به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش کارآموزی

محل کارآموزی : شرکت چکادنوآوران عصر اطلاعات (النون)

استاد

دکتر میبدی

محمدمهدی آقاجانی

۹۳۳۱۰۵۶

مهرماه ۹۶

چکیده

در دوره کارآموزی که در شرکت چکادنوآوران عصر اطلاعات (النون) گذرانده شد ، هدف اصلی یادگیری برنامه نویسی وب سرویس توسط زبانی به نام Scala بود. این زبان یک زبان تابعی[۱] و شی گرا[۲] میباشد که بر روی ماشین مجازی جاوا[۳] اجرا میگردد و به همین دلیل میتوان از کد های جاوا نیز در بین کدهای این زبان استفاده نمود. شرکت چکادنوآوران عصر اطلاعات بر روی یک پیام رسان فارسی با متدهای ایرانی و کاربردی به نام "بله" کار میکردند که سعی داشتند امکانات جدیدی را نیز به آن اضافه نمایند و از نمونه های مشابه خارجی (همچون تلگرام) متمایز گردند. دوره کارآموزی در تیم بخش سرور این شرکت گذرانده شد که مشخصا بر روی آزمون تحمل[۴] برای سرور پیاده سازی شده فعالیت میکردند. در همین راستا ابتدا فیلم های آموزشی دیده شد و سپس با ویژگی های اصلی سرور آشنایی حاصل گردید. هدف نهایی هم اجرای آزمون تحمل بر روی سرور اصلی بود تا مشخص گردد که توانایی ها و امکانات سرور اصلی پیام رسان در چه حدی قرار دارد. در ادامه نیز این پروژه تا حدودی انجام گردید و نتایج زیر حاصل شد :

۱. سرور میتواند تا ۱۰۰۰ درخواست را همزمان پاسخگو باشد
۲. سرور میتواند به هر شخص پیام ارسال کند و همچنین دریافت نماید و از این نظر مشکلی ندارد
۳. با تاخیر تقریبا ۱ ثانیه ای میتواند بین هر ۱۰۰۰ درخواست به مابقی درخواست ها با اطمینان ۱۰۰ درصد پاسخ دهد.
۴. همچنین مکانیزم تولید کلید برای هر پیام مشکل دارد و نمی تواند اعداد کاملا تصادفی تولید کند که باید این موضوع رفع گردد.
۵. همچنین از نظر پاسخ دهی سرعت مناسبی دارد و در مقیاس متوسط میتواند به هر درخواست در کمتر از یک دهم ثانیه پاسخ دهد.

در ادامه تصمیم بر آن شد که آزمون تحمل بر روی سرور اصلی نیز اجرا گردد تا مشخص شود آیا توانایی پاسخ گویی به حدود ۲۰ میلیون کاربر را به طور همزمان دارد یا خیر. البته با توجه به قدرت پردازشی سیستمی که سرور بر روی آن اجرا میگردد (۱۶ هسته و ۱ ترابایت حافظه RAM) پیش بینی میگردد که مشکلی وجود نداشته باشد. همچنین مقرر گردید تا بخش هایی که از حیث زمان پاسخگویی دچار ضعف هستند ، مجددا کدهای آن ها بررسی شود تا منبع مشکلات مشخص گردد.

---

[۱] Functional
[۲] Object Oriented
[۳] Java Virtual Machine (JVM)
[۴] Stress Test

فهرست مطالب

فهرست اشکال و جداول

مقدمه

دنیای امروز دنیای ارتباطات است. از این رو در عصر فناوری اطلاعات ، وجود ابزارهایی که ارتباطات مردم را کنترل میکند دور از ذهن و غیر قابل باور نیست و شرکت هایی که صاحب این ابزار ها هستند، بدون شک در عصر جدید قدرت بالایی در تحلیل آمایش های سرزمینی و اطلاعات ملل مختلف خواهند داشت از همین منظر شبکه های اجتماعی به عنوان یک ابزار قدرتمند معرفی میگردد و دولت ها از آنها به عنوان مولفه های قدرت خود استفاده مناسب میکنند. وجود شبکه های اجتماعی بیگانه در سرزمین هایی دور میتواند مولفه های قدرت اجتماعی یک دولت و یا یک شرکت را افزایش دهد و به همین میزان از قدرت اجتماعی جامعه هدف می کاهد.

در ایران و در چند سال اخیر ظهور شبکه اجتماعی تلگرام و فراگیر شدن آن باعث شد که برخی شرکت های توانمند در عرصه اطلاعات حتی با حمایت های دولتی شروع به فعالیت نمایند تا بتوانند راه کاری جایگزین به جای تلگرام ارایه کنند. یقینا برای کنار زدن تلگرام و جایگزینی محصول داخلی باید پارامتر های زیادی را ارضا نمود و همچنین کیفیتی مناسب از جهت سرعت و دقت ارتباطات ارایه کرد تا مردم و مخاطبین به جهت استفاده از محصول جایگزین اقناع گردند. در نتیجه استفاده از افرادی که توانایی بالای علمی و فنی دارند در این شرکت ها اجتناب ناپذیر است.

بی شک دانشگاه ها همواره منبع استعدادهای جوان و پویا بوده و هستند. از همین جهت این شرکت ها با نگاهی درون گرا و تیزبینانه به دنبال دانشجویان مستعد و فعالی هستند که بتوانند مشکلات خود در زمینه ارایه راهکار مناسب برای جایگزینی شبکه های اجتماعی بیگانه را در زمان کم و با دقت بالا حل نمایند. فرصت دوره های کارآموزی بهترین و طلایی ترین فرصت برای این شرکت ها می باشد که نیروهای خود را از همین محل پیدا کنند و بر روی آن ها سرمایه گذاری نمایند. وجود مخاطبین بسیار در این حوزه نیز باعث شده که خود به خود کار کردن در حوزه ارتباطات برای دانشجویان نیز جذاب و گیرا باشد.

آن چه که در بالا به آن پرداخته شد در واقع دلیل و هدف انجام دوره کارآموزی در شرکت هایی ست که در حوزه شبکه های اجتماعی فعالیت میکنند. شرکت چکادنوآوران عصر اصلاعات (النون) از جمله شرکت هایی ست که بر روی شبکه اجتماعی "بله" کار میکند و سعی دارد تا امکاناتی جدید و به روز را ارایه کند. یکی از مهمترین قمست های این شبکه احتماعی بخش سرور آن می باشد که باید توانایی بالایی در مدیریت درخواست ها[۵] داشته باشد. زیرا نرم افزار های پیام رسان به طور همزمان باید بتوانند درخواست های زیادی را پاسخگو باشند و این موضوع کیفیت بالایی را در سرور های این نرم افزار ها می طلبد.

[۵] Request Handling

سرور پیام رسان "بله" با زبان Scala نوشته شده است که زبانی تابعی[۶] و شی گرا[۷] می باشد. معماری کلی سیستم بر اساس ارسال پیام[۸] بوده و کاملا بر مبنای Actor می باشد. در ابتدای کار لازم بود تا آموزش های لازم فرا گرفته شود و سپس با ویژگی های این سرور آشنایی حاصل گردد. از همین جهت ابتدا فیلم های آموزشی که از سایت Coursera تهیه شده بود مشاهده گردید و بعد هم در طی جلسه ای ویژگی های اصلی سرور بیان شد.

در این گزارش سعی بر آن است که مراحل اصلی کار و نتایج به دست آمده ذکر گردد. همچنین نکات و روش های مورد استفاده ای که برای بهبود وضعیت سرور استفاده شد بیان می شود. هدف اصلی اجرای آزمون تحمل[۹] بر روی سرور اصلی می باشد و بررسی گردد که تا چه میزان میتواند به درخواست ها به طور همزمان پاسخ گویی کند. به همین دلیل ابتدا به طور محدود بر روی سروری آزمایشی مراحل تست برگزار شد که شامل انواع مختلف تست بود. سپس پس از انجام مراحل مختلف این آزمون بر روی سرور اصلی انجام گردید و نتایج مهمی نیز از آن حاصل شد که در ادامه به آن خواهیم پرداخت.

---

[۶] Functional

[۷] Object Oriented

[۸] Message Passing

[۹] Stress Test

معرفی محل کارآموزی

امروزه ما در عصری زندگی میکنیم که نمی توان اهمیت فناوری اطلاعات و ارتباطات را انکار نمود زیرا نمی توان بدون ابزار های این حوزه حتی زندگی عادی و روزمره خود را گذراند. همچنین با توجه به رشد فزاینده این حوزه فضای بسیار وسیعی فراهم گردیده است تا علاوه بر هزینه های کمتر با تکیه بر نیروی انسانی خلاق و ایده پرداز بتوان به یک تجارت بزرگ در مقیاسی جهانی تبدیل شد.

با توجه به آن چه که در بالا گفته شد شرکت چکادنوآوران عصر اطلاعات (النون) در سال ۱۳۹۱ با هدف کسب سهمی شایسته در این حوزه تشکیل شد ؛ تا با تکیه بر استعداد های نیروهای خود و ریسک پذیری بالا بتواند سرویس های با کیفیتی ارایه کند که توانایی رقابت در عرصه جهانی را داشته باشد. این شرکت از بدو تاسیس تا به امروز سعی داشته تا به حوزه مدیریتی توجه ویژه ای داشته باشد و همزمان با حفظ کیفیت بدنه فنی خود بدنه مدیریتی خود را نیز تقویت نماید از این رو مدیران شرکت به مدیریت منابع انسانی و مدیریت بازاریابی توجه ویژه ای دارند.

این شرکت جلب رضایت مشتریان و جلب اعتماد آن ها را سرلوحه کار خود قرار داده و این موضوع یک سیاست کلی حاکم بر تمامی فعالیت های شرکت می باشد زیرا رضایت مشتری و پاسخگویی به نیاز ها مشتری باعث رشد و موفقیت یک شرکت میگردد.

در دنیای فناوری اطلاعات فاکتوری که باعث برتری و موفقیت میگردد وجود ایده های نو و تازه است که حاصل وجود ذهن هایی خلاق و پویا می باشد. از همین رو شرکت النون ارزش بسیار بالایی برای ذهن های ایده پرداز قایل است و همان طور که از نظر گذشت اهتمام ویژه ای به بدنه نیروی انسانی خود دارد.

از جمله ویژگی هایی که شرکت چکاد نوآوران عصر اطلاعات (النون) را از رقبا و دیگران متمایز می سازد داشتن تیم فنی با توانایی های بسیار بالا می باشد که علاوه بر تخصص در حوزه فنی در کارهای تحقیقاتی و علمی نیز فعالیت های ویژه ای دارند و همین امر نشان از اهتمام ویژه این شرکت به امر تحقیق و توسعه دارد. النون با علم بر ارزش بسیار بالای نیروهای انسانی خود همواره تلاش فراوانی برای رشد و پیشرفت همکاران خود داشته است.

ارزش های شرکت النون

بر اساس تجارب مدیران شرکت و مطالعاتی که در حوزه ارتباط با مشتری و توسعه کسب و کار انجام گردیده است ارزش هایی بر شرکت چکادنوآوران عصر اطلاعات حاکم شده که به برخی از آن ها باید اشاره نماییم :

- **یادگیری مستمر** : برای کارکنان این شرکت یادگیری و رشد یک ارزش بنیادی است. به همین سبب همه تلاش میکنند که در سطح فردی و هم در سطح سازمانی همواره در این یادگیری سهیم باشند.

- **نوآوری مستمر در محصولات :** با توجه به اهمین نوآوری در بقا و رشد شرکت ها در اقتصاد پویای امروز دنیا ، این شرکت نوآوری را رمز ماندگاری خود میداند. به همین جهت در تمام سطوح شرکت تلاش می شود که نوآوری در محصول و فرآیند های شرکت جریان داشته باشد.
- **شعف مشتری :** کارکنان شرکت النون اعتقاد راسه دارند که فلسفه حضور آن ها در شرکت ، ارایه خدمت به مشتریان گسترده محصولات شرکت است. به عبارتی دیگر بدون حضور و رضایت مشتریان، فلسفه وجود شرکت نقض می گردد. به همین سبب کارکنان سعی میکنند خارج از چهارچوب های قراردادی و عرفی ، تا حد امکان موجب رضایت و شعف مشتری گردند.
- **شرکتی دانش بنیان :** مسلما اقتصاد آینده کشور بر پایه نوآوری و کوشش های شرکت های دانش بنیان شکل میگرد. به همین دلیل برخی مزایای تشویقی برای این شرکت ها در نظر گرفته شده است. مثلا این شرکت ها میتوانند برخی از نیرو های اصلی خود را برای انجام پروژه تخصصی به جای دوره سربازی به نیروهای مسلح معرفی نمایند. امید است که در آینده با تلاش و استقامت این شرکت ها اقتصاد کشور بیش از پیش شکوفا گردد. در همین راستا در سال ۱۳۹۴ بر اساس بررسی های فنی و مدیریتی انجام شده توسط معاونت علمی و فناوری ریاست جمهوری از محصولات و فرآیندهای النون ، این شرکت به عنوان شرکتی دانش بنیان معرفی و شناخته شد.
- **بهره گیری از حلقه گسترده مشاورین :** مطمینا دانش و تجربه بشر در ذهن افراد مختلف پراکنده است و هر چه دامنه مشورت انسان بیشتر شود بهره بیشتری از این دریای دانش می برد. در همین راستا و با توجه به ارتباط گسترده کارکنان شرکت هم در بخش فنی و هم در بخش مدیریتی از مشورت حلقه گسترده ای از مشاورین استفاده میگردد.

حوزه کاری شرکت النون

ماموریت اصلی شرکت النون پیاده سازی ایده های نو در زمینه سرویس های اینترنتی میباشد. این شرکت تا کنون چندین سرویس با قابلیت های متنوع به عموم عرضه کرده و بر روی سرویس های جدیدتر نیز در حال تحقیق و توسعه می باشد. برخی از سرویس هایی که النون تا کنون عرضه کرده است عبارتند از :

- سرویس آموزشی از طریق سرگرمی
- سرویس های قرآنی
- پست الکترونیک عمومی و سازمانی با قابلیت های جدید و یکپارچگی با شماره تلفن همراه
- سرویس های ارتباطی و گفت و گو محور

اطلاعات تماس شرکت چکادنوآوران عصر اطلاعات

آدرس : تهران ، خیابان ستارخان ، خیابان شهید دکتر حبیب الله ، تقاطع سروش (یکم) ، پلاک ۹۶

شماره تلفن : ۶۳۱۰۸۲۷۵

نمابر : ۶۶۵۰۸۴۷۹

شرح موضوع کارآموزی

موضوع دوره کارآموزی ساخت یک برنامه برای انجام آزمون تحمل[۱۰] برای سرور اصلی پیام رسان است. همچنین فعالیت اصلی شرکت ، طراحی و توسعه نرم افزار پیام رسان "بله" می باشد که یک پیام رسان تمام ایرانی ست و هم نسخه سمت کاربر (کلاینت) و هم نسخه سرور آن طراحی متخصصین شرکت می باشد. در طراحی نسخه کلاینت نرم افزار "بله" از نسخه کلاینت تلگرام استفاده نشده بلکه تماما طراحی آن توسط خود شرکت انجام گرفته است. همچنین یام رسان "بله" در سمت کلاینت دارای نسخه های IOS و Android می باشد که به ترتیب با زبان های Objective-C و جاوا نوشته شده اند.

اما در سمت سرور ، این نرم افزار از زبان Scala استفاده کرده بود. زبان Scala یک زبان قدرتمند برای طراحی و توسعه وب سرویس ها در مقیاس بزرگ و کار با داده های بزرگ[۱۱] می باشد. این زبان به روش برنامه نویسی تابعی[۱۲] و شی گرا[۱۳] استفاده میشود. برای شروع توسعه برنامه آزمون تحمل ابتدا لازم بود که آشنایی با زبان Scala حاصل گردد که با دیدن فیلم های آموزشی که مسوول آموزش تیم سرور در نظر گرفته بود این امر محقق گردید. همچنین برای توسعه بهتر و سریعتر نیاز است که یک چهارچوب کاری[۱۴] مناسب انتخاب گردد که انتخاب تیم سرور در شرکت النون محیط کاری Akka بود که مناسب برای توسعه وب سرویس ها می باشد.

توسعه ابزار آزمون تحمل یکی از مهمترین وظایف در طول توسعه سرور میباشد لذا در همین جهت بخش قابل توجهی از ظرفیت نیروی انسانی شرکت النون نیز خود درگیر همین موضوع بودند. این ابزار آزمون تحمل شامل بخش های مهمی از قبیل ارتباط با خارج (که توسط پروتکل HTTP انجام میگرفت) ، مانیتورینگ برنامه ، تبادل اطلاعات و پیام ها (که توسط پروتکل Protobuf انجام میگرفت و این محصول متعلق به شرکت گوگل می باشد و جلوتر به آن اشاره خواهد شد) ، ذخیره سازی اطلاعات ، تنظیمات سرور و مدیریت کلاینت ها می باشد.

در قسمت ارتباط با خارج که توسط پروتکل HTTP انجام میگرفت باید برنامه ای طراحی میشد که بتوان همزمان به کلاینت های مختلف دستور داد که عمل خاصی را انجام دهند که از این طریق بتوان عملکرد سرور را مورد بررسی قرار داد . مثلا لازم بود که دستوراتی پیاده سازی شود تا کلاینت ها به طور تصادفی به چندین کلاینت دیگر پیام بدهند و د راین بین عملکرد سرور که په مقدار از این پیام ها سالم و صحیح دست مقصد میرسید حایز اهمیت بود.

---

[۱۰] Stress Test

[۱۱] Big Data

[۱۲] Functional Programming

[۱۳] Object Oriented

[۱۴] Framework

برای قسمت تبادل اطلاعات قبلا از روتکل Json استفاده شده بود که باید این پروتکل به پروتکل Protobuf تغییر می یافت و همین موضوع خود زمان خاصی را می طلبید. همچنین در بخش ذخیره سازی اطلاعات باید از PostGre SQL استفاده میشد و این محصول باید جایگزین سیستم redis می گردید. از مشکلات استفاده از redis مصرف زیاد حافظه داخلی (RAM) می باشد.

در قسمت مدیریت کلاینت ها نیز لازم بود که دستورات جدیدی ازقبیل ارسال تصادفی و مستقیم به یک کاربر خاص و یا ارسال در بازه زمانی مشخص ، یا چاپ خروجی های معین برای هر دستور اضافه می گردید تا بتوان موضوعات مختلفی را برای سرور آزمایش کرد.

همچنین هر یک از قسمت هایی که در بالا ذکر شد نیز باید دوباره آزمایش می شدند تا از عملکرد آن ها و درستی گزارشاتی که از آن ها حاصل میشد اطمینان پیدا کرد. نکته مهم این بود که باید ویژگی های اضافه شده توانایی ارسال صحیح ۹۹ درصد پیام ها را داشته باشند که البته در چشم انداز شرکت رسیدن به درصد ۱۰۰ مد نظر قرار گرفته بود.

یکی دیگر از موضوعات مهم هم توانایی مانیتورینگ کامل سرور اصلی بود تا واکنش های آن به دستوراتی که در بالا ذکر شد مشخص گردد. باید دقیقا معین میگردید که سرور در پاسخ به آزمون های بالا چه واکنشی نشان داده است و مشخص کردن مکانیزمی دقیق برای مانیتورینگ سرور نیز یکی زا چالش های پیش رو بود. مثلا اگر دستور ارسال پیام برای ۱۰۰ نفر به طور تصادفی ارسال میشد باید معین میگردید که آیا سرور توانست تمام این پیام ها را به مقصد مشخص بفرستد یا خیر و با توجه به این نتیجه باید تصمیم های جدید اتخاذ میگردید.

توضیح اصطلاحات پرکاربرد

در این فصل اصطلاحات مهم و پرکاربرد که در طول دوره کارآموزی استفاده شده و به کار گرفته شده اند توضیح داده خواهد شد.

اسکالا(Scala) : اسکالا (به انگلیسی: Scala) یک زبان برنامه‌نویسی شیءگرا[۱۵] و تابعی[۱۶] است. نام اسکالا آمیزه‌ای است از "scalable" و "language" به معنی زبان مقایس‌پذیر، از اهداف اصلی ایجاد زبان اسکالا، ارائه زبانی است برای تولید نرم‌افزار مقیاس‌پذیر به روشی چابک و سریع، و به دور از مشکلات مرسوم. اسکالا تلفیق زبان‌های شی‌گرا همچون روبی و جاوا با زبان‌های تابعی همچون Haskell و Erlang است. از دیدگاه چابکی و کارایی، عده‌ای اسکالا را جمع دو دنیای زبان‌های پویا[۱۷] و ایستا[۱۸] می‌دانند. یکی از دلایل دیگری که باعث مقبولیت و همچنین کارایی بالای این زبان می‌شود، دستور زبان (Syntax) منعطف آن است. اسکالا توسط پروفسور مارتین اودرسکی که خالق Generic های جاوا و از برترین توسعه دهندگان کامپایلر javac می‌باشد، در سال ۲۰۰۳ طراحی و به مرور توسط ایشان و جامعه اسکالا بهبود و توسعه داده شده است.(Ying, 2014)

این زبان در سمت سرور برای پیام رسان "بله" مورد استفاده قرار گرفته است و ماژول آزمون تحمل نیز بر مبنای همین زبان نوشته خواهد شد. برخی از مهمترین ویژگی های این زبان عبارت است از :

- رایگان و متن باز
- شی گرایی
- تابعی بودن
- دارای ابزار های توسعه ای گوناگون
- بررسی هنگام کامپایل (Type-Safe)

این ویژگی ها باعث شده که این زبان از تمامی زبان های هم سطح خود مانند .... , Go , Java , Ruby متمایز گردد.

ماشین مجازی جاوا(JVM) : ماشین مجازی جاوا (به انگلیسی: Java Virtual Machine) که به صورت اختصار JVM مشخص می‌شود، مجموعه‌ای از برنامه‌های نرم‌افزاری و ساختمان‌داده‌هایی است که برای مدلسازی ماشینی مجازی اجرای برنامه‌های سایر رایانه‌ها و اسکریپت‌های دیگر سامانه‌هاُ به کار می‌رود. مدلی که جی‌وی‌ام برای اجرا می‌پذیرد، شکلی از زبان میانی را اجرا می‌نماید. به این زبان میانی جاوا بایت کد اطلاق می‌گردد. این زبان میانی،

---

[۱۵] Object Oriented

[۱۶] Functional

[۱۷] Dynamic-Type

[۱۸] Static-Type

به صورتی مفهومی، مجموعه‌ای از دستوراتِ زبان برنامه‌نویس مبتنی به پشته و معماری قابلیت مبتنی بر امنیت است. سان، ادعا نموده که هم اکنون، ۴/۵ میلیارد دستگاه در جهان از جی‌وی‌ام استفاده می‌نماید.(Oracle, 2017)

اگرچه جی‌وی‌ام در ابتدا تنها با هدف ترجمه برنامه‌های جاوا پا به عرصه گذاشته بود، اما امروزه بسیاری از زبان‌های دیگر نیز قادر به اجرای برنامه‌های خود بر روی آن هستند.(Tolksdorf, 2013)

آزمون تحمل (Stress Test) : برنامه ای ست که سرور هدف را در برابر شرایط مختلف و پیچیده و سخت قرار میدهد و سرور را ارزیابی میکند. همچنین خروجی های لازم را نیز در اختیار قرار میدهد.

Akka : یک چهارچوب توسعه[۱۹] رایگان و متن باز میباشد که ساخت نرم افزار های هم روند[۲۰] و توزیع شده[۲۱] را بر روی JVM بسیار راحت میکند. این چهاچوب مدل های زیادی را برای برنامه نویسی همروند پشتیبانی می کند اما تاکید ویژه ای بر روی مدل Actor-Based concurrency دارد. این چهارچوب از زبان Erlang برخاسته است. Akka با زبان اسکالا نوشته شده و هر دو زبان اسکالا و جاوا را نیز پشتیبانی میکند. بعد از اسکالا 2.10 این چهارچوب یعنی Akka عضوی از کتابخانه استاندارد اسکالا شد.(Jovanovic, 2013)

همچنین سرور های زیادی همچون Ebay , Amazon از این ابزار استفاده مینماید.

Protobuf : در واقع یک نوع پروتکل ابداعی از سمت گوگل می باشد که برای سریال کردن داده ها و ارسال آن ها بین برنامه های مختلف با زبان های مختلف و یکسان قابل استفاده می باشد. مهمترین ویژگی آن سرعت بالای آن در برابر دو رقیب سنتی یعنی Json , XML است اما بزرگترین مشکل آن سختی پیاده سازی و پیچیدگی های مربوط به مرحله پیاده سازی می باشد. البته فعلا زبان های رایج مانند ... , scala , php , java , c++ , c از این پروتکل پشتیبانی میکنند. برای مقایسه عملکرد protobuf با دیگر استاندارد ها به جدول ۱ مراجعه نمایید.

Log4j : یک سیستم log کردن اطلاعات برای زبان جاوا و اسکالا می باشد. این سیستم با فرمت دلخواه و قابل تنظیم تمامی سطوح خطا را میتواند به گونه مناسب نمایش دهد و امر خطایابی را بسیار راحت می نماید.

PostGreSQL : یک ابزار مدیریت پایگاه داده شی-رابطه[۲۲] می باشد که برای سیستم عامل های مختلفی از جمله لینوکس و ویندوز و مک در دسترس قرار دارد. تیم توسعه آن شامل تعداد زیادی از افراد داوطلب است که تا به امروز افزونه های زیادی برای این سیستم نوشته اند و کار مدیریت داده را بسیار راحت نموده اند.

---

[۱۹] Framework

[۲۰] Concurrent

[۲۱] Distributed

[۲۲] relational

	FlatBuffers (binary)	Protocol Buffers LITE	Rapid JSON	FlatBuffers (JSON)	pugixml	Raw structs
Decode + Traverse + Dealloc (1 million times, seconds)	0.08	302	583	105	196	0.02
Decode / Traverse / Dealloc (breakdown)	0 / 0.08 / 0	220 / 0.15 / 81	294 / 0.9 / 287	70 / 0.08 / 35	41 / 3.9 / 150	0 / 0.02 / 0
Encode (1 million times, seconds)	3.2	185	650	169	273	0.15
Wire format size (normal / zlib, bytes)	344 / 220	228 / 174	1475 / 322	1029 / 298	1137 / 341	312 / 187
Memory needed to store decoded wire (bytes / blocks)	0 / 0	760 / 20	65689 / 4	328 / 1	34194 / 3	0 / 0
Transient memory allocated during decode (KB)	0	1	131	4	34	0
Generated source code size (KB)	4	61	0	4	0	0
Field access in handwritten traversal code	typed accessors	typed accessors	manual error checking	typed accessors	manual error checking	typed but no safety
Library source code (KB)	15	some subset of 3800	87	43	327	0

جدول ۱ مقایسه سرعت عملکرد و میزان استفاده از فضای Protocol Buffer(protobuf) با دیگر استاندارد ها

گزارش دوره کارآموزی

در این قسمت گزارش های دوره ای که در طول مدت کارآموزی نوشته شده است می آید. تمام طول دوره کارآموزی شامل سه بخش زیر میباشد :

۱. آموزش و فراگیری زبان Scala و فریم ورک Akka
۲. آشنایی با سرور و قابلیت های آن و همچنین توسعه چند برنامه کوچک با اسکالا
۳. پیاده سازی وظایف محوله درباره بخش آزمون تحمل

این گزارش های دوره ای بنا به درخواست مسوول کارآموزان شرکت چکادنوآوران عصر اطلاعات نگاشته شده است و به این علت که مستند و دقیق است همان گزارشات برای تکمیل این گزارش آورده شده است.

لازم به ذکر است که این گزارشات شامل ۴ بخش زیر می باشند که قالب کلی است که مسوول کارآموزان شرکت برای نگارش گزارش ها در نظر داشته است :

۱. توضیح مختصر درباره آموزش های روزانه
۲. اقدامات انجام شده در آن روز
۳. برنامه اقدامات فردا
۴. مشکلات پیش آمده

گزارش های دوره ای که در ادامه خواهد آمد تمامی مطالب را پوشش میدهد همچنین بره دلیل خوانایی بهتر هر گزارش در صفحه ای جداگانه قرار گرفته است.

تاریخ : ۹۶/۰۵/۰۱ لغایت ۹۶/۰۵/۰۵

شماره گزارش : ۱

توضیح مختصری از آموزش های دیده شده این دوره :

در این هفته فیلم های آموزشی Functional Programming با محوریت زبان Scala دیده شد که این دوره آموزشی توسط آقای Martin Odersky ارایه شده است. همچنین در ادامه فیلم های آموزشی مربوط به Reactive Programming نیز مشاهده گردید.

در این فیلم ها درباره خصوصیات زبان Scala بحث شده است. همچنین درباره Higher Order Function ها بحث شد. سپس به نحوه کلاس بندی و متد ها در Scala اشاره گردید و مفاهیم اصلی در این زبان همانند OOP و غیره صحبت شد. در ادامه هم چندین نکات ریز دستوری[۲۳] در زبان Scala مورد بحث قرار گرفت.

اقدامات انجام شده امروز:

- حل برخی از مسایل مطرح شده در فیلم آموزشی
- پیاده سازی مسایل مطرح شده با Scala

برنامه اقدامات فردا :

- مشاهده و اتمام تمامی فیلم آموزشی Reactive Programming

مشکلات پیش آمده :

- عدم درک درست ساختار For در زبان Scala
- عدم درک درست مفهوم Scalability

[۲۳] Syntactical

تاریخ : ۹۶/۰۵/۰۷ لغایت ۹۶/۰۵/۱۲

شماره گزارش : ۲

توضیح مختصر آموزش های این دوره :

مباحث مربوط به تساوی مقادیر بررسی شد که در آن نظریه تساوی مطرح میگردد و با بیان چند مثال سعی میکند مساوی بودن دو value را تشریح کند.در فیلم بعدی همچنین Observer pattern معرفی گردید و اشکالات آن از قبیل strong coupling بررسی گردید و بعد مفهوم سیگنال در scala مطرح شد که تا حدودی معایب observer pattern را جبران میکند.همچنین نحوه پیاده سازی (functional reactive programming) FRP نیز بررسی شد.

فصل چهار از دوره آموزشی reactive programming با موضوع monad و تاثیرات آن شروع میگردد و با دونوع کلی برنامه نویسی که async , sync میباشد ادامه پیدا میکند و در ادامه بیشتر به معرفی برنامه نویسی async میپردازد.برای بررسی تاثیرات monad اول از بحث exception ها شروع میکند و نشان میدهد که استفاده از monad ها میتواند exception handling را ساده تر کندکه در این بخش از try استفاده میکند در بحث بعدی به تاثیردیگر استفاده از monad اشاره میکندکه جلوگیری از مشکلات ناشی از تاخیر و کاهش آن است این مبحث را با مثالی از شبکه اینترنت و ارسال و دریافت packet ها پیگیری میکندو به تاخیر هایی که در کد ها ی ما پنهان است اشاره مینماید و وجود این تاخیر ها را موجب بروز مشکل میداند در نتیجه برای جلوگیری از مشکلات ناشی از تاخیر استفاده از monad ها پیشنهاد میشودکه در این بخش از future استفاده میکندسپس combinator ها را بر روی future توضیح میدهد و همچنین توضیح میدهد برای استفاده امن و مقاوم از future چگونه باید از function های recover و recoverWith استفاده کنیم.

سپس در فصل بعدی actor را معرفی میکندو علت استفاده از آن را بیان میکندکه در واقع به خاطر رعایت اصل non-blocking objects از آن استفاده میشود زیرا پیروی از sync objects میتواند باعث بروز مشکلاتی از قبیل deadlock و ... گردد.

اقدامات انجام شده این دوره :

- دیدن فیلم های آموزشی reactive programming از فصول سوم و چهارم و بخش پنجم با نظر مربی دیده نشد.
- کتابخانه akka بر روی IDE نصب گردید

برنامه اقدامات دوره بعد:

- مشاهده فیلم آموزشی Reactive Programming فصول ششم و هفتم و همچنین برنامه نویسی با Actor Model

مشکلات پیش آمده :

- پیاده سازی while با توابع درونی Scala دچار مشکل شد که با مراجعه به مربی برطرف گردید
- عدم درک درست مفاهیم و کاربرد Monad که با مراجعه به مربی برطرف گردید
- عدم درک درست مفهوم partial function که با مراجعه به مربی برطرف گردید

تاریخ : ۹۶/۰۵/۱۴ لغایت ۹۶/۰۵/۱۶

شماره گزارش : ۳

توضیح مختصر آموزش های این دوره :

در ابتدا فیلم آموزشی message passing syntax مشاهده شد که قواعد انتقال پیام بین actor ها را مشخص میکند که البته در ادامه متاسفانه سیستم لپ تاپ بنده دچار مشکل شد و مجبور شدم تا دوباره لینوکس را نصب کنم که البته نصب آن با توجه به اینکه وایرلس لپ تاپ را نمیشناخت طول کشید و بعد هم با توجه قطعی طولانی نت شرکت نتوانستم مشکل را برطرف کنم و متاسفانه روند دیدن فیلم های آموزشی متوقف گردید.

اقدامات انجام شده این دوره :

- دیدن فیلم آموزشی درباره message passing
- نصب مجدد سیستم عامل لینوکس

برنامه اقدامات دوره بعد:

- تمرین برنامه نویسی با Actor Model
- شروع دیدن فیلم های Akka

مشکلات پیش آمده :

- قطعی طولانی نت شرکت
- عدم شناخت کارت شبکه لپ تاپ توسط سیستم عامل لینوکس

تاریخ : ۹۶/۰۵/۱۷ لغایت ۹۶/۰۵/۱۹

شماره گزارش : ۴

توضیح درباره آموزش های دیده شده در این دوره :

دیدن فیلم های Akka را شروع کرده و یک تکه کد از آن زدم. سری اول فیلم در مورد concurrency بود سپس درباره actor model صحبت شده که در آن اگر هر کلاس یا قسمتی از برنامه را Actor درنظر بگیریم Actor ها با حرف زدن با یکدیگر ارتباط برقرار میکنند که این ارتباط در واقع از طریق ارسال پیام به یکدیگر محقق خواهد شد. هم چنین از دیگر ویژگی های این ارتباط ، حالت non-blocking و async بودن است. نکته مهم این که حالت race condition کنترل شده و هیچ گاه رخ نمیدهد.

در ادامه با یک مثال که مثال شمارنده بود کل سیستم را یک بار پیاده سازی کرد و بر روی آن مفاهیم را دوباره توضیح داد. همچنین به پیاده سازی کم دردسر FSM ها با فریم ورک Akka اشاره کرد و به جای مفهوم پایه CRUD مفهوم CQRS را معرفی نمود که فار خواندن و نوشتن را از هم جدا میکند.

اقدامات انجام شده این دوره :

- دیدن بخش اول فیلم های آموزشی Akka

برنامه اقدامات دوره بعد :

- دیدن مابقی فیلم Akka
- پیاده سازی یک مثال با actor model

مشکلات پیش آمده :

- در این دوره مشکل خاصی اتفاق نیفتاد

تاریخ : ۹۶/۰۵/۲۲ لغایت ۹۶/۰۵/۲۶

شماره گزارش : ۵

توضیح مختصر اقدامات انجام شده این دوره :

در ابتدا مختصر توضیحی راجع به actor داده میشود و چند ویژگی آن را بیان میکند که مهمترین آن این است که actor ها فقط از طریق پیام های async با هم در ارتباط هستند.از طرفی actor context را مطرح میکند که در واقع مسیول عملیات اجرایی همان actor است.actor ها متدی به نام receive دارند که باید پیاده سازی شود و در پیاده سازی آن از pattern matching استفاده میکنند . در ادامه متد become را معرفی میکند که این متد در context هر actor وجود دارد و یک متد async هست . این متد بلافاصله اجرا نمیشود بلکه هنگام رسیدن پیام بعدی اجرا میگردد.همچنین برای ساخت actor می توان از متد actorOf استفاده کرد.در فیلم بعدی در ابتدا معرفی میکند که آدرس هر actor بهصورت actorRef می‌باشد و همچنین در ضمن ارسال پیام آدرس ارسال کننده پیام نیز به گیرنده در قالب sender ارسال میگردد . در موضوع بعدی مطرح میکند که یک actor به تنهایی به صورت single-threaded کار میکند یعنی پیام‌ها به صورت کاملاً ترتیبی اجرا میشوند. سپس مطرح میکند که بهتر است پیام‌ها را در قالب object ها در بیاوریم و در همین جا می‌گوید که actor ها همانند آدم‌ها می‌توانند با هم همکاری داشته باشند غالباً این همکاری منتج به ایجاد actor دیگری می‌شود که ارتباط این دو را کنترل میکند و این موضوع را با مثال bank account تکمیل میکند.سپس این بحث را مطرح میکند که پیام‌های بین actor ها نمیوانند به خودی خود قابل اطمینان باشند زیرا در ازای ارسال پیام مشخص نیست که ما جواب خواهیم گرفت یا نه به همین خاطرتکنیکهایی را باید راعایت کنیم که ارسال درست پیام را تضمین کند.در آخر هم توضیح میدهد که ترتیب رسیدن پیام‌ها به گیرنده اصلاً مشخص نیست و برای مدیریت ارتباط بهتر است که سیگنال acknowledge تعریف کنیم.

در فیلم بعدی مبحث طراحی یک سیستم بر اساس actor را مطرح میکند و میگوید برای این طراحی کافی ست تصور کنیم یک اتاق پر از آدم داریم! میتوان به هر یک وظایفی اختصاص داد و کافی ست با actor ها نیز همانند یک آدم برخورد کنیم و سعی کنیم task ها را تا آنجا که می‌شود دسته بندی و طبقه بندی کنیم و هر یک را به گروهی از actor ها محول نماییم این موضوع را با مثالی از بدست آوردن لینک های یک صفحه وب به صورت تو در تو پیگیری میکند و دیاگرامی از actor ها میکشد که نحوه تعامل آن‌ها را مشخص میکند. سپس بعد از طراحی سیستم در فیلم بعدی به سراغ تست یک سیستم می‌رود که در اینجا کتابخانه Akka test kit را معرفی مینماید.در ابتدای فصل هفت به سراغ نحوه هندل کردن fail شدن میرود.در این مبحث مطرح میکند که بروز مشکل در سیستم actor ها همانند زندگی انسان هاست . به فرض مثال اگر در یک شرکت فردی دچار مشکل شود موضوع

به رئیس آن قسمت اطلاع داده می‌شود و بعد از رفع شدن ممکن است موضوع به اطلاع مسیولین بالاتر نیز برسد.همین سیستم در actor ها نیز پیاده می‌شود و باعث می‌شود تا سیستم actor ها resilient باشد . البته این موضوع به دوظرط حادث می‌شود اول اینکه در اثر بروز مشکل آن مشکل به بقیه سیستم سرایت پیدا نکند(که این موضوع با توجه به کپسوله بودن کامل actor ها خود به خود برقرار است) و موضوع دیگر این است که fail شدن یک actor نباید توسط خودش هندل شود و باید توسط actor دیگر هندل شود (ما بقی این مبحث با نظر آقای جوان دیده نشد)

در ابتدای فصل هشتم درباره eventual consistency صحبت میکند و سعی میکند actor ها را اینگونه معرفی کند که یک actor طراحی می‌شود تا بتواند منتشر شود(بر روی شبکه) و گسترش یابد برای همین منظور در ابتدا تأثیرات یک ارتباط network را نسبت به ارتباط local بررسی میکند که از مهمترین تأثیرات آن تأخیر بیشتر و کمبود پهنای باند و همچنین تأثیرات کیفی آن از قبیل از بین رفتن پیام یا پاسخ آن است هم چنین در برنامه نویسی برای سیستم‌های چند هسته ای نیز تقریباً همین مشکلات وجود دارد . در ادامه به معرفی actorPath میپردازد و فرق های ref و path را بیان میکند و همچنین متد actorSelect از actor context را معرفی میکند که با داشتن path میتوان actorRef را بدست آورد و به actor مورد نظر پیام ارسال کرد سپس به معرفی cluster میپردازد که درواقع گروهی از actor ها ست که بر روی یک وظیفه همکاری میکنند و ما بقی actor ها میتوانند عضو یک cluster بشوند که این عمل با ارسال در خواست عضویت به هریک از گره‌های cluster امکان‌پذیر است.

اقدامات انجام شده این دوره :

- دیدن فیلم های آموزشی فصل ششم و هفتم و بخضی از فصل هشتم فیلم آموزشی Akka
- پیاده سازی تمرین bank account با actor model

برنامه اقدامات برای دوره بعد :

- مشاهده فصل آخر فیلم reactive programming

مشکلات پیش آمده :

- عدم درک درست context.become که با مراجعه به مربی برطرف گردید
- عدم درک نحوه کارکرد cluster

تاریخ : ۹۶/۰۵/۲۱ لغایت ۲۵ /۹۶/۰۵

گزارش : ۶

توضیح مختصر آموزش این دوره :

در ابتدا در مورد cluster توضیح میدهد که درواقع cluster مجموعه‌ای از node های actor است که در آن هر گره میداند که کدام actor عضو cluster نیست و کدام یک هست.تمامی اعضای cluster می‌توانند بر روی یک task با هم همکاری داشته باشند و همچنین سایز cluster به مرور زمان میتواند تغییر کند. در ادامه مبحث eventual consistency را بررسی میکندکه اگر مثلاً از sync استفاده کنیم به سطح strong consistency میرسیم زیرا تغییرات داده سریعاً در مقدار var نمایان می‌شود اما اگر یک دیتا به اشتراک گذاشته شده بین دو actor به گونه‌ای update شود که پس از هر update با اطلاع actor آپدیت کننده بقیه هم از آپدیت شدن آن دیتا با خبر شوند به این حالت eventual consistency میگویند. حال باید توجه کرد که actor ها خود نوعی از ثبات را دارا میباشند اما به صورت پیش‌فرض eventual consistency نیستند و هنگام همکاری با یک دیگر باید در کد نویسی eventual consistency را خودمان رعایت کنیم. در فیلم بعدی جزییات بیشتری در مورد actor ها بازگو میکند که نوع actor اسمی نمیباشد بلکه ساختاری ست و یک actor با ساختار خودش تعریف می‌شود که ساختار آن در واقعاً پیام‌هایی ست که actor به آن‌ها جواب میدهد پس با توجه به این مطلب ممکن است در طول زمان نوع actor که درواقع ساختمان آن است تغییر کند از همین رو نوع actor خیلی نمی‌تواند مفید باشد و باید از ساختار آن استفاده کرد .

در حین فیلم از pipeTp استفاده کرده بود که این تابع درواقع نتیجه یک Future را به یک actor میفرستد و این کار را به صورت امن انجام میدهد.

در فیلم بعدی به موضوع scalability اشاره میکند . باید توجه داشت که در سیستم‌هایی که single thread هستند در اثر زیاد شدن request ها زمان پاسخگویی افت محسوسی میکند پس باید scale up انجام داد به این معنی که باید resource ها را افزایش دهیم و مثلاً cpu اضافه کنیم در این صورت می‌توان برای انجام task ها چندین actor ایجاد کرد و آن‌ها را بر روی منابع مختلف پخش کرد و با توجه به کپسوله سازی شدید actor ها این کار راحت انجام میگیرد. در حالت single-threaded یک task بر روی یک actor انجام می‌شود اما در این حالت همان actor میتواند request های مختلف در آن task را به actor های دیگر ارجاع دهد و از آنجا که تنها با پیام‌ها ارتباط برقرار می‌شود client احساس نمیکند که این عمل تقسیم وظایف صورت گرفته است اما همین عمل تقسیم وظایف میتواند با دو استراتژی انجام گردد :

- stateful : که بر مبنای برخی الگوریتم ها انجام میگردد

- stateless : که تقسیم request ها بین actor ها به صورت random انجام می‌شود

سپس برخی از الگوریتم ها را مطرح میکند مانند round robin , shared queue

در فیلم بعدی و آخرین فیلم این دوره در مورد responsiveness صحبت میکند.تعریف میکند که توانایی سیستم در پاسخگویی به درخواست ها در مدت محدود را responsiveness میگویند و طبق مباحث قبلی باید به یاد داشته باشیم که رعایت سه اصل قبلی reactive programming میتواند responsiveness را نتیجه دهد

اقدامات انجام شده این دوره :

- دیدن فیلم های آموزشی فصل هشتم
- تمرین برنامه نویسی با actor model

برنامه اقدامات برای دوره بعد :

- تمرین پیاده سازی cluster
- شروع کار با Akka

مشکلات پیش آمده :

- نا توانی در پیاده سازی cluster
- عدم شناخت pipeTo که با مراجعه به مربی برطرف گردید
- عدم درک درست result aggregation که با مراجعه به مربی برطرف گردید

تاریخ : ۹۶/۰۵/۲۸ لغایت ۹۶/۰۶/۱

گزارش : ۷

توضیح مختصر آموزش این دوره :

در ابتدای دوره آموزشی akka در مورد مفهوم actor ها و مشکلات برنامه نویسی multi thread صحبت میکند که در قبل هم اشاره شد.در فیلم بعدی در مورد concurrency و parallelism صحبت میکند و بعد فرق بین async و sync را مطرح میکند که این موضوع نیز در قبل اشاره شدسپس race condition را مطرح میکند که به معنای این است چند منبع بخواهند همزمان به یک داده به اشتراک گذاشته شده دسترسی داشته باشند که موجب بروز اشکال است.در فیلم بعدی تمرینی ساده از actor model پیاده‌سازی شد.در فیلم بعدی در مورد actor system توضیح میدهد که درواقع شاخه اصلی actor ها میباشد و هنگام fail شدن actor ها آن‌ها را مدیریت میکند یا میتواند آن‌ها را configure کند و یا فعالیت‌های actor ها را log کندسپس کامپوننت های مختلف actor system از جمله user guardian actor , scheduler , configuration و... را معرفی مینماید . در فیلم بعدی روند نحوه کار یک actor را بررسی میکند که در ابتدا یک پیام به actorRef داده می‌شود سپس dispatcher صدا زده می‌شود و بعد این کامپوننت mail box را ران میکند و یک پیام را از آن استخراج میکند و تحویل actor میدهد باید توجه کرد که هر actor دقیقاً یک mail box دارد در فیلم بعدی در مورد props توضیح میدهد که درواقع یک configuration class هست برای اینکه بتواند optionهایی را هنگام ساخته شدن actor به آن اضافه کند. در فیلم بعدی در مورد فرق بین tell و ask صحبت میکند که وقتی از tell استفاده میکنیم دیگر به زمان آماده شدن جواب کاری نداریم اما وقتی از ask استفاده می‌شود جواب یک future است که میتوان آن را بلاک کرد. اما توصیه می‌شود که تا آنجایی که می‌شود از ask استفاده نشود. در فیلم بعدی در مورد supervision صحبت میکند که دارای دو استراتژی یک پدر در برابر فرزندان است :

- oneForOne

- oneForAll

در supervision در یک actor پدر میتوان اتفاقات فرزندان را مشاهده کرد مثلاً اگر یک فرزند در حالت restart قرار گرفت آنگاه پدر میتواند آن را بفهمد و برای این حالت تصمیم گیری کند اما در monitoring لازم نیست حتماً اتفاقات یک actor فرزند را مشاهده کرد بلکه میتوان actor های دیگر را نیز مشاهده نمود.در فیلم بعدی در مورد actorRef , actorPath صحبت میکند که قبلاً در مورد این موضوع نیز صحبت شد.در فیلم بعدی در مورد تغییر رفتار یک actor توسط become و unbecome صحبت شد.با وجود این متد ها میتوان actor را وارد یک state

مشخص کرد و در هر مرحله یک پیام از mail box خواند از طرفی stash را نیز معرفی میکند که یک‌جور صندوق موقت برای پیام‌هایی است که در حال حاضر actor نمیتواند به آن‌ها پاسخ دهد و میتوان در آن پیام‌ها را نگه داشت و در موقع لازم آن‌ها را unstash کرد .در فیلم بعدی در مورد persistence actor صحبت میکند که درواقع actor ای است که میتوان آن را recovery کرد و از لحاظ ظاهری اندکی با actor معمولی تفاوت دارد و دارای دو متد receive میباشد که یکی از آن‌ها مربوط به حالت recovery است.

اقدامات انجام شده در این دوره :

- دیدن بخش اول فیلم های آموزشی Akka
- تمرین برنامه نویسی remote

برنامه اقدامات دوره بعد :

- تکمیل مشاهده فیلم آموزشی Akka

مشکلات پیش آمده :

- فرق بین tell و ask

تاریخ : ۹۶/۰۶/۰۴ لغایت ۹۶/۰۶/۸

گزارش : ۸

توضیح مختصر آموزش این دوره :

در ابتدا توضیحی راجع به cluster داده می‌شود و ویژگی‌های آن از جمله اینکه میتواند fail شدن را تشحیص دهد و همچنین استفاده از پروتکول gossip را درون آن شرح میدهدسپس توضیح میدهد که cluster درواقع مجموعه‌ای از node ها میباشند که وضعیت عضو بودنشان در این مجموعه مشخص است سپس به معرفی seed node ها میپردازد که درواقع node هایی هستند که دیگران برای ورود به cluster به آن‌ها گوش میدهند . در فیلم‌های بعدی در مورد singleton cluster صحبت میکند و مزایا و معایب آن را بیان میکند.در فیلم بعدی در مورد cluster shrading صحبت میکند که به معنای cluster ای است که node ها در آن در سیستم‌های مختلف پخش هستند و هر گاه منابع زیادی را مصرف کردند این توزیع به صورت خودکار انجام میگیرد در ضمن در cluster shrading تمامی node ها با نام خود قابل دسترسی هستند.در فصل بعدی در مورد تست actor صحبت می‌شود و نحوه پیاده‌سازی آن توضیح داده میشوداین پیاده‌سازی را با ساختار it in و با testkit انجام میدهد و سپس با دستور expectMsg نشان میدهد که چه مقداری انتظار می‌رود تا تولید شود در فیلم بعدی هم در مورد تست کردن رابطه parent -child توضیح میدهد در فصل بعدی در مورد akka streamming صحبت میکند و حوزه استفاده آن را در پردازش داده‌های بزرگ ذکر میکندالبته در راستای streamming دو مشکل ممکن است به وجود بیاید که یکیblocking است و دیگری back presure میباشد در ادامه در مورد source and sink صحبت میکند که source درواقع مبدأ stream است و sink مقصد است و بخش flow را نیز معرفی میکند که مابین دو بخش ذکر شده است و عملیات بر روی داده‌ها در این بخش انجام می‌شود در فیلم بعدی در مورد تست stream ها صحبت میکندکه به سه صورت ساده و یا با استفاده از testkit و یا با استفاده از stream testkit میتواند انجام شود سپس در فیلم بعدی به معرفی graph میپردازد که درواقع همان flow است که میتواند چندین ورودی و چندین خروجی داشته باشد و با ترکیب کردن flow های مختلف میتوان عملیات های پیچیده‌ای بر روی ورودی ها انجام داده و خروجی های پیچیده نیز تولید کرد به همین منوال با تولید یک فایل sync میتوان از io اقدام به خواندن و نوشتن نیز کرد.

در فصل بعدی در مورد akka http صحبت میشود.همان طور که برای توزیع در thread ها actor ها پیشنهاد می‌شوند و برای اجرا در ماشین‌های متفاوت cluster مطرح می‌شود برای ارتباط با دنیای خارج نیز akka http مطرح میشود.یک ماژول akka http دارای کامپوننت های زیر می‌باشد :

- akka http core :که کارهای اصلی توسط آن انجام می‌شود مانند connection

- akka http

- akka http testkit

- akka http spray json

- akka http xml

در فیلم بعدی در مورد بخش client side صحبت میکند و موضوع را در سه سطح بررسی میکند :

- connection level

- host level

- request level

باید توجه داشت که بر مبنای اپلیکیشن باید یکی از سه سطح را انتخاب کنیم و معمولاً بیشتر برنامه‌ها در همان سطح request میمانند .

در فیلم بعدی در مورد server side صحبت می‌کند و آن را در دو سطح low level و high level بررسی میکند. Low level با http core در ارتباط است اما high level با akka http در ارتباط است در low level مباحثی همچون connection management و parsing and rendering مطرح است حال آنکه در high level مباحثی همچون routing مطرح می باشد.

اقدامات انجام شده این دوره :

- تمرین برنامه نویسی cluster
- اتمام فیلم های Akka

برنامه اقدامات دوره بعد :

- شروع پروژه و تکمیل Akka

مشکلات پیش آمده :

- مشکل خاصی در این دوره اتفاق نیفتاد

تاریخ : ۹۶/۰۶/۱۱ لغایت ۹۶/۰۶/۱۵

گزارش : ۹

توضیح مختصر آموزش این دوره :

در فصل نهم برخی از روش‌ها و الگو های متداول معرفی میگردد . اولین آن‌ها مربوط به balancing dispatcher هست.که در این مدل یک master خواهیم داشت و چندین worker که master سعی میکند به صورت متعادل کارها را بین worker ها پخش کند.در فیلم بعدی در مورد الگوی throttling messages صحبت میکند که البته این مبحث با نظر آقای جوان دیده نشد.در فیلم بعدی در مورد shutdown pattern صحبت میکنددر این مبحث مطرح میکند که اصل این الگو بر مبنای reaper actor هست که درواقع با ما بقی actor ها در ارتباط میباشد و هنگامی که همه actor ها به حالت پایانی خود رسیدند فرمان خاموش کردن actor system را میدهد. برای اینکار در ابتدا هر actor که به وجود می‌آید پیامی را به reaper ارسال میکند و reaper نیز او را watch میکند و هرگاه از actor پیام terminate ارسال شود آنگاه reaper او را از لیست actor های فعال خارج میکند و هنگامی که لیست actor های فعال خالی شد دستور خاموشی سیستم را میدهد.در فیلم بعدی در مورد ordered termination صحبت میکند. در این الگو علاوه بر ساختار master – worker یک terminator وجود دارد که درحال watch کردن master است و هر گاه master از بین رفت او تمام worker های او را ازبین می برد(البته درواقع حتماً لازم نیست که در اثر از بین رفتنmaster تمامی worker ها نیز از بین بروند بلکه میتوان برای آن worker ها یک master جایگزین کرد).در فیلم بعدی در مورد scheduling periodic messages صحبت میکندو نحوه پیاده‌سازی آن را توضیح میدهد که سعی کردم مدل آن را پیاده‌سازی کنم و خوشبختانه خروجی درست هم گرفتم.

در ادامه وارد پروژه میشویم:

طبق توضیحاتی که آقای جوان به بنده دادند نسیم یک سرور اصلی دارد که اسمش actor server است. در این پروژه سعی میکنیم که میزان بار تحمیلی به سرور را تست کنیم بدین صورت که کل پروژه تحت عنوان nasim-perf میباشد و این پروژه خودش دارای سرور و کلاینت میباشد و همچنین ماژول هایی در آن وجود دارد که از آن میتوان برای ارتباط با actor server استفاده کرد .برای تست میزان بار در nasim-perf کلاینت هایی از جنس actor ساخته می‌شود که در نقش کلاینت های اصلی در دنیای واقعی هستند همچنین یک actor به نام manager وجود دارد که میتواند با کلاینت ها ارتباط برقرار کند و به آن‌ها دستور دهد که از سرور اصلی چه درخواستی داشته باشند.ما برای تست بار باید کلاینت ها را مجبور کنیم تا پیام‌هایی را به سرور اصلی بفرستند و این اجبار از طریق manager انجام می‌شود بدین صورت که ما از طریق akka http با manager ارتباط برقرار میکنیم و به او میگوییم که به کلاینت ها چه دستوری بدهد و manager هم به کلاینت ها دستور میدهد تا آن‌ها با سرور اصلی

ارتباط برقرار کنند و از این طریق بار تحمیلی به actor server تست شود. حال باید توجه داشت که کلاینت ها برای ارتباط با سرور از RPC استفاده میکنند و request ها در قالب sdk ای ست که خود نسیم طرح کرده است و تحت عنوان ماژولی به نام nasim-mproto وجود دارد. از طرفی باید توجه کرد که کلاینت ها برای اینکه از طرف سرور به رسمیت شناخته شوند ابتدا باید با درخواست های validate, start و یا signup خود را در سرور رجیستر کنند تا سرور بتواند با آن ها ارتباط برقرار کند . در نتیجه سه دستور ابتدایی هر ارتباطی بین کلاینت و سرور درخواست های ذکر شده میباشد. در ادامه با نظر آقای جوان قرار بر این شد که ابتدا نگاهی به مطالب , log4j typesafe config , akka http بیندازم.

: Log4j

از سه کامپوننت اصلی تشکیل شده :

۱- loggers :مسیول گرفتن اطلاعات برای log کردن میباشد

۲-appenders ؛ مسیول انتشار لاگ ها در سیستم‌های مختلف

۳-layouts : مسیول فرمت بندی لاگ ها میباشد

هر یک از این کامپوننت ها درواقع object هستند که در دسته core object قرار میگیرند log4j دسته دیگری از اشیاء دارد که به نام support object معروف هستند و درواقع optional هستند.logger ها از سطوح مختلف لاگ همچون fatal , info , warn , ..استفاده میکنند . هر appender نیز ویژگی‌هایی دارد ک با آن‌ها رفتارش مشخص میگردد level , target , layout و ... . از طرفی layout ها هم میتوانند قالب‌های مختلفی همچون , HTML , XML pattern , Date و ... داشته باشند.

: Typesafe config

این مبحث تقریباً در فیلم‌های آموزشی نیز دیده شده بود درواقع میتوان یک فایل .conf درست کرد و config های مربوطه را در آن نوشت سپس در کد میتوان با استفاده از configFactory و توابع مربوطه به این configuration ها دسترسی پیدا کرد و در جایی که به عنوان پارامتر از آن استفاده می‌شود از آن استفاده کرد.

اقدامات انجام شده این دوره :

- تمرین پیاده سازی scheduling
- مشاهده آخرین بخش فیلم های Akka
- توضیحات اولیه پروژه توسط مربی

- مطالعه در مورد log4j و type safe

برنامه اقدامات دوره بعدی :

- شزوع کار پروژه اصلی

مشکلات پیش آمده :

- عدم درک درست ordered termination که با مراجعه به مربی برطرف گردید.

تاریخ : ۹۶/۰۶/۱۹ لغایت ۹۶/۰۶/۲۳

گزارش : ۱۰

توضیح مختصر آموزش این دوره :

در ابتدا مروری بر مبحث akka http انجام شد. Akka http خود دارای ماژول های مختلفی از جمله akka http core و akka http و akka http testkit میباشد.در فیلم بعدی در مورد client side صحبت میکند و سه سطح آنکه connection , host , request میباشد را معرفی مینماید.بیشتر برنامه‌ها در سطح request متوقف میشوند.پایین ترین سطح ، connection میباشد که اختیار کامل را فراهم میکند و سطح بعدی host است و در آخر هم‌سطح request قرار دارد سپس به پیاده‌سازی این سطوح میپردازد.در فیلم بعدی در مورد server side صحبت میکند که آن هم دارای دو سطح میباشد یکی low level که سطح اصلی و پایینی است و دیگری high level که سطح بالاتری است low level با akka http core ماژول در ارتباط است و high level با ماژول akka http در ارتباط میباشد.در low level بخش‌هایی مربوط به connection management , render and parsing , timeout management است و بخش‌های routing , file serving در بخش high level میباشد.سپس به پیاده‌سازی این موارد میپردازد.

در ادامه documentation هایی در مورد گیت و جیرا باید خوانده شود که توسط آقای جوان به این‌جانب تحویل داده شد.

اقدامات این دوره :

- مطالعه akka http
- خواندن documentation های جیرا و گیت

جمع بندی

در طول دوره کارآموزی آشنایی با مفاهیم و تکنولوژی های مفیدی حاصل شد. مفاهیمی همچون برنامه نویسی تابعی[۲۴] و massage passing مفاهیم ارزشمندی بودن که ارزش وقت گذاشتن را داشتند و آشنایی با آن ها حاصل شد. همچنین اشنایی با یک زبان جدید و قدرتمند به نام Scala نیز موقعیت خوبی بود و استفاده از فریم ورک Akka این فرصت را داد تا تسلط بیشتری بر برنامه نویسی وب سرویس ها حاصل گردد.

شرکت النون با رعایت قوانین مدیریت مشخص سعی بر منظم به پیش رفتن دارد که این خود درس بزرگی می باشد و در همین راستا آشنایی با ابزار های مدیریتی و کنترلی و مفاهیم مدیریت پروژه مانند اسکرام و کنترل کد مانند git فرصت بسیار خوبی بود. امید است که شرکت چکادنوآوران عصر اطلاعات با همین روند به جایگاه رفیع در عرصه کشوری و جهانی که به حق لیاقتش را دارند برسند.

[۲۴] Functional Programming

فهرست منابع

(n.d.).

Jovanovic, V. (2013). *The Scala Actors Migration Guide.* Chicago: Adventure Works Press.

Oracle. (2017, 10 13). Retrieved from About Java Technology: http://www.java.com/en/about/

Tolksdorf, R. (2013, June 14). *Programming languages for the Java Virtual Machine* . Retrieved from http://vmlanguages.is-research.de/

Ying, J. (2014). Hello World. *Scala*, 2.