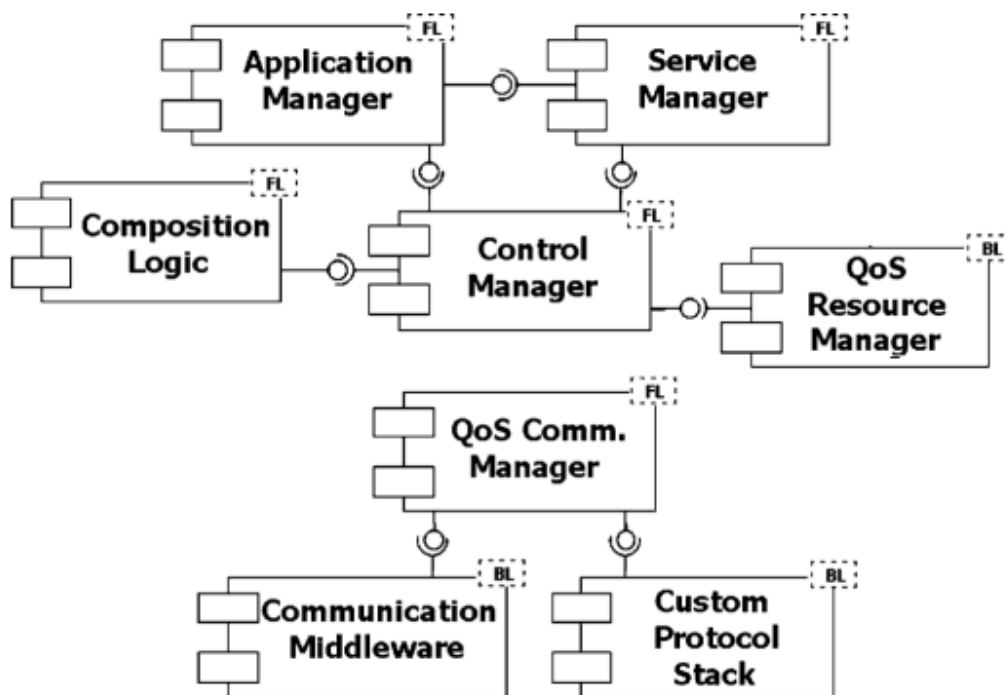
لایه ستون فقرات ارتباطات و مدیریت منابع (CBL) شامل اجزای زیر می شود.

• میان افزار ارتباطات هسته ستون فقرات (CC) یا مدل ارتباطی هسته. این اجازه می دهد تا از هر یک از تعاملات میان افزار پایه استفاده نماییم و از آن بهره مند شویم: P / S (SUB / pub)، MOM (میان افزار پیام محور)، DOM (میان افزار شی توزیع شده)، و یا RPC (فراخوانی روال از راه دور).

• مدیر منابع QoS (QoSRM) که هدف آن، مدیریت منبع مبتنی بر QoS است. برنامه ریزی برای مدیریت چند منبع در داخل این بخش نهفته است. از این شکل‌های هندسی اولیه RTOS (به عنوان مثال، زمانبندی پیش‌دستانه مبتنی بر اولویت و دسترسی به تایمرها با وضوح بالاتر) برای اجرای تکنیک های برنامه ریزی ارائه دهنده پاسخگویی استفاده از منابع عمومی، ایزولاسیون زمانی و تغییر حالت پروتکل ها استفاده می کند.

• پشته پروتکل سفارشی (CPPS)، مدیریت ترافیک شبکه را فراهم می کند. این اجازه می دهد تا سازگاری با سیستم های قبلی ارائه شود و همچنین برنامه های کاربردی با شرایط زمان واقعی سخت از جمله پروتکل های برنامه ریزی شبکه زمان واقعی حمایت شوند.

