MapReduce框架

杨志丰
yzf@net.pku.edu.cn

TFS小组

March 19, 2008

Outlines

微软首席预言家：并行计算将成未来重点

sina 科技时代 | 科技时代 > 业界 > 正文

微软首席预言家：并行计算将成未来重点

http://www.sina.com.cn 2008年03月13日 21:49 新浪科技

微软首席研究和战略官克莱格·穆迪

预测未来并不难

穆迪2006年接替微软联合创始人比尔·盖茨，成为了微软首席预言家。他目前正在准备帮助微软完成一次重要技术转型，即向并行计算转型。他认为，此次转型将同当初PC和互联网的兴起同样重要。他说：“要对技术的发展方向有相当准确的感知并不困难，我们甚至可以朝着这一方向前进，取得不错的技术进展。但是，几乎我们做的每一件事所花费的时间都超过了预期。”

穆迪控制着微软70亿美元的研发预算，他十分清楚有前途的技术需要很长时间来开发。毕竟，他一直在负责微软的网络电视和非传统计算项目。最近几年来，并行计算吸引了业界的高度关注，被认为是下一个重大技术进展。利用并行计算技术，计算机可以将不同的任务分配给多个处理器同时处理，而不是由一个处理器每次处理一项任务，从而提升了计算机的运算速度。

微软首席预言家：并行计算将成未来重点

并行计算第一步

并行计算的全部潜力几乎深不可测，它可以推动机器人领域的重大技术进展，还可以加速多语种实时翻译软件的出现。计算机行业已经在走向并行计算的道路上迈出第一步，这就是多核处理器的普及。但穆迪认为，这只是冰山的一角。

为了将计算能力最大化，软件厂商需要改变软件程序员的工作方式。目前，全世界只有少数程序员知道如何在代码中将计算任务划分为多个部分，并交由多个处理器同时处理，而不是使用传统的线性方式，即每次处理一个任务。要实现这一目标，需要一种全新的编程语言，可能会影响软件每一部分的开发方式。

穆迪承认，这一问题难以解决，它所带来的挑战将于未来五到十年变得更加明显。加盟微软之前，他曾经在超级计算机公司Alliant Computer Systems从事并行计算开发。由于处理器主频的提升遭遇了散热和功耗方面的阻碍，芯片行业开始开发多核处理器，希望通过这一方式进一步提升处理器性能。在这种情况下，向并行计算转型变得更加必要。英特尔和AMD已经推出了四核处理器。硅谷创业公司Tilera预计，1000核处理器将于2014年出现。

等待杀手级应用

过去两年里，穆迪和微软首席软件设计师雷·奥谢(Ray Ozzie)分担了盖茨的职责。其中，前者主要负责把握长期技术方向，而后者则负责制定短期规划。在穆迪的建议和领导下，微软研发部门有超过800名博士正在研发新技术，包括网络搜索、实时翻译和触摸屏技术等等。在此之中，并行计算当然是一个高优先级项目，因为它可能会影响微软的方方面面。

穆迪预计，未来20年里，将会出现比现在强大100倍的计算机，企业数据中心甚至可以封装到笔记本或手机的芯片之中。他说，杀手级应用将使这种计算能力进入最前沿，就像字处理和电子表格软件推动PC普及，以及电子邮件和浏览器促进互联网流行一样。

对于微软这样拥有约8万名员工的大企业来说，放弃自己过去的优势和传统运作模式去关注新技术，并不是一件容易的事情。穆迪就此表示：“盖茨和我都曾经说过，除非是一名极度乐观主义者，否则很难完成这样的工作，因为要对抗所有反对变革的人。”(孟帆)

“云计算”

Google在中国宣布"云计算"计划-产业-《财经网》

Google在中国宣布“云计算”计划

《财经》记者 何华峰 《财经网》 [03-17 10:49]

该计划是未来国际IT巨头争夺的制高点，也是Google下一步重要战略

【《财经网》专稿/记者 何华峰】3月17日，Google全球CEO埃里克·斯密特（Eric Schmidt）在北京访问期间，宣布在中国大陆推出“云计算（Cloud Computing）”计划。

“云计算”是一个新兴概念，被认为是Google下一阶段的重要战略。目前的计算模式以传统的个人电脑为中心，而“云计算”则是将计算和数据分布在大量的分布式计算机上，用户可以通过多种接入方式，比如电脑和手机方便地接入网络获得应用和服务。

“云计算”也是目前国际最主要的巨头争夺的制高点。

Google于去年10月在全球宣布了此项计划，Google与IBM开展雄心勃勃的合作，要把全球多所大学纳入“云计算”中。

亚马逊（Amazon.com）已于2007年向开发者开放了名为“弹性计算机云”的服务，让小软件公司可以按需购买亚马逊数据中心的处理能力。

2007年8月，IBM高调推出“蓝云（Blue Cloud）”计划。这一计划已经在上海推出。

2007年11月，雅虎也将一个小规模的服务器群，即“云”，开放给卡内基-梅隆大学的研究人员。

在中国的“云计算”计划中，清华大学将是第一家参与合作的高校。它将与Google合作开设“大规模数据处理”

TFS Group 天网搜索

例子：排序问题

- 分治
- 分布式如何做?

传统高性能并行计算

目标:求解大规模问题和复杂系统

- 新药研制
- 催化剂设计
- 燃料燃烧原理
- 海洋模型模拟
- 数字解剖
- 空气污染
- 蛋白质结构设计
- 图像理解
- 密码破译

并行计算机体系结构

- SISD
- SIMD (PVP)
- MISD
- MIMD (SMP, DSM, Cluster)

DISC: *Data Intensive* Super Computing[1]

they acquire and maintain continually changing data sets, in addition to performing large-scale computations over the data

- Web search without language barriers
- Inferring biological function from genomic sequences
- Predicting and modeling the effects of earthquakes
- Discovering new astronomical phenomena from telescope imagery data
- Understanding the spatial and temporal patterns of brain behavior based on MRI data

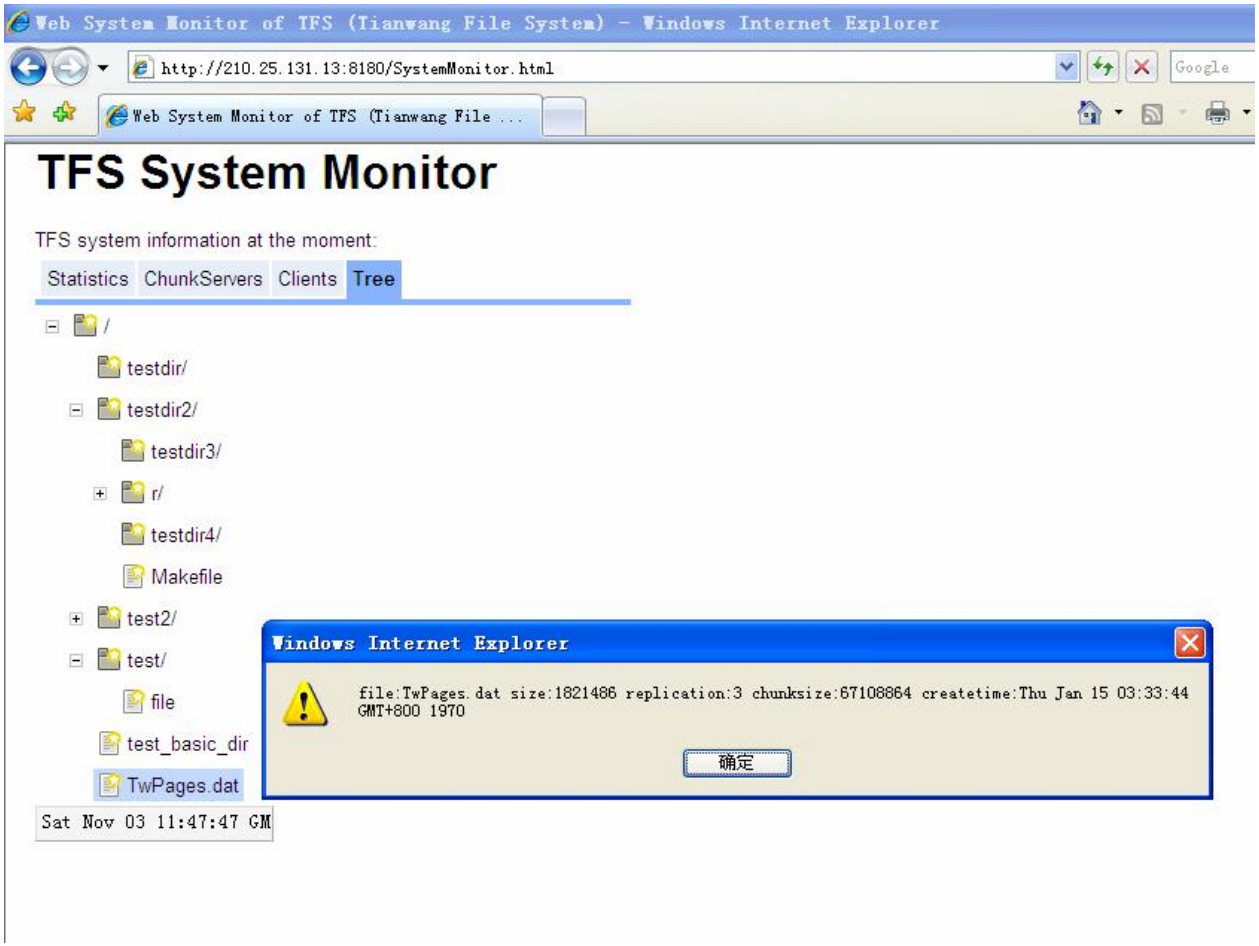
DISC Key Principles

- Intrinsic, rather than extrinsic data
- High-level programming models for expressing computations over the data
- Interactive access
- Scalable mechanisms to ensure high reliability and availability

Google: A DISC Case Study

- The Google File System [15]
- MapReduce [4]
- Bigtable [5]

System Monitor



TFS Shell

```
[tfs@build1 tfs]$ ./tfs_shell
tfs_shell:/> ls
drw-rw-rw-  1              0        0 2007-11-11 11:14 test
drw-rw-rw-  1              0        0 2007-11-11 19:59 data
-rw-rw-rw-  3        2839798 67108864 2007-11-11 19:33 testpage.dat
-rw-rw-rw-  3      629146974 67108864 2008-00-03 17:32 TwPages.dat
-rw-rw-rw-  2        2839798  8388608 2008-00-03 17:36 page.dat
-rw-rw-rw-  3          19383 67108864 2008-00-03 18:54 cc_page_out.dat
-rw-rw-rw-  2          45101  8388608 2008-00-05 14:01 tfs_master.log
-rw-rw-rw-  2         480242  8388608 2008-00-05 14:01 mapred_master.log
-rw-rw-rw-  2         544021  8388608 2008-00-05 14:13 doc.txt
-rw-rw-rw-  3           1344 67108864 2008-00-11 21:42 cc_twpages_out.dat
-rw-rw-rw-  3        7800000 67108864 2008-00-11 22:23 charset.txt
-rw-rw-rw-  3           3339 67108864 2008-00-11 22:49 doc_out.dat
-rw-rw-rw-  2          73236  8388608 2008-02-11 21:41 Makefile
tfs_shell:/> cd data
tfs_shell:/data/> ls
-rw-rw-rw-  3   212720721991 67108864 2007-11-11 19:59 Cwt200G.dat
tfs_shell:/data/> help
Usage:
ls
cd <tfs_dir>
mkdir <tfs_dir>
rm <tfs_file>
rm -rf <tfs_dir>
put <local_file> [<tfs_file>]
get <tfs_file> [<local_file>]
pwd
quit
!<system_command>
tfs_shell:/data/> bye
[tfs@build1 tfs]$
```

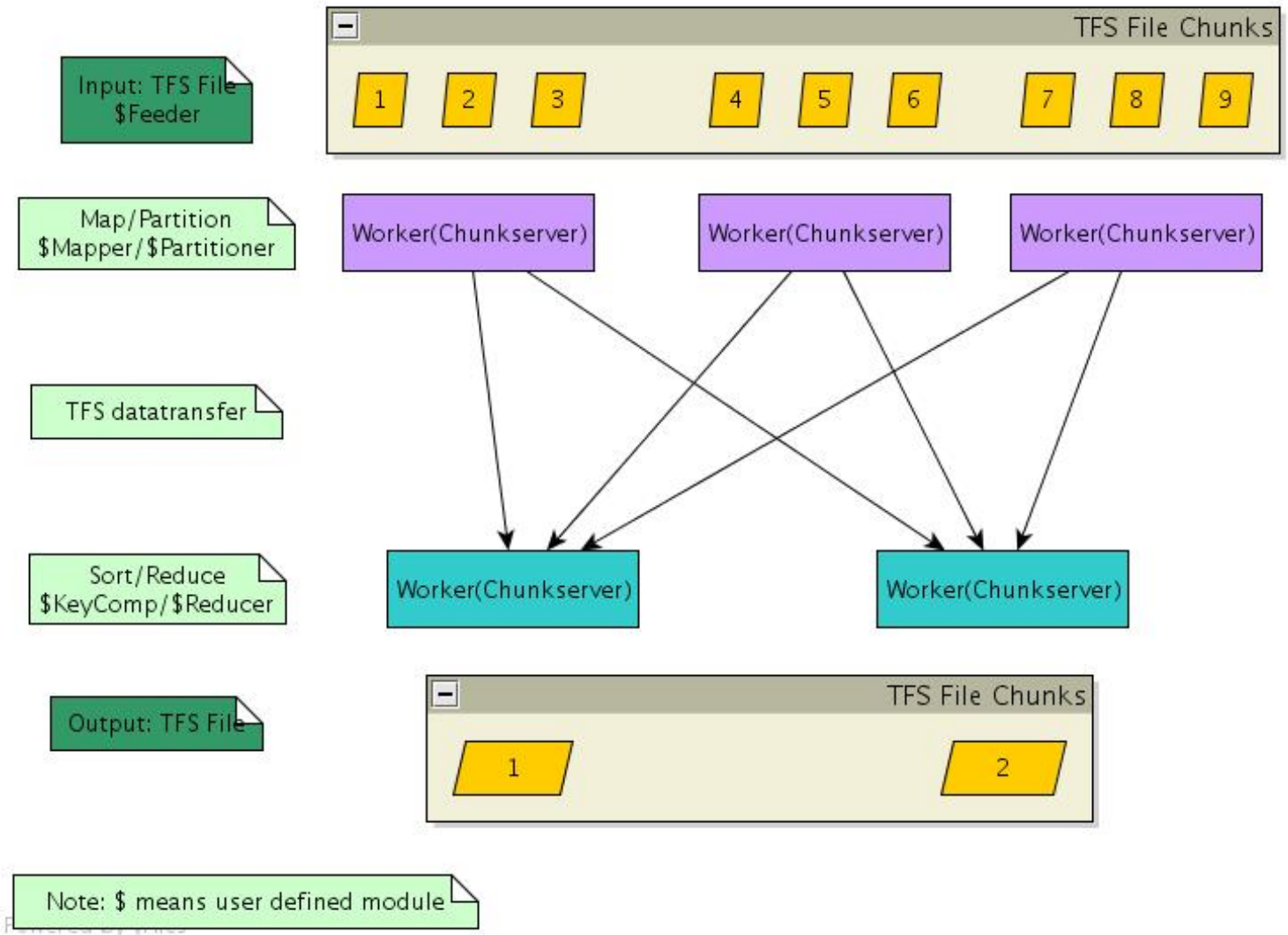
TFS Sample Application

```
#include <tfs.h>
using namespace tfs::api;

int main() {
    char *s = new char[100000];
    char *ss = new char[100000];

    AppendStream as("/dir/filename", 0);
    as.append (s, sizeof(s));
    as.close();

    InputStream is("/dir/filename", 0);
    is.read(ss, sizeof(ss));
    is.close();
}
```

MapReduce特性

- 基于分布式文件系统TFS
- 对一类并行程序的抽象
- 海量数据处理基础设施
  - 自动并行化
  - 高度可扩展
  - 高容错

交互示意图


Input: TFS File
$Feeder
TFS File Chunks
1
2
3
4
5
6
7
8
9
Map/Partition
$Mapper/$Partitioner
Worker(Chunkserver)
Worker(Chunkserver)
Worker(Chunkserver)
TFS datatransfer
Sort/Reduce
$KeyComp/$Reducer
Worker(Chunkserver)
Worker(Chunkserver)
Output: TFS File
TFS File Chunks
1
2
Note: $ means user defined module

MapReduce运行时支持

- 任务动态调度，负载均衡
- 错误恢复（任务失败，Worker失效）
- 任务重做
- TFS支持，map操作更高效

MapReduce编程模型

$$map \quad (k1, v1) \quad \rightarrow list(k2, v2) \tag{1}$$

$$reduce \quad (k2, list(v2)) \quad \rightarrow list(v2) \tag{2}$$

MapReduce API: Feeder

```
namespace mapreduce{
    /**
     * Feeder supplies a interface to extract record c
     * */
    class KeyValue;

    class IFeeder : public classloader::IUnknown {
        public:
            /**
             * Access the next record.
             * @return : KeyValue ojbect;
             *           NULL if reach the end.
             */
            virtual boost::shared_ptr<KeyValue> next()
    };
}
```

MapReduce API: Mapper

```
namespace mapreduce{
    class IMapper : public classloader::IUnknown {
        public:
            /**
             *map the data
             *
             */
            virtual int map(const KeyValue& inKeyValue
                            std::vector<boost::shared_ptr<KeyV
    };
}
```

MapReduce API: KeyHasher

```
namespace mapreduce{
    class IKeyHasher : public classloader::IUnknown{
        public:
            /**
             * hash function on the key
             *
             */
            virtual unsigned int hash(const KeyValueF
    };
}
```

MapReduce API: KeyComparison

```
namespace mapreduce {
    class IKeyComparison : public classloader::IUnknow
        public:
            /**
             * compare two keys
             *
             */
            virtual int compare(const KeyValueField& f
    };
}
```

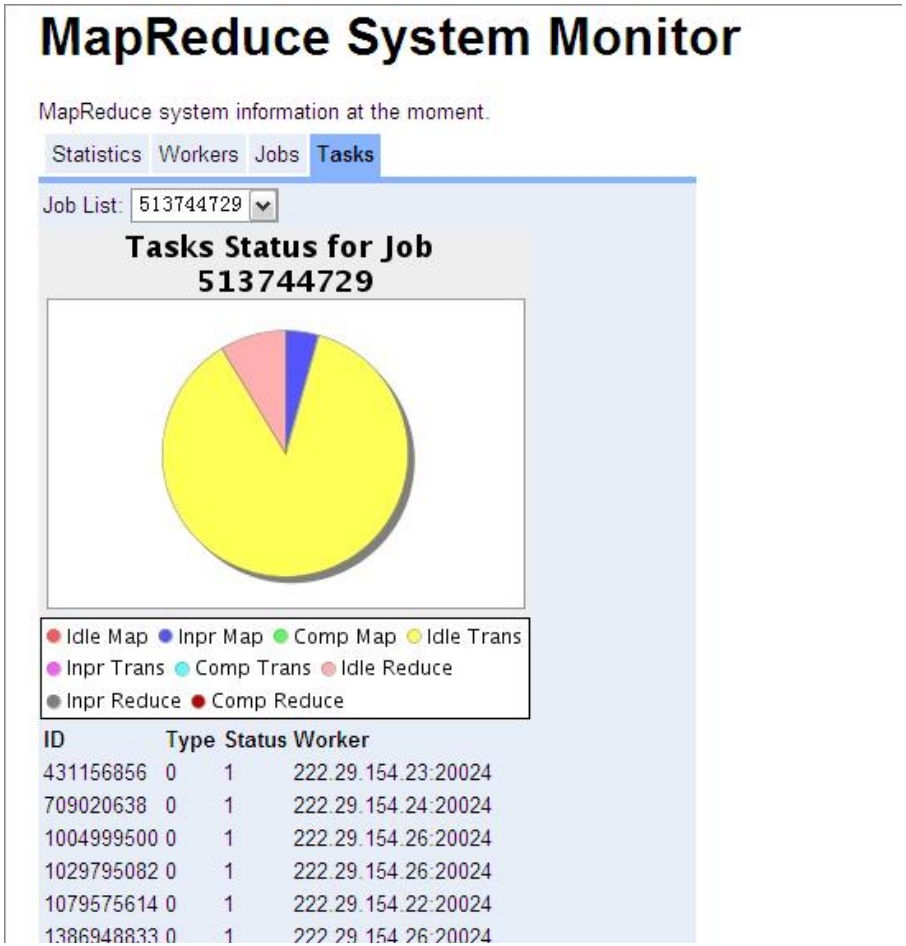
MapReduce API: Reducer

```
namespace mapreduce{
    class KeyValue;
    class IReducer : public classloader::IUnknown{
        public:
            virtual int reduce(const std::vector<boost
                                    std::vector<boost::shar
    };
}
```

MapReduce API: Writer

```
namespace mapreduce{
    class IWriter : public classloader::IUnknown {
    public:
        /**
         * serialize the keyvalue record
         *
         * @param keyValue
         *
         * @return
         */
        virtual Buffer convert(KeyValue &keyValue) cor
    };
}
```

MapReduce Monitor



相关工作

- Interpreting the Data: Parallel Analysis with Sawzall[14]
- Dryad[10]
- Map-reduce-merge [2]

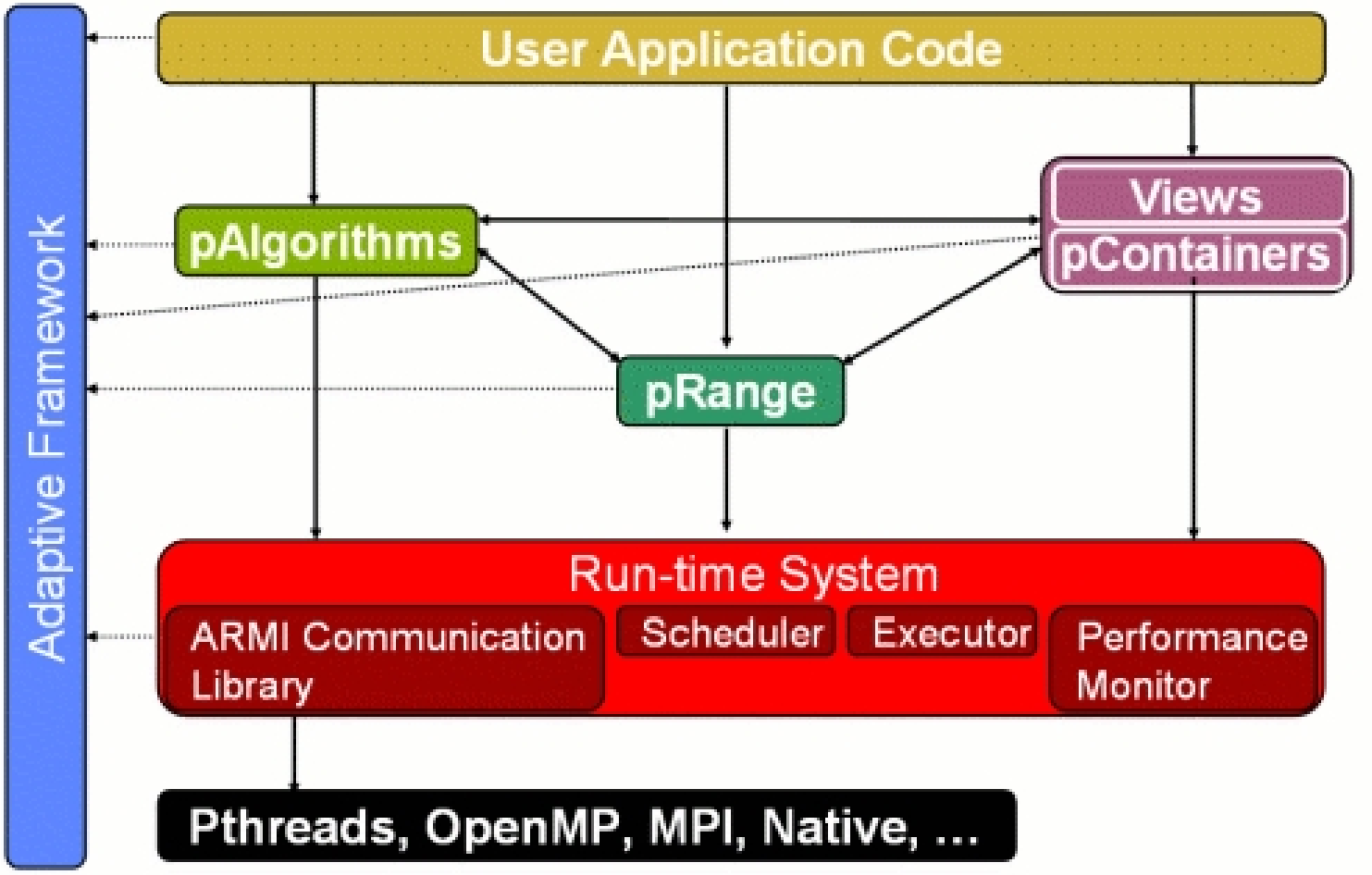
- Map-Reduce for Machine Learning on Multicore[3]
- Fully Distributed EM for Very Large Datasets[18]
- Design and Implementation of Distributed Ray Tracing Using MapReduce [19]

相关系统

- Hadoop [7] [8]
- KFS [11]
- Bigtable [5]
- Hypertable [9]
- MCSTL [12][16], STXXL [17], GCC parallel mode [6]

相关系统

STAPL: Standard Template Adaptive Parallel Library



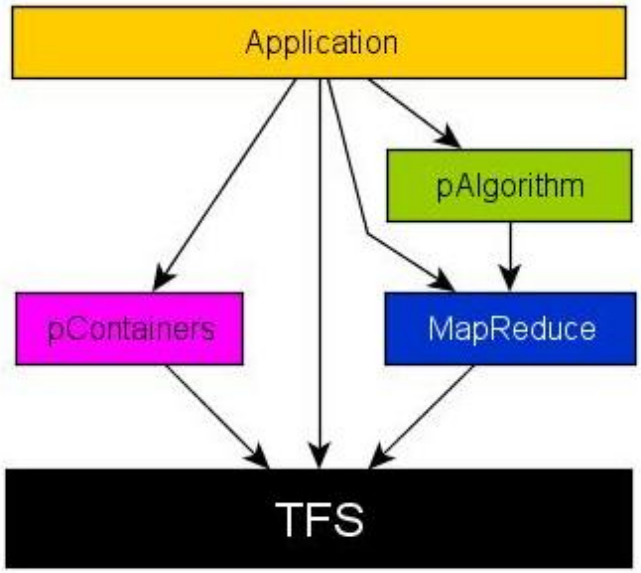
Patterson: The data center is the computer[13]

- The data center is now the computer
- What is the equivalent of the ADD instruction for a data center?
- If MapReduce is the first instruction of the pdata center computer, I canot wait to see the rest of the instruction set, as well as the data center programming language, the data center operating system, the data center storage systems, and more.

我们想做什么?

TPlatform：基于MapReduce的通用并行计算平台

- 给用户提供更高的抽象，更简单的编程
- 基于MapReduce提供STL algorithm中大部分函数的MapReduce实现
- 基于TFS提供STL containers中的大部分容器的分布式实现



Bibliography I

Bryant, R. E.
Data-intensive supercomputing:the case for disc.
Tech. rep., School of Computer Science, CMU, 2007.

chih Yang, H., Dasdan, A., Hsiao, R.-L., and Parker, D. S.

Map-reduce-merge: simplified relational data processing on large clusters.
In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 2007), ACM, pp. 1029–1040.

Bibliography II

Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G. R., Ng, A. Y., and Olukotun, K.

Map-reduce for machine learning on multicore.

In *NIPS* (2006), B. Schölkopf, J. C. Platt, and T. Hoffman, Eds., MIT Press, pp. 281–288.

Dean, J., and Ghemawat, S.

Mapreduce: Simplified data processing on large clusters.

In *OSDI* (2004).

Fay Chang, J. D., and Ghemawat, S.

Bigtable: A distributed storage system for structured data.

In *OSDI* (2006), pp. 205–218.

GCC.

The libstdc++ parallel mode.

Bibliography III

Hadoop.
The hadoop project.
http://lucene.apache.org/hadoop/, 2006.

Hadoop.
The Hadoop Distributed File System: Architurecture and Design, 2007.
Available at
http://lucene.apache.org/hadoop/hdfs_design.html.

Hypertable.
Hypertable: An open source, high performance, scalable database.

Bibliography IV

Isard, M., Budiu, M., Yu, Y., