معماری PVM

محدوده:

در این مستند معماری *PVM* با جزئیات مورد بررسی قرار میگیرد.

تاریخچه:

ردیف	نویسنده	تاریخ	شماره ویرایش	توضیحات
۱	تیم فنی و مهندسی	۹۱/۰۹/۱۵	۱٫۰٫۰	
۲	تیم فنی و مهندسی	۹۲/۰۵/۰۱	۳.۰.۰	
۳	تیم فنی و مهندسی	۹۴/۰۹/۱۴	۴.۰.۲	

فهرست مندرجات

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

فهرست تصاویر

فهرست جداول

فهرست ضمائم

دفتر مرکزی: یزد -خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

چکیده

مستند با ذکر مباحثی در اهمیت بحث طراحی آغاز میگردد. در ادامه کلیدهای اصلی در طراحی *PVM* مطرح میگردد و به دنبال آن نکاتی در مورد پیاده‌سازی *PVM*.
دلیل توسعه سامانه به جای ترجمه یا کار بر روی پروژه های متن باز و همچنین دلایل انتخاب *KVM* از دیگر مطالبی هستند که در این مستند پوشش داده شده است.

کلید واژه ها

PVM, Design

۱ مقدمه

انتخاب معماری و اجزای نرم‌افزار از مهمترین گامها در طی توسعه محصول میباشد چرا متضمن حیات نرم‌افزار و موفقیت آینده محصول از جهت پاسخگویی به نیازهای آینده و همچنین توان تغییر در مقابل درخواستهای ناگهانی میباشد.

مقرون به صرفه بودن نرم‌افزار وابسته به هوشمندی تیم توسعه آن به جهت استفاده از معماری منطقی و مناسب میباشد.

تیم شرکت پویش داده نوین بعد از قریب به ۱۵ سال تجربه کار در زمینه نرم افزارهای سیستمی و امنیتی بر بستر سیستم عامل لینوکس، چه از نظر توسعه و چه از نظر کاربری، بر اساس فرصت تشخیص داده شده اقدام به توسعه سامانه *PVM* نمود.

تجربه تیم در توسعه نرم افزارهایی همچون *IPTables-TNG*, *UTM* و همچنین کاربری نرم افزارهای امنیتی و شبکه در سطح وسیعی از شبکه‌های ادارات پشتوانه بسیار خوبی برای درک نیاز این‌گونه نرم‌افزارها و همچنین وضعیت موجود انتظارات از یک محصول زیرساختی میباشد.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

شرکت پویش داده نوین بر پایه تجارب ارزشمند خود توسعه *Framework*های مورد نیاز بـه جـای اسـتفاده و وابستگی به *Framework*های موجود را، از ابتدای حرکت در مسیر توسعه *PVM*، در برنامه خود داشته است. ایـن مهم برگرفته از تجارب موفق جهانی در شرکتهای بزرگ فناوری اطلاعات جهان و همچنین به دلیل تضمین چرخـه حیات طولانی نرم‌افزار و توان پشتیبانی از آن توسعه انجام شده است.

عمده دلیل در انتخاب این موضوع در بخش بعد با عنوان تجارب جهانی در حوزه رایانش ابری ذکر شده است.

۱.۱ تجارب جهانی در حوزه رایانش ابری

هم‌اکنون ارایه دهندگان و توسعه دهندگان بزرگی در حوزه رایانش ابری در حال فعـالیت هسـتند کـه از جملـه آن‌ها میتوان به *Amazon, Google, Facebook, Redhat* اشاره کرد.

به دلیل ماهیت کاری شرکت، تمامی نرم افزارهای متن باز موجود در حوزه رایانش ابـری بـه صـورت کامـل از منظر معماری و کاربرد مورد بررسی قرار گرفتند. همچنین تیم فنی شرکت تحقیق جـامعی در نحـوه کـار و توسـعه نرم‌افزارها در شرکتهای مطرح حوزه رایانش ابری انجام داده است که عمده خروجیهای این بررسیها به ایـن شـرح است:

- این شرکتها تمامی *Framework*های مورد نیاز خـود را بـه صـورت کامـل توسـعه میدهنـد و عمـده ایـن *Framework*ها به صورت متن باز ارایه نمیشوند (تقریباً تمام آنها)

  دلیل توسعه *Framework*ها عدم وابستگی به نرم افزارهای ثانویه و قـدرت انعطـاف در توسـعه قابلیتهـا و پاسخگویی به نیازها میباشد.
- پردازش تمامی اطلاعات وب در این شرکتها توسط برنامه‌هایی به زبـان *C* و یـا *C++* انجـام میشـود. ایـن شرکتها از *Web Server*ها به شیوه معمول استفاده نمیکنند. یکی از نکات بسیار مهم در این شرکتها جایگـاه ویژه زبانهای *C* و *C++* در توسعه نرم افزارهای سیستمی میباشد.
- صاحب تکنولوژی بودن برای این شرکتها بسیار حائز اهمیت است. به عنوان مثال شرکت ردهت بـرای کـار در حوزه مجازی سازی شرکت دارنده *KVM* را خریداری کرد. این موضـوع بـدین معنـا اسـت کـه بـرای دستیابی به خروجی مناسب باید تا حد امکان صاحب *Framework*ها و تکنولوژیهای مورد نیاز باشید.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

- تمام این شرکتها از نحوه توسعه به صورت متدولوژی متن باز استفاده میکنند. این‌گونه بدان معنی است که شما از نرم افزارهای پایه و کاملاً متن باز استفاده میکنید و اقدام به توسعه *Framework*های مدیریتی یا تکنولوژیهای خاص خود میکنید.
- در فضای متن باز هیچگونه محصول آماده و مجانی برای کار در محیطهای *Enterprise* وجود ندارد. اکثر نرم افزارهایی که به صورت متن باز و مجانی ارایه می‌شوند به صورت یک *Prototype* عرضه می‌شوند که علاوه بر استخراج نیازهای محصولات شرکتهای دارنده تکنولوژی، محمل بسیار خوبی برای *Bug Fix* و درک نیاز کاربران میباشد.
- بدون توسعه زیرساختهای مورد نیاز در محصول خود، قابلیت پاسخگویی به نیاز کارفرمایان و پشتیبانی از محصول در طولانی مدت برای دارنده محصول فراهم نخواهد شد.

جمیع موارد ذکر شده و تجارب شرکت در توسعه و کاربری نرم افزارهای سیستمی پشتوانه انتخاب مسیر توسعه *PVM* و تکنولوژیهای مورد استفاده در این مسیر بوده است.

شرکت پویش داده نوین اعتقاد دارد ارایه پشتیبانی مناسب به کارفرمایان و همچنین تضمین حیات طولانی مدت محصول بدون توسعه امکان‌پذیر نمیباشد.

در ادامه به بررسی جزئیات بیشتر در اجزای ساختار *PVM* و فلسفه وجودی و کاربرد آن‌ها میپردازیم.

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۲ معماری *PVM*

در معماری *PVM* سه اصل مورد توجه بوده است:

- توسعه *Framework*های مورد نیاز
- *Modular* بودن و قدرت و انعطاف مناسب در توسعه
- سادگی و روانی و عدم وابستگی به نرم افزارهای ثانویه

اجزای اصلی سامانه *PVM* به شرح زیر میباشند. قابل ذکر است ترتیب ذکر شده از پایه‌ای ترین لایه به سطوح بالاتر میباشد:

- *PParam*
- *Putil*
- *Sball*
- *PVM Modules*

در ادامه به بررسی دقیقتر این اجزا میپردازیم.

۲.۱ زبان برنامه نویسی

در *PVM* از چند زبان برنامه نویسی بر اساس نیازهای توسعه استفاده میشود:

- زبان *C++* که برای توسعه هسته (*PVM Core*) در سیستم عامل لینوکس استفاده شده است. عمده مباحث فنی و توسعه ای *PVM* و نکات اصلی در معماری *PVM* در این لایه میباشد.

  قابل ذکر است این زبان در رنکینگ زبانهای برنامه نویسی دنیا در رده ۵ زبان برتر دنیا قرار دارد و به دلیل ماهیت شیءگرایی و همچنین سرعت اجرای آن، بسیار مورد توجه قرار گرفته است.
- *Bash* که در سمت هسته برای توسعه برنامه‌های مکمل کوچک همچون سرویسها مورد استفاده قرار گرفته است.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

- *C#* برای توسعه رابط کاربری ویندوزی استفاده میشود.
- *Python* برای توسعه رابط تحت وب *PVM* مورد استفاده قرار گرفته است. این زبان نیز در رده ۵ زبان برتر دنیا قرار دارد.

۲.۲ PParam

کتابخانه *Pparam* پایه‌ای ترین کتابخانه در *PVM* میباشد که به صورت متن باز در *github* ارایه شده است. در ابتدای شروع به توسعه *PVM* دغدغه اصلی استفاده از پروتکل و فرمت ارتباطی مناسب بین نودهای کلاستر و همچنین ارتباط کاربر با آن بود.

آنچه که هم‌اکنون بسیار مرسوم است استفاده از *XML* و *JSON* میباشد. ما به دنبال شیوه ای مناسب جهت تبدیل اشیاء و کلاسها در زبان *C++* به *XML* و یا *JSON* بودیم و به طور کلی به دنبال لایه‌ای که ذغدغه های تبدیل فرمت اطلاعات را به عهده بگیرد. اولویت یک در این لایه سبکی و سادگی آن بود.

بعد از بررسیهای گوناگون اقدام به توسعه کتابخانه *Pparam* نمودیم. این کتابخانه که منطبق با استانداردهای زبان *C++* توسعه یافته است، ساختار تعریف پارامترهای داده‌ای را مشخص میکند.

با این کتابخانه برنامه نویس قادر است به راحتی شیء خود را تعریف کرده و آن را به فرمتهای داده‌ای مختلف تبدیل کند و یا اینکه از فرمتهای داده‌ای مختلف شیء خود را بارگزاری کند.

توسعه پارامترهای گوناگونی در این کتابخانه در قالب *sparam* علاوه بر غنی کردن این کتابخانه در جهت پشتیبانی ازفرمتهای داده‌ای گوناگون، صحت داده‌ای این پارامترها را نیز تضمین میکند. از جمله این پارامترها عبارتند از:

- *IP4Param*
- *IP6Param*
- *MACParam*
- *BoolParam*
- ...

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

```
class Server : public XMixParam
{
public:
        Server() :
                XMixParam("server"),
                ip("ip"),
                uptime("uptime", 0, -1),
                cpuUsage("cpu_usage", 0, 100),
                ramUsage("ram_usage", 0, 100)
        {
                addParam(&ip);
                addParam(&uptime);
                addParam(&cpuUsage);
                addParam(&ramUsage);
        }
        bool key(string &_key)
        {
                _key = ip.value();

                return true;
        }

        IPxParam                ip;
        XIntParam<XULong>       uptime;
        XIntParam<XFloat>       cpuUsage;
        XIntParam<XFloat>       ramUsage;
};
```

تصویر ۱: نمونه ای از کلاس توسعه یافته با *PParam*

بر اساس استاندارد توسعه یافته در *Pparam* شیء ویژه ای به نام *Xobject* توسعه پیدا کرده است. در فضای *Xobject* تمامی اشیاء موجود در مجموعه قادر خواهند بود با یکدیگر اتصالاتی (*Connection*) برقرار کنند. این اتصالات دنیایی از اشیاء و اتصالات آن‌ها را فراهم میکند.

Xobject مبنای(والد) و به تبع آن *Xparam* والد تمامی اشیاء در *PVM* میباشند که این موضوع قاعده ارث بری و شیء گرایی در *PVM* را به شکلی یک‌دست و همگن موجب شده است.

آگاهی به قواعد *Xparam* و *Xobject* برنامه نویس را قادر میسازد حجم بسیاری از قواعد حاکم بر *PVM* را درک نماید.

این یک قاعده در *PVM* است که قابلیتهای توسعه به صورت درختی از ریشه‌ای ترین کتابخانه تا لایه‌های بالاتر توسعه میابد. ریشه این درخت *Pparam* و به دنبال آن *Xobject* میباشد.

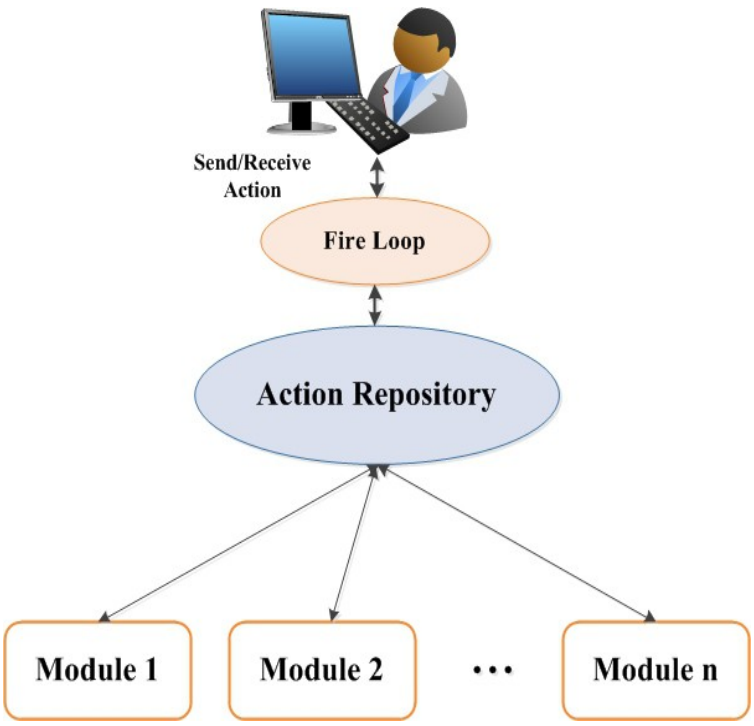
دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

در هر کدام از این کتابخانه‌های قواعدی برای صحت اشیاء و پارامترهای داده‌ای تعریف می‌شود که به تبع آن توسعه در سطوح بالاتر را راحتتر میکند.

۲.۳ *Putil*

این کتابخانه بر پایه *Pparam* با هدف ایجاد قابلیت توسعه *Modular* سامانه *PVM* پیاده‌سازی گردیده است. با استفاده از این کتابخانه، افزودن ماژول جدید به سامانه نیازمند تغییراتی در هسته برنامه به منظور پذیرش آن نیست.

نمایی از اجزای این کتابخانه به این شرح است:



تصویر ۲: نمایی از کتابخانه *putil*

قابلیتهای ماژولها در قالب *Action* در سامانه *PVM* معرفی میشوند. *Action*ها در اصل *API*های هر ماژول میباشند. کاربر قادر خواهد بود با فراخوانی *API*ها از طریق *putil* اقدام به تعامل با آن ماژول نماید.

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

- *Modules*
  - نماینده ماژولهای موجود در *PVM* میباشند.
  - هر ماژول در *PVM* با یک *ID* یکتا که از صفر شروع میشود، مشخص میشود.
- *Action Repository*
  - یک *Action* عملیاتی است که کاربر میتواند به واسطه آن با سیستم(ماژول) تعامل کند.
  - هر ماژول مجموعه‌ای از *Action*ها را در سیستم مشخص میکند که این *Action*ها هر کدام با یک *ID* که از صفر شروع میشود مشخص میشوند.
  - ترکیب *ID* مربوط به یک ماژول و *ID* مربوط به هر کدام از *Action*های آن ماژول یک عدد یکتا برای *Action* ایجاد میکند. به عبارتی هر *Action* با *ID* مربوط به ماژول و *ID* مربوط به خودش مشخص شده و صدا زده میشود.
  - *Action Repository* به صورت یک انباره مجتمع تمامی اطلاعات *Action*های موجود را میزبانی و امکان فراخوانی آن‌ها را فراهم می آورد.
- *Fire Loop*
  - وظیفه آن تعامل با کاربر و دریافت دستورات وی و اجرای *Action*های مورد نظر کابر میباشد.
  - هم‌اکنون *Fire Loop* تمامی دستورات کاربر را از طریق *UNIX Socket* و *TCP Socket* دریافت میکند.

چنانچه مشاهده میفرمایید تمامی امکانات تعاملی سامانه *PVM* با استفاده از *putil* فراهم شده است که قابلیت بسیار مناسبی به جهت اعمال سیاستهای تعاملی مورد نیاز، به صورت متمرکز، را فراهم می‌آورد.

همچنین این کتابخانه میتواند درگاههای ارتباطی متعدد مانند *UNIX Socket*و *TLS*و *TCP Socket* و هر نوع درگاه ارتباطی مورد نیاز را پیاده‌سازی نماید. این پیاده‌سازی کاملاً مستقل از کتابخانه‌های بالادستی *PVM* انجام می‌شود و از این منظر قدرت انعطاف مناسبی را ایجاد میکند.

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۲.۳.۱ ارسال دستورات به *PVM*

ارسال دستورات از طریق ارسال یک رشته به فایل */vat/run/sballd.sock* انجام میشود. فرمت رشته ارسالی که یک رشته *XML* است به این شرح است:

```
<cmd>
    <sid/>
    <cid/>
    <params/>
</cmd>
```

sid: مشخص کننده *sub-system id* یا همان *module-id* میباشد

cid: مشخص کننده *command-id* یا همان *action-id* میباشد

params: دربرگیرنده پارامترهای مورد نیاز *action* مورد نظر میباشد.

۲.۳.۲ پاسخ به دستورات ارسالی

PVM با ارسال یک رشته *XML* به دستور ارسالی پاسخ میدهد. فرمت این رشته به این شکل است:

```
<response>
    <status/>
    <description/>
</response>
```

تگ *status* وضعیت پاسخ را نشان میدهد که مقادیر آن عبارتند از: *success, failed, warning*

warning وضعیتی را نشان میدهد که دستور به صورت ناقص اجرا شده است، *success* اجرا موفقیت آمیز بوده است و *failed* اجرا به صورت کامل ناموفق بوده است.

Description شامل جزئیات پاسخ دستور ارسالی میباشد.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۲.۴ Sball

اسبال بزرگترین کتابخانه *PVM* میباشد که وظیفه مهم *Cluster Engine* یا موتور همروندی در *PVM* را بر عهده دارد. قابل ذکر است در اوایل توسعه *PVM* از محصولات متن باز در این لایه استفاده میشد که به دلیل بروز مشکلات متعدد و باگهای گوناگون در این محصولات اقدام به توسعه حساسترین و مهمترین لایه نرم‌افزار یعنی *Cluster Engine* نمودیم.

وظایف اسبال در *PVM* به این شرح است:

- انتخاب *Master* در کلاستر به منظور انجام فعالیتهای مدیریتی در کلاستر
- به هنگام سازی اطلاعات اشیاء (اطلاعات ماشینهای مجازی، سوییچهای مجازی و ..) در تمامی نودهای کلاستر
- آگاهی از وضعیت نودهای کلاستر و تشخیص وضعیت جاری آن‌ها
- مدیریت ماژولهای *PVM*

این ماژول از دو ماژول زیرمجموعه به شرح زیر تشکیل شده است:

- انباره اشیاء *Sball*: که انباره کلاستری اشیاء موجود در *PVM* میباشد. این ماژول از جمله مهمترین ماژولهای و وظایف *Sball* میباشد
- انباره مدیریت کلاستر: مدیریت وضعیت نودها و پیامرسانی بین آن‌ها را بر عهده دارد.

۲.۵ انباره اشیاء

از جمله مهمترین وظایف اسبال مدیریت انباره توزیع شده از اشیاء در سطح کلاستر میباشد. در *PVM* همه چیز در قالب اشیاء تعریف و مدیریت می‌شوند که از آن جمله میتوان به ماشین مجازی، سوییچ مجازی، زمان، سایت پشتیبان و .. اشاره کرد.

برای پیروی مناسب از قاعده شیءگرایی، بر اساس استاندارد *Pparam* و *Xobject* شیء *SballObject* توسعه یافته است. این شیء والد تمامی اشیاء در *PVM* میباشند که تمامی آن‌ها در قالب انباره اشیاء در *PVM* و در سطح

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

کلاستر نگهداری میشوند.

اسبال وظیفه دارد این انباره را در سطح کلاستر با یکدیگر همگام نگه دارد.

```
/**
 * \class VM
 * define and manage virtual machine parameters.
 */
class VirtualMachine : public SballObject
{
        friend class VMMngrModule;
public:
        typedef VirtualMachineStatus    Status;
        typedef VMMigrationParam        Migration;
        /**
         * \typedef StatusVector
         * Defines a vector of virtual machine statuses.
         */
        typedef vector<Status::Status>  StatusVector;

        /**
         * \enum VMCommnad
         * Virtual machine monitor commands.
         */
        enum VMCommand {
                VMCMD_CAPABILITIES,
                VMCMD_POWERDOWN,
                VMCMD_RESET,
                VMCMD_PAUSE,
                VMCMD_RESUME,
                VMCMD_MIGRATE,
                VMCMD_MIGRATE_CLIENTINFO,
                VMCMD_QUIT,
                VMCMD_MAX,
        };

        VirtualMachine();
        virtual void type(Type &t) const;
        /**
         * Set Virtual machine console port if the port has been not set.
         */
        void set_consolePort(XUInt port);
        XUInt get_consolePort() const;
        /** return hypervisor value. */
        string get_hv();
        Status::Status get_status();
```

تصویر ۳: نمایی از شیء *VirtualMachine* بر اساس *SballObject*

۲.۵.۱ *Sball Object*

ساختار *XML* شیء اسبال به این شرح است.

<sball_obj>

<uuid/>

<name/>

<type/>

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

// Object Special sata

</sball_obj>

تگ *uuid* که مخفف *Universally Unique Identifier* میباشد، به عنوان کلید *Object* در تمام *PVM* مورد استفاده قرار میگیرد. به عبارتی تمام ارجاعات به *Object*های مورد نیاز توسط *uuid* که کلید است انجام میپذیرد. این فیلد در تمام طول عمر *Object* ثابت است.

تگ *name* نام موجودیت را تعیین میکند که در چرخه زندگی *Object* قابل تغییر است و راه ارتباطی قابل تفسیر توسط کاربر با *Object* میباشد.

تگ *type* نوع *Object* را تعیین میکند.

فیلدهایی داده که در ادامه می‌آیند بستگی به نوع *Object* دارد(*type*) و برای هر *Object* به صورت جداگانه تعریف میگردد.

همچنین قابل ذکر است که تگهای ابتدایی و انتهایی اشیاء بر اساس نوع شیء میتوانند متفاوت باشند.

۲.۵.۲ *Object List*

یک لیست از *Object*ها است. هر لیست دارای یک نام یکتا در *PVM* میباشد. چنانچه لازم باشد تا لیست تمامی *Object*های مربوط به لیست در اختیار کاربر قرار گیرد ساختار *XML* مربوطه به این شرح میباشد:

<pvm_object_list_name>

<sball_obj/>

.../

</pvm_object_list_name>

pvm_object_list_name نام یکتای لیست میباشد.

۲.۵.۳ *Object Repository*

مجموعه تمامی *Object List*های موجود در *PVM* توسط *Object Repository* مدیریت میشوند. هر مأژول

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

در *PVM* میتواند بر اساس نیاز خود *Object List*هایی را در *Object Repository* رجیستر کند.

این انباره در سطح کلاستر مدیریت میشود.

۲.۶ توسعه ماژول در *PVM*

توسعه ماژول در *PVM* بر اساس قواعد استاندارد موجود در کتاخانه های بالادستی به شرح زیر انجام میشود:

- *Pparam*
- *Putil*
- *Sball*

هر ماژول اقدام به توسعه اشیاء خود که از *SballObject* ارث برده‌اند میکند و قابلیتهای موردنیاز کاربران را در قالب *Action*ها توسعه میدهد.

سپس این اشیاء و لیستهای مورد نیاز جهت نگهداری آن‌ها در *Sball* رجیستر می‌شوند و همچنین *Action*در *putil* رجیستر میشوند.

از آن به بعد اسبال وظیفه مدیریت و همگام سازی اشیاء در سطح کلاستر را بر عهده خواهد گرفت و همچنین کاربر قادر خواهد بود با فراخوانی *Action*ها اقدام به تعامل با ماژول مورد نظر نماید.

به عبارتی توسعه ماژول در *PVM* به توسعه اشیاء بر اساس قواعد *SballObject* و همچنین *Action*های مورد نیاز خلاصه شده است و این به دلیل روح حاکم بر معماری *PVM* به دلیل ایجاد قواعد درختی بر اساس ساختار وابستگی درختی بین کتابخانه‌ها از پایه‌ای ترین کتابخانه تا کتابخانه‌های بالادستی میباشد.

قابل ذکر است که توسعه ساختار ماژول نیز بر اساس قواعد حاکم بر اشیاء اسبال و با ارثبری از *SballModules* انجام میشود.

در ادامه نمایی از کد *SballModule* قابل مشاهده است.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

```
 * \class Module
 * Sball Module definition.
 *
 * Sball could grow by developing new modules for him as ".so"
 * Sball interface with modules is "SballModule".
 * Eatch module should implement inherited class from "SballModule".
 * load/unload/save/createBackup/restoreBackup would be implemented
 * by mudules.
 * Each module should have "getModuleObj" in this form:
 *      static SballModule *getModuleObj().
 * After loading of ".so" file, sball would search this function to achive
 * pointer to module object.
 * This object would be sball interface with module.
 */
class SballModule : public SballObject
{
public:
        typedef SballObject::Object     Object;
        typedef SballObject::ObjectList ObjectList;
        typedef SballObject::iterator   iterator;

        SballModule();
        virtual int getID() const;
        void *get_moduleHandle();
        void set_moduleHandle(void *_moduleHandle);
        /**
         * Returns module version in sepration of major, minor, micro.
         */
        void get_version(XUInt &major,XUInt &minor,XUInt &micro);
        /**
         * Returns module version in "major.minor.micro" format.
         */
        string get_version();
        /**
         * Set module version.
         */
        void set_version(const XUInt major,
                         const XUInt minor, const XUInt micro);
        string get_name();
        void set_name(const string &name);
        string get_rdate() const;
```

تصویر ۴: نمایی از *SballModule* و مستندات موجود در کد

چنانچه ملاحظه میگردد قاعده ارثبری از *SballObject* در کل اجزای *PVM* رعایت شده است و این موضوع همبستگی معماری *PVM* و همچنین قابل توسعه مدیریت شده و سریع در آن را فراهم میکند.

اینگونه طراحی علاوه بر افزایش کیفیت کد کاهش هزینه‌ها در توسعه را نیز به همراه دارد.

یکی از وظایف اسبال در حوزه ماژولها بررسی تطابق ویرایشهای مختلف ماژول موجود در سطح کلاستر میباشد. به عبارتی اسبال کنترل میکند که نسخه های مختلف از یک ماژول که ممکن است بر روی نودهای یک کلاستر هستند با یکدیگر تطابق دارند و یا خیر؟ در صورت عدم تطابق نودی که دارای عدم تطابق هست را از کلاستر خارج میکند تا از بروز مشکلات برای سیستم جلوگیری کند.

در عکس فوق نمایی از نحوه مستندسازی(*Comment*) کدهای *PVM* نیز قابل مشاهده است.

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

```
class VSMModule : public SballModule
{
public:
        enum objectListID {
                VSMOL_VIRTUALSTORAGE    = 110,
                                //SballObjRepoManager::ORMOL_MAX,
                VSMOL_VIRTUALDISK,
                VSMOL_ISOIMAGE,
                VSMOL_MAX
        };

        VSMModule();
        virtual int getID() const;
        virtual void load() throw (Exception);
        virtual void save();
        virtual void createBackup();
        virtual void restoreBackup(const string &backupID);
        virtual void unload() throw (Exception);
        static bool hasContentConflicting(const int storageType,
                                          const int diskType,
                                          const int contentType);

private:
        static const int                contentConflictTable[VIRTUALDISK_NO][
                                                VIRTUALSTORAGE_NO];
        static const string             objectListName[VSMOL_MAX -
                                                VSMOL_VIRTUALSTORAGE];
        static SballLogSystem           logs;
        /** Pointer to the list of virtual Storages */
        static ObjectList               *virtualStorageList;
        /** Pointer to the list of virtual disks */
        static ObjectList               *virtualDiskList;
        /** Pointer to the list of ISO images */
        static ObjectList               *isoImageList;
        static VSMModuleActionList      actionList;
};

extern  VSMModule       *vsmModuleObject;

} // namespace pvmm

#endif  //  PVM VSM HPP
```

تصویر ۵: نمایی از کد ماژول مدیریت استوریج

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۰۳۵۱ - ۷۲۶۵۳۴۳ فکس: ۰۳۵۱ - ۷۲۴۵۰۸۷
www.pdnsoft.com Email: info@pdnsoft.com

۳ پیاده‌سازی *PVM*

چنانچه در بخش قبل اشاره شد زبان *C++* به عنوان زبان اصلی برنامه نویسی در *PVM* میباشد. در ادامه جزئیاتی از تکنیکهای به کار رفته در پیاده‌سازی اجزای *PVM* مورد بررسی قرار میگیرد.

۳.۱ ارتباط با ماشین مجازی

جهت ارتباط با ماشین مجازی از پروتکل *QMP* استفاده میشود. این پروتکل که بر اساس *JSON* میباشد مبنای فعالیتهای مدیریتی ماشین مجازی میباشد.

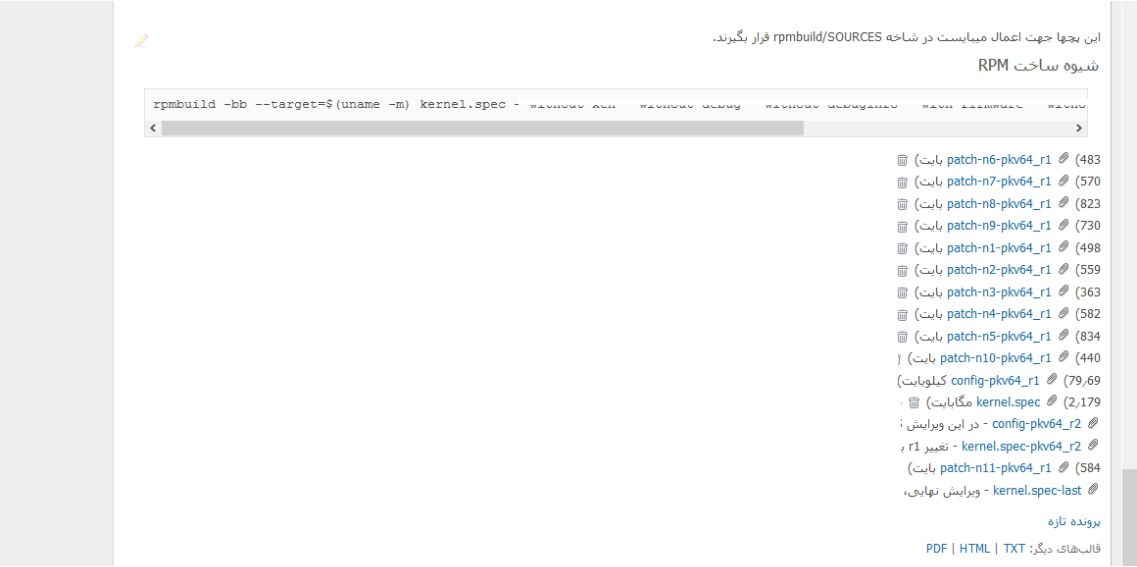
این پروتکل در ویرایشهای مختلف ساختار *KVM* ثابت میباشد و در هر ویرایش نسبت به ویرایش قبلی تواناییهایی اضافه میشود.

۳.۲ کامپایل هسته

هسته سیستم عامل *PVM* بر اساس هسته های ردهت میباشد و بر اساس نیازمندهای ماشینهای مجازی و به جهت دستیابی به کارآیی بالا تنظیم و کامپایل مجدد شده است.

تمامی فعالیتهای شرکت بر روی این هسته به راحتی قابل انتقال به ویرایشهای بالاتر هسته لینوکس میباشد.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com



تصویر ۶: نمایی از مستندسازی نحوه کامپایل کرنل

۳.۳ مستندسازی

در شرکت پویش داده نوین مستندسازی از اصول اساسی و پایه محسوب میشود. تمامی ماژولهای *PVM* در سطوح طراحی و پیاده‌سازی دارای مستندسازی با جزئیات مناسب میباشند.

سامانه مدیریت پروژه در شرکت محور تمامی فعالیتها در مسیر توسعه محصول میباشد. همچنین از ابتدای توسعه *PVM* از نرم‌افزار *LibreOffice* (پروژه متن باز جایگزین *MSOffice*) به منظور مستند سازی استفاده میگردد.

۳.۴ سیستم عامل پایه

از سیستم عامل *Redhat* به عنوان سیستم عامل پایه در ساخت *PVM OS* استفاده شده است. این موضوع قابل توجه است که ساختار توسعه یافته توسط *PVM* وابستگی به سیستم عامل خاصی ندارد.

به عبارتی ما قادر هستیم که *PVM* را بر روی هر نسخه لینوکسی برپا کنیم، البته دلیل انتخار *Redhat* پشتیبانی

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

سطح حرفه‌ای است که این سیستم عامل دارد.

۳.۵ سبک کد نویسی یا *Coding Style*

تیم توسعه *PVM* از استانداردهای موجود در توسعه محصولات متن باز همچون هسته لینوکس بهر گرفته و سبک کد نویسی حرفه‌ای برای توسعه خود انتخاب کرده است.

نمونه از نمونه کد نویسی در پروژه *Pparam* که به صورت متن باز عرضه شده است به خوبی قابل مشاهده است.

۳.۶ استانداردهای خط تولید

مسیر توسعه *PVM* در شرکت به شکل یک خط تولید مدیریت میشود. تمامی ملزومات این خط تولید به شکل استاندارد درآمده اند که از جمله مواردی که در دانش سرای شرکت منتظر شده است میتوان به مستند «شیوه تنظیم شماره نسخه (*Version*) در توسعه نرم‌افزار» اشاره کرد.

۳.۷ دانشسرای پویش

طبیعت هر کار توسعه ای زایش دانش و اطلاعات جدید میباشد. این امر به نوبه خود در شرکت پویش داده نوین اتفاق افتاده است و دانش شرکت در قالب «دانشسرای پویش» عرضه شده است.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۴ چرا تولید به جای فارسی‌سازی محصولات موجود

این سؤالی است که بعضا از تیم پویش داده نوین پرسیده میشود. پاسخ و بررسی این پرسش از چند منظر قابل بررسی است:

۴.۱ از منظر تیم توسعه دهنده (شرکت ارایه کننده محصول مجازی‌سازی)

ارایه محصول بر اساس محصولاتی اینچنین و در حد ترجمه رابط کاربری، باعث افزایش هزینه‌های پشتیبانی و توسعه ی تیم توسعه دهنده میشود. این موضوع به این دلیل است که روند توسعه محصول و ارایه *Bug Fix*ها خارج از کنترل تیم توسعه بوده و هر گونه تغییر در روال پروژه اصلی، موجب ایجاد تنش و هزینه برای تیم تولید و پشتیبان میشود. این پروژه‌ها معمولا با ارایه انواع لایسنسها (*Free, Enterprise, Commercial*) موجب هدررفت زمان و گیج شدن تیم توسعه دهنده و پشتیبانی میشوند.

این موضوع در صورتی قابل تحمل است که تیم توسعه قوی دارای تسلط کافی بر کد محصول وجود داشته باشد که هزینه این موضوع با تولید محصول جدید برابری میکند.

موضوع قابل تامل دیگر در این مسیر نحوه برخورد با موارد خطا و اشکالات به وجود آمده است، در این موارد به دلیل استفاده از انبوهی از پروژه‌ها که بعضا با اهداف و کاربریهای گسترده نوشته شده و در کاربریهای محدودی استفاده میشوند، روال اشکال یابی خطاهای به وجود آمده(در سایت کارفرما و یا در مرحله تست) بسیار زمان‌بر و در بعضی موارد خارج از کنترل میباشد. این موضوع را با زمانی مقایسه کنید که شما تا حد امکان به نرم افزارهای سیستمی نزدیک شده‌اید و بروز هر گونه مشکل در سیستم به راحتی قابل رهگیری و رفع است.

از جمله مواردی که در مسیر توسعه *PVM* با آن روبرو شدیم داستان *RGManager* و *Pacemaker* بود. در ابتدا *PVM* بر اساس *RGManager* توسعه پیدا کرد، به عنوان لایه مدیریت *HA*، که در محیطهای عملیاتی دارای *Bug*های فراوانی بود(که البته تمامی نرم‌افزارهای متن‌باز و مشابه از آن استفاده میکنند). بعد از مدتی متوجه شدیم که روال توسعه این پروژه به کل متوقف شده‌است و هم‌اکنون تمرکز اصلی بر روی *Pacemaker* است. هزینه انتقال

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

از یک نرم‌افزار به نرم‌افزار دیگر واقعا سرسام آور است، بحث تست و اطمینان کامل جهت انتقال به سایت مشتری را برای مشتریان بزرگ در نظر بگیرید.

با توجه به اینکه مسیر توسعه ما تکمیل ماژول *HA* در *Sball* بود به جای انتقال به سمت *Pacemaker*، ماژول *Sball-HA* تکمیل و به کار گرفته شد که خروجی بسیار بهتر و قابل اتکایی دارد. یکی از مزیتهای *HA* در *Sball* نسبت به *Pacemaker* آن است که در *Pacemaker* بحث *HA* کاملا عام دیده شده است ولی در *Sball* ملزومات و حساسیتهای ماشینهای مجازی به صورت خاص مورد توجه قرار گرفته است.

قانون کلی در توسعه محصول به صورت متدولوژی متن‌باز آن است که تا حد امکان از پروژه‌های کاملا متن باز (کتابخانه‌ها، هسته و ...) استفاده گردد و از وابستگی به پروژه‌هایی که در طول زمان دستخوش تغییرات(منظور تغییرات استراتژیکی و فنی) میگردند جلوگیری شود، کاری که در توسعه *PVM* کاملا مورد توجه قرار گرفته و جوابگو نیز بوده است.

این سیاست در بلند مدت محصولی پایدار و با پشتیبانی عالی را موجب میشود.

بحث آینده مجوز نرم‌افزار بسیار مهم است، حوزه مجازی‌سازی حوزه بسیار حساس با پشتیبانی بالا است، چنانچه نرم افزار در مسیر توسعه تغییر قالب دهد و یا اینکه صرفا کدهای تست نشده را جهت *Bug Fix* به صورت *Community* عرضه کند، عرصه محصول در این فضا بسیار پر چالش و هزینه بر خواهد بود و به نوعی آبروی شرکت را در معرض خطر قرار میدهد.

موضوع بعدی حاصل تجارب چند ساله ما در حوزه متن‌باز است. اگر نگاهی به گستره نرم‌افزارهای متن باز در کل جهان داشته باشید، حوزه *Community* در حال حرکت به سمت بستر تست(*Test*) است. به عبارت دقیق‌تر شرکتهای تولید کننده بعد از ارایه کدها و محصولات در بستر *Community* و گرفتن خروجیهای تست از کاربران کل دنیا، نسخه‌های *Enterprise* خود را روانه بازار میکنند.

در این فضا تولید نرم‌افزار و ارایه به صورت متن‌باز و دست و پنجه نرم کردن با دنیای متن‌باز وجهه مناسبی از شرکت در سطح تکنیکال به جهان ارایه میدهد و این امکان را فراهم خواهد کرد تا شرکت تولید کننده قادر باشد از امکانات *Community* ها به نحو مطلوب در جهت توسعه محصول خود استفاده کند.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

به عبارتی توسعه(تولید) محصول در دنیای متن‌باز در طولانی مدت باعث کاهش هزینه و تقویت برند (*Brand*) میشود در صورتیکه در صورت صرفا استفاده از نرم‌افزارهای متن‌باز شرکت تبدیل به *Test Case* تولید کنندگان و افزایش هزینه‌های سالانه و تضعیف *Brand* میشود.

ارایه محصول بر اساس محصولات دیگر و صرفا به صورت ترجمه، حوزه رقابت را گسترده و مزیتهای رقابتی را از میان میبرد. تولید محصول امکان ارایه مزیتهای رقابتی برای خارج از ایران را نیز فراهم می‌آورد.

۴.۲ حساسیت حوزه مجازی‌سازی

حوزه مجازی‌سازی در سازمانها به دلیل تشکیل بستر فناوری اطلاعات سازمان حوزه حساس و پرچالشی است، در بازدیدهایی که از سازمانها داشتیم به مواردی برخوردیم که از اجرای ماشینهای حساس بر فراز *VMWare* خودداری کرده بودند، که مهمترین چالش آنها آینده محصول مجازی‌سازی، پشتیبانی و امنیت بود.

این موضوع مؤید این است که تیم ارایه دهنده راهکار میبایست از لحاظ فنی و تکنیکی تسلط کامل بر ساختار محصول ارایه شده داشته باشد تا قادر باشد تمامی زمینه‌ها و جوانب محصول خود را تضمین و گارانتی کند. این موضوع بحث قدرت تکنیکال و مانور تیم توسعه و پشتیبانی را در حد زیادی روشن میکند.

۴.۳ از نگاه کارفرما

یکی از این موارد در بند ۲ مطرح شد. بحث پشتیبانی و میزان تسلط تیم توسعه بسیار مهم است. همچنین از نگاه کارفرمای ایرانی تیم توسعه داخلی و ایرانی میتواند قابل توجه باشد.

از جمله مزیتهای دیگر توسعه محصول قابلیت سفارشی سازی آن است. *PVM* بر اساس نیازهای محلی توسعه یافته است که از جمله این موارد قابلیت *mirror site* و بحث *Backup OffLine* است که به واسطه نیازهای سازمانی داخل ایران توسعه یافته است.

این امکان وجود دارد تا درصورت اعلام نیازهای ویژه در سطح *Enterprise* موارد در *PVM* اعمال گردند. تولید محصول قابلیت مانور و پاسخگویی سریع به نیازهای کارفرما را موجب میشود که در طول حیات محصول افزایش رضایت مشتری را به دنبال دارد.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۴.۴ سخن پایانی در پاسخ به این سوال

تیم *PVM* کمبود قابلیت ها نسبت به اینگونه محصولات را قبول دارد و رسیدن به تمامی این ویژگیها را در زمانی کوتاه و سریع بر خود فرض میداند، به عبارتی ذکر نکات فوق تایید کننده کمبود ویژگیها نیست، بلکه فلسفه حرکت شرکت پویش را نشان میدهد.

هر چند که نبود این ویژگیها مؤید تولید محصولی ایرانی است، قابل ذکر است که در سالهای اخیر زمان زیادی از توسعه به ایجاد زیرساختهای لازم اختصاص یافته است که ثمره آن در *PVM* نسل دوم به خوبی نمایان است. این زیرساخت پیاده‌سازی ویژگیهای محصولات مشابه را امکان‌پذیر میسازد.

Sball تنها رقیب دنیای متن‌باز برای *Libvirt* است، به عبارتی دومین ساختار مدیریت *KVM* در دنیای متن‌باز میباشد که امید داریم با متن‌باز کردن آن، عرصه جدیدی را در فضای متن‌باز بگشاییم.

عرضه نرم‌افزار متن‌باز این امکان را به وجود می‌آورد تا به نام ایران مبحث قابل عرضه در انجمنها و کنفرانس های سالانه متن‌باز داشته باشیم.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۵ چرا KVM؟

همیشه از شرکت پویش داده نوین سؤال میشده است که چرا در توسعه *PVM* از تکنولوژی *KVM* استفاده شده است. دلایلی که در ادامه ذکر می‌شود عمده دلایل شرکت در ابتدای حرکت به سمت توسعه *PVM* میباشد:

- **kvm از متد توسعه متن باز بهره میبرد**

توسعه با متد متن باز به منزله همفکری تعداد زیادی از توسعه‌دهندگان و به عبارتی برخورداری از حوزه *MindShare* بزرگی است که این موضوع قابلیت اطمینان این بسته نرم‌افزاری را در حد زیادی افزایش داده است.

این موضوع در مقابل راه‌حلهای متن بسته و نیمه باز مزیت بسیار مهمی محسوب میشود. اهمیت این موضوع با نگاه به آینده و لزوم پشتیبانی طولانی مدت آن اهمیت بیشتری نیز پیدا میکند.

- **kvm از ابزار و استانداردهای متن‌باز بهره میبرد**

توسعه *kvm* به دلیل قرارگیری در حوزه متن‌باز وابسته به استانداردها و ابزار متن‌باز موجود میباشد. از جمله این موارد میتوان به فایل‌سیستم اشاره کرد که از فایل‌سیستمهای جاری در سیستم‌عامل لینوکس استفاده میشود که این موضوع برای تدوین راه‌حلهای پشتیبانگیری از اطلاعات بسیار حائز اهمیت است. این موضوع میتواند با فایل‌سیستم اختصاص بعضی از راه‌حلهای موجود مقایسه کرد که امکان دسترسی مستقیم به فایلها وجود ندارد.

همچنین تعامل بین ابزار متن باز، این امکان را فراهم می‌آورد که بتوان با اتخاذ راه‌حلهای مناسب بهره‌وری سیستم را تا حد زیادی افزایش داد. از این جمله میتوان به دیسکهای ماشینهای مجازی اشاره کرد که میتوانند هم به صورت فایل ذخیره شوند و یا قسمتی از یک دیسک مستقیما به ماشین اختصاص یابد. تلفیق این توانایی با ابزار *LVM* فرصتهای مناسبی را در مدیریت بهینه دیسک و حتی ساخت *Snapshot*ها فراهم‌می‌آورد.

همچنین این نکته نیز قابل توجه است که هم‌اکنون ابزارهای متن‌باز گوناگونی جهت توسعه کاربردها و

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

قابلیتهای ماشینهای مجازی در حال توسعه می‌باشد (مانند *openvswitch* - سوییچ مجازي).

وجود ابزار متن باز امکان پیاده‌سازی نیازهای کاربردی و غیرکاربردی(*Functional/NonFunctional*) را در یک محیط عملیاتی فراهم می‌آورد، و قدرت و حیطه عملیاتی مناسبی را در اختیار توسعه‌دهندگان قرار می‌دهد.

همچنین تعامل سازنده با ابزار متن باز امکان پشتیبانی طولانی مدت راه‌حلهای مبتنی بر *kvm* و همچنین انجام توسعه‌های افزایشی (*incremental*) را فراهم می‌آورد.

- **kvm عضوی از هسته سیستم‌عامل لینوکس است**

بنابراین هر جا که لینوکس است، *kvm* هم خواهد بود. رمز موفقیت لینوکس پشتیبانی از سخت‌افزارهای گوناگون و همچنین توانایی عملیاتی در حوزه‌های *Desktop* و *server*میباشد که این قابلیت به دلیل انعطاف هسته لینوکس جهت انجام تنظیمات بهینه برای هر حوزه میباشد.

در نتیجه *kvm* حوزه سخت‌افزاری وسیعتری را نسبت به راه‌حلهای موجود پشتیبانی میکند و این به دلیل تفکیک روشن وظایف بین هسته لینوکس(مدیریت فعالیتهای سیستم‌عامل مانند سخت‌افزار) و *kvm*(مدیریت ماشینهای مجازی) میباشد.

همچنین به دلیل انعطاف هسته لینوکس،طیف کارآیی *kvm* از کاربری خانگی تا کاربری حرفه ای گسترده است که موجب افزایش کاربران آن و به تبع گسترش سریعتر خصوصیات کاربردی و غیرکاربردی (*Functional/NonFunctional*) آن میشود.

- **kvm ساده‌تر و سبکتر از راه‌حلهای موجود است**

چنانچه ذکر شد راه‌حلهای مطرح شده یک سیستم‌عامل کوچک هستند که علاوه بر انجام اعمال مربوط به ماشینهای مجازی، میبایست وظایف جاری یک سیستم‌عامل همانند زمانبندی (*Scheduling*)، مدیریت منابع، توسعه درایورهای سخت‌افزاری و ... را انجام دهند.

kvm عضوی از هسته لینوکس است که تنها وظیفه واگذار شده به آن مدیریت ماشینهای مجازی است بنابراین از تواناییهای خارق‌العاده لینوکس به عنوان یک سیستم‌عامل تمام عیار به خوبی استفاده میکند. از

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

این منظر *kvm* یک راه‌حل سبک و ساده بوده که میتواند نرخ توسعه سریعتری نیز داشته باشد(به دلیل حجم اندك وظایف واگذار شده به آن).

- ***kvm* ماشینهای مجازی را در قالب *process* های لینوکس مدیریت میکند**

به عبارتی در فضای *kvm* هر ماشین‌مجازی یک *process* در سیستم‌عامل لینوکس میباشد. بنابراین هرگونه عملیاتی که برای یک پردازش در لینوکس مطرح است برای یک ماشین‌مجازی نیز قابل طرح است که از آن جمله است:

۱- اعمال اولویت به ماشین‌مجازی،

۲- اعمال محدودیت در استفاده از منابع سیستم،

۳- مانیتورینگ بسیار راحت ماشین‌مجازی،

۴- استفاده از ابزارهای مدیریت پردازشها برای راهبری ماشین‌مجازی

- ***kvm* یک *bare metal hypervisor* است**

kvm به صورت مستقیم بر روی سخت‌افزار اجرا شده و تعامل مستقیم با آن دارد. ارتباط مستقیم با سخت‌افزار، بر روی بهره‌وری ماشینهای مجازی و سرعت انجام عملیات *io* آنها تاثیر بسزایی دارد.

- ***kvm* توانایی شبیه سازی سیستمهای گوناگون را دارد**

به عبارتی *kvm* این امکان را دارد که انواع سیستمهای کامپیوتری را بر بستر سیستم فعلی شبیه سازی کند. البته این موضوع در توسعه نرم‌افزارهای *Multi-Platform* کاربرد دارد اما وجود آن دلیلی بر وجود تواناییهای قابل توجه و طراحی مناسب این نرم‌افزار در پیوند با نرم‌افزارهای متن باز موجود میباشد.

این توانایی حاصل پیوند *kvm* با نرم‌افزار متن باز *Qemu* به عنوان یك شبیه ساز سیستم است که یك سیستم کامپیوتری را به صورت کامل شبیه سازی میکند. این سیستم کامپیوتری میتواند یك *PC* و یا سیستمهای غیر *PC* مانند *PowerPc, Sparc32, Sparc64, MIPS, ARM* باشد.

- ***kvm* درایورهای سخت‌افزاری گوناگونی را شبیه سازی میکند**

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

kvm برای هر سیستم کامپیوتری، درایورهای گوناگون کارت شبکه، کارت صدا و نوع دیسك را شبیه سازی میکند. وجه مطلوب این موضوع برای سیستم عاملهای قدیمی قابل توجه است که میتوان درایور مناسبی را برای سیستم انتخاب کرد.

همچنین در پیوند انواع ماشینهای مجازی با سیستم عاملهای گوناگون که به عنوان بستر (*host*) استفاده میشوند (بسته به نوع تنظیمات هسته) میتوان درایوری که بهترین خروجی را دارد انتخاب کرد.

- ***kvm* یعنی لینوکس**

اطمینان از وجود راه حلهای فنی برای مشکلات پیش رو در انجام یك پروژه حساس و بزرگ بسیار مهم است.

لینوکس یعنی انعطاف و اطمینان. به عبارت بهتر کار در فضای لینوکس این اطمینان را برای توسعه‌دهنده ایجاد میکند که به هر صورت برای هر نیاز راه حل فنی وجود دارد و یا حداقل امکان توسعه یك راهکار مناسب متصور است. این موضوع به دلیل حوزه کاربری وسیع لینوکس میباشد.

دفتر مرکزی: یزد - خیابان مطهری - مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com

۶ سخن پایانی

تجارب چندین ساله شرکت پویش داده نوین در زمینه توسعه و به کارگیری نرم افزارهای سیستمی و امنیتی و همچنین تجربه تحویل کامل یک پروژه سنگین (*UTM*) به کارفرمای مربوطه، همه و همه پشتوانه فنی شرکت به منظور توسعه نرم افزاری زیرساختی، حساس و حجیم به نام *PVM* بوده است.

تجارب ما در پشتیبانی *PVM* در یک دهه گذشته مؤید پیش فرضهای ما در توسعه *PVM* میباشد که امیدواریم این مسیر با همبستگی بیشتر منجر به افتخاری گران برای کشور عزیزمان گردد.

دفتر مرکزی: یزد-خیابان مطهری-مرکز فناوری اقبال - واحد C۲۵ کدپستی: ۸۹۱۷۶۹۷۹۹۸ تلفن (۳ خط): ۷۲۶۵۳۴۳ - ۰۳۵۱ فکس: ۷۲۴۵۰۸۷ - ۰۳۵۱
www.pdnsoft.com Email: info@pdnsoft.com