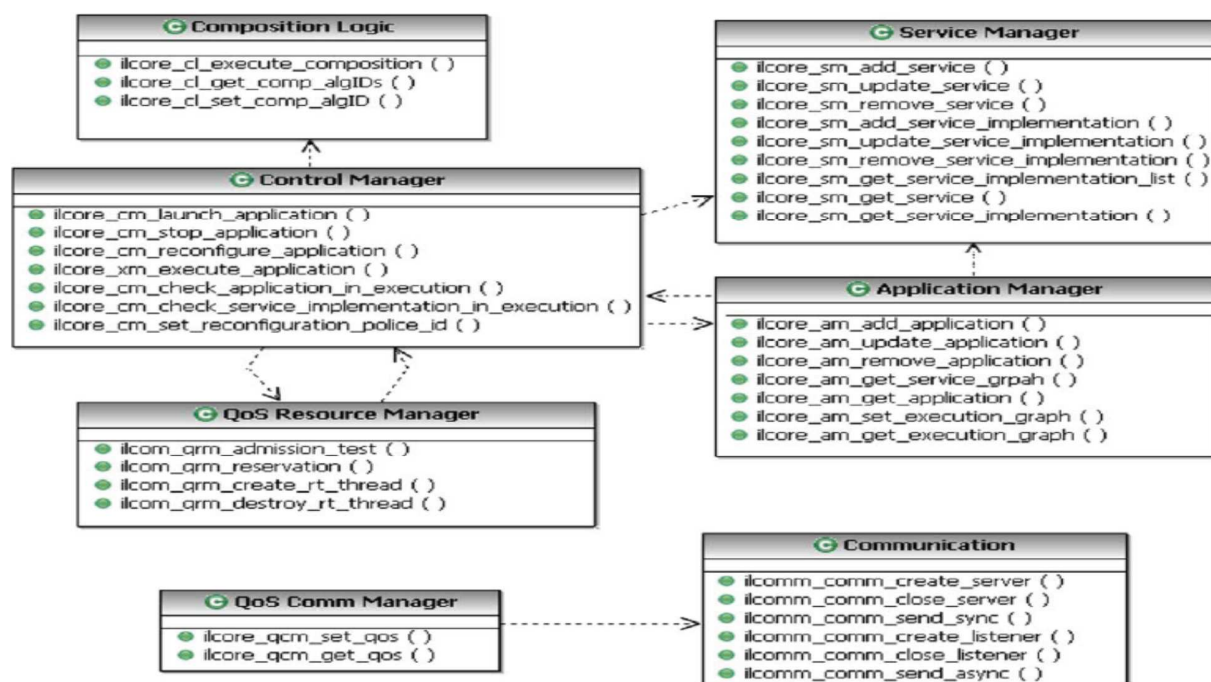
لایه شبکه (NL) شامل قابلیت پایه برای انتقال زمان واقعی در شبکه های عمومی می شود که / UDP TCP را روی IP ارائه می دهد. یک پروتکل ارتباطی خاص-نرم افزار می تواند در ماژول خاص مورد استفاده قرار گیرد، از جمله ارتباطات جریان با پروتکل های تخصصی به صورت RTP ارتقایافته با ارتباطات مبتنی بر RTCP در سطح حمل و نقل.



شکل 4. نمودار اجزاء.

B. دیدگاه های مولفه و ساختاری دیدگاه جزئی از iLAND در شکل 4 ارائه شده است که نشان دهنده ماژول مربوطه از میان افزار به عنوان اجزای تشکیل دهنده است.

این دیدگاه توسط دیدگاه ساختاری (در UML) مشخص شده در شکل 5 تکمیل می شود که یک نمودار کلاس نشان دهنده قابلیت اجزای iLAND و واسطه های آنهاست. می توان دید که اجزای SM و AM، مشخص نمودن شکل 5، خدمات، پیاده سازی سرویس، و برنامه های کاربردی، از جمله خصوصیات QoS خود را میسر می سازند.



شکل 5. نمودار کلاس ساده شده.

یکی از مولفه های کلیدی میان افزار، CM است که پیکر بندی دوباره را هماهنگ می کند. جزء CM با مولفه های SM و AM برای به دست آوردن اطلاعات در مورد خدمات، پیاده سازی های سرویس، و خود نرم افزار تعامل پیدا می کند. این با جزء CL برای اجرای الگوریتم ترکیب و با QOSRM برای اجرای آزمون کنترل پذیرش در تعیین در دسترس بودن منابع و قابلیت زمانبندی زمانی نرم افزار ترکیب می شود. در صورتی که یک راه حل عملی وجود نداشته باشد (آزمون پذیرش قبول نمی شود)، میان افزار ممکن است دوباره بر سر کیفیت پایین تر برای نرم افزار مذاکره مجدد نماید و یا کیفیت دیگر برنامه های در حال اجرا و در حال نصب پیکربندی پیش فرض را کاهش دهد.

هنگامی که برنامه، انجام بیدرنگ شرایط مورد نیاز برنامه را صورت می دهد، جزء CM باید اجرای برنامه را توسط راه اندازی پیاده سازی های خدمات شروع نماید و در نتیجه جریان اجرای برنامه شروع می شود. 5

اجرا توسط جزء QOSRM تحت نظارت قرار می گیرد؛ همچنین بودجه های منبع اختصاص داده شده به پیاده سازی خدمات را با توجه به خط مشی برنامه ریزی منبع و تضمین ایزولاسیون زمانی اجرا می کند. QOSRM دارای یک

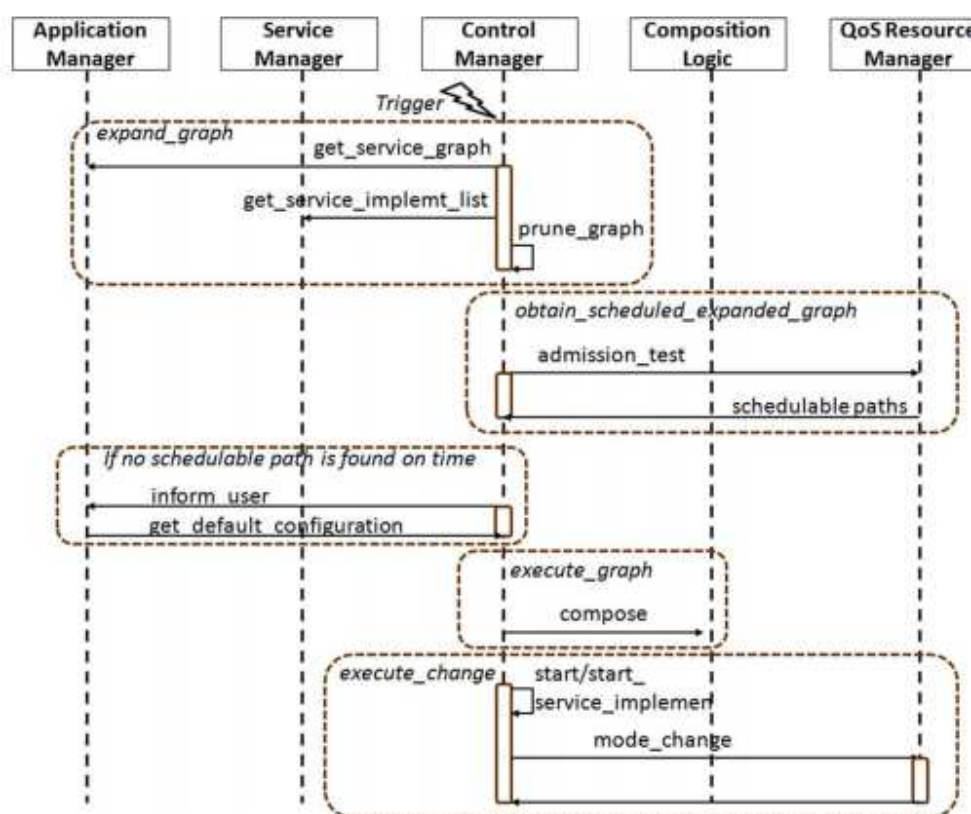
بخش متمرکز و توزیع شده است. نظارت محلی بر مصرف منابع در هر گره انجام می شود. یکی از گره ها به عنوان پایه عمل می کند دارای نمای کلی از وضعیت سیستم است.

در طول اجرای برنامه، پیاده سازی های سرویس با یکدیگر یا با استفاده از این موارد برقراری ارتباط می نمایند: (1) CC توابع جزء زیرا استفاده از ویژگی های ارتباطی QoS از میان افزار ستون فقرات اساسی، به عنوان مثال، موارد ارائه شده توسط DDS، یا (2) جزء CPPS که استفاده از یک پروتکل شبکه در زمان واقعی سفارشی را میسر می سازد. ویژگی های ارتباطی با توابع ارائه شده توسط جزء QS تنظیم می شوند.

وابسته به قدرت محاسبه گره ها و نقشی که آنها در سیستم بازی می کنند، آنها حاوی یک نسخه متفاوت از معماری هستند: ستاره، سیاره، و یا ماهواره ای. یک گره ستاره شامل منطق ترکیب (گره پایه) می شود، این سیاره شامل تمام کد iLAND به جز برای ترکیب و قسمت های کنترل پیکر بندی دوباره می شود. گره های ماهواره ای تنها شامل قابلیت های اساسی به عنوان قطعات دارای تعامل در شبکه iLAND می شوند. گره های ماهواره، سبکترین وزن نسخه از میان افزار را اجرا می کنند، زیرا برخی از اجزا وجود ندارد.

C. پیکربندی دوباره زمان محدود

در اینجا، ما قابلیت خاص مولفه های میان افزار و اثر متقابل آنها برای رسیدن به پیکر بندی دوباره- زمان محدود توسط مدیر CM (مدیر کنترل لایه CBL) را توصیف می کنیم. CM، محرک های پیکر بندی دوباره را آشکار می کند که یک درخواست کاربر، یک رویداد منطق برنامه، و یا یک وضعیت زمان اجرا از حالت سیستم هستند. دومی توسط QOSRM آشکار می شود که اجرای پیاده سازی خدمات را تحت نظارت قرار می دهد. شرایط نیازمند پیکر بندی دوباره می تواند خرابی یک گره یا یک پیاده سازی سرویس، سرریز بودجه منابع اختصاص یافته برای برخی از پیاده سازی خدمات، و یا تاخت و تاز کردن مکرر در تلاش یک پیاده سازی سرویس داده شده باشد. پروتکل پیکر بندی دوباره توسط CM هماهنگ می شود (شکل 6)، و شامل تنظیم مجدد داخلی تکالیف منابع برای مقوله های در حال اجرا می شود. اگر وضعیت شناسایی توسط QOSRM قابل به کار گیری نباشد (به عنوان مثال، هیچ نمودار اجرای قابل زمانبندی وجود ندارد)، علت آن در یک مطالعه قبلی از سیستم در مقدار دهی اولیه فاز یا توسط یک وضعیت اجرای استثنایی بیان شده است.



شکل 6: تعاملات کامپوننت در طول مجدد تنظیم.

هنگامی که پیکر بندی دوباره تحریک می شود، جزء CM، نمودار نرم افزار (AG) را گسترش می هد. برای انجام این کار، CM، محدودیت های معرفی شده توسط پارامترهای QoS را چک می کند؛ به تبع آن، همه پیاده سازی سرویس موجود بخشی از XG نهایی هستند. بنابراین XG برای شمول پیاده سازی های سرویس که از چک کردن های متقابل سازگاری گذشته اند هرس می شود. سپس CM، پروتکل پذیرش برای همه مسیرهای ممکن XG را آغاز می کند. فقط آن دسته از مسیرهایی قابل زمانبندی هستند از XG استخراج می شوند.

QOSRM شامل الگوریتم هایی می شوند که قابلیت زمانبندی مجموعه ای از پیاده سازی های سرویس را با استفاده از نظریه زمان واقعی (تجزیه و تحلیل استفاده و یا بر اساس زمان پاسخ) تعیین می کنند. XG بیشتر هرس خواهد شد تا فقط مسیرهای قابل زمانبندی کنار گذاشته شود و منجر به یک گراف گسترش یافته قابل زمانبندی یا گراف قابل زمانبندی می شود. شایان ذکر است که برای فرآیند پیکر بندی دوباره باید زمان محدود باشد و مدت زمان تعامل بین CM و QOSRM برای به دست آوردن نمودار قابل زمانبندی باید محدود شود. فاز مقداردهی اولیه تضمین می کند که

XG شامل مسیرهای قابل زمانبندی می شود که اجازه می دهد، در بدترین حالت و برای شیارهای پیکر بندی، حداقل یک مسیر در اولین دور انتخاب شود.

سپس نمودار زمانبندی شده به الگوریتم ترکیب در مولفه CL تغذیه می شود. در نتیجه، یکی از مسیرهای زمانبندی شده انتخاب می شود که شامل گراف اجرا (EG) می شود، یعنی، برنامه ای که باید اجرا شود. تولید گراف اجرا از معیارهای QoS (پارامتر Q) پیروی می کند که زمان پاسخ انتها به انتهاست. این پارامتر، عنصر مرکزی برای تمام حوزه های کاربرد اعتباردهی iLAND است. هرچند، در کاربردهای زمان-واقعی، یک رویکرد پذیرفته شده گسترده، استفاده از مقادیر زمانی است (یعنی، مهلت انتها به انتها). QoS RM، راه اندازی EG را از طریق یک تغییر حالت هماهنگ خواهد کرد. اگر هیچ راه حل قابل زمانبندی یافت شده درون شیار قابل پیکربندی وجود نداشته باشد، سیستم، پیکربندی پیش فرض ایجاد شده در فاز مقداردهی اولیه را تنظیم می کند.

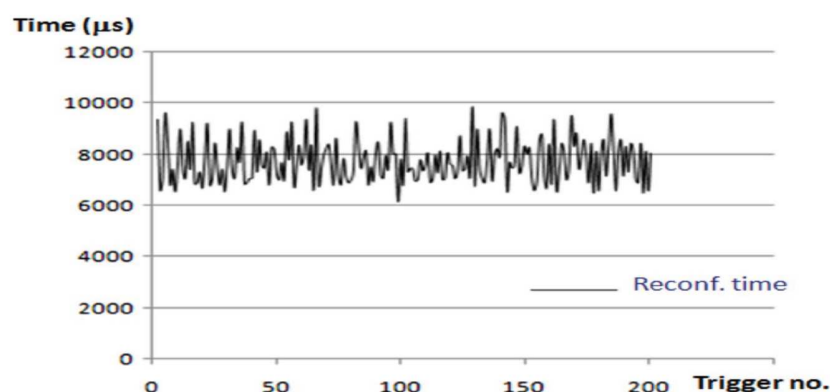
vi. اعتباردهی آزمایشی

اعتباردهی معماری روی کاربرد نظارتی ویدئویی مقیاس کوچک برای نظارت راه دور فرآیندهای صنعتی انجام شده است که در یک محیط توزیع شده پیاده سازی و به کارگیری می شود. نتایج بعد از توسعه پیاده سازی مرجع iLAND به دست آمده است. اعتباردهی حالت شامل تمام عناصر مورد نیاز و استقرارهای ممکن میان افزار برای اعتباردهی به آن می شود. گره های مختلف نمونه اولیه در اینجا داده شده است.

1) دستگاه ترمینال که شامل یک نسخه ماهواره ای از میان افزار عامل به عنوان نقطه دسترسی برای یک اپراتور خارجی برای تعامل با سیستم می شود که به نوبه خود، باعث فرآیند پیکر بندی دوباره می شود.

2) دستگاه کنترلر حاوی یک نسخه ستاره از میان افزار که در نتیجه منطق هماهنگی فرآیند پیکر بندی دوباره را تعبیه می کند. محرک های پیکر بندی دوباره را از دستگاه ترمینال را دریافت می کند و با همه گره های شامل خدمات برای پیکربندی مجدد سیستم تعامل دارد.

3) دستگاه سرویس با یک نسخه سیاره از میان افزار. دارای منطق برنامه در قالب سه پیاده سازی سرویس مختلف برای سه خدمات مختلف.



شکل 7. اجرای یک برنامه نظارت ویدئویی واقعی.

جدول 1 دفعات پیکربندی دوباره

<i>Time</i>	<i>Value</i>
Max.	9578 μ s
Min.	6530 μ s
Average	7753 μ s

این شبکه دارای یک سوئیچ 100/10 مگابیت در ثانیه است. تمام گره ها یک لینوکس اوبونتو 10.4 توزیع با یک پچ زمان واقعی را اجرا می کند که استفاده از برنامه ریزی پیشگیرانه مبتنی بر ضرورت برای برنامه ریزی زمان واقعی را میسر می سازد. سطح CBL، از قابلیت های زمان واقعی DDS به عنوان هسته های ارتباطی ستون فقرات در حال اجرا برای اجرای Prismtech v4.5.1 استفاده می کند.

نشاندنده متشکل از یک نرم افزار ویدئویی تمام HD است که نرخ فریم 24 فریم در ثانیه (فریم در ثانیه) را ارائه می دهد و دارای سه سرویس است: (i) ضبط ویدئو که از یک دوربین به عنوان ورودی فریم استفاده می کند، (ب) فشرده سازی ویدیویی، و (ج) نمایش ویدئو در صفحه نمایش. این سیستم، آلارم های شبیه سازی را تولید می کند که پارامترهای نظارت بر فرایند فیزیکی باید تغییر کنند (به عنوان مثال، دوربین های مختلف باید فعال شوند فرمت ، فشرده سازی باید تغییر کند تا وضوح تصویر افزایش یابد، و یا خود سیستم یک بار بیش از حد تشخیص داده است و گره ها به توازن مجدد نیاز دارند). فشرده سازی ویدئو یک سرویس مقیاس پذیر است بنابراین به عنوان پیاده سازی سرویس های مختلف ارائه می شود. اسلات پیکر بندی دوباره در 41 میلی ثانیه تنظیم می شود که یک محدودیت

زمانی امن در نظارت تصویری با کیفیت است. شکل 7 نتایج به دست آمده برای پیکر بندی دوباره در اجرای واقعی گره های سیستم را در حضور تداخل نشان می دهد.

نتایج اجرا روی اندازه فاصله نماینده 200 پیکر بندی دوباره را نشان می دهند که نشان دهنده رفتار کارآمد سیستم است. دفعات پیکربندی دوباره در حضور تداخل از خدمات و از فعالیت های کنترل پیاده سازی DDS خاص استفاده شده نشان داده شده است. در این شرایط، اثبات می شود که میان افزار دارای یک سطح قابلیت پیش بینی خوب و یک فرآیند پیکربندی دوباره زمان-محدود است (که در جدول 1 نشان داده شده است) با مقدار متوسط 7753 میکروثانیه از مقدار تنظیم شده در فاز مقداردهی اولیه تجاوز نمی کند. انحراف معیار 957 میکروثانیه است که نشان می دهد که دفعات پیکربندی دوباره به دست آمده عمدتاً پایدار و برای محیط های زمان-واقعی نرم، مناسب هستند. رابطه بین زمان های دفعات استاندارد و میانگین، 12٪ است. همچنین، قابل ذکر است که اجرای داخلی میان افزار، کافی و پایدار است و بنابراین انحراف اصلی در اندازه گیری ها از استفاده از شبکه TCP/IP بدون زمانبندی ترافیک زمان-واقعی نتیجه می شود که منجر به قابلیت پیش بینی افزایش یافته می شود.

VII. نتیجه گیری ها

این مقاله، معماری یک میان افزار را ارائه نموده است که پیکربندی دوباره زمان-قطعی در محیط های توزیع شده زمان واقعی با یک مدل نرم افزار مبتنی بر خدمات را ارائه نموده است. معماری نرم افزار تفصیل و مطابق با الزامات سیستم هیا هدف و ویژگی های موقتی آنها توجیه شده است. با توجه به راه حل های میان افزار موجود که فشرده سازی و دینامیک برنامه را هدف قرار می دهند، میان افزار ارائه شده توسط گنجاندن پیکربندی دوباره زمان-محدود و الگوریتم های ترکیب مبتنی بر خدمات را مساعدت می نماید که در بالای مدیریت منبع زمان-واقعی ساخته می شوند؛ این از اجرای قابل پیش بینی در برابر دیگر راه حل های موجود حمایت می کند. اعتبار معماری میان افزار روی یک محیط توزیع شده واقعی انجام شده است که یک سیستم نظارت ویدیویی مقیاس کوچک است؛ این محیط می تواند در واقع در محیط های متجاوزانه استفاده شود. اندازه گیری های آزمایشی، اعتبار طراحی عمودی و توسعه و پایداری مرزهای زمانی به دست آمده برای فرآیند پیکربندی مناسب با نیازهای سیستم های زمان-واقعی نرم توزیع شده را نشان می دهد.



این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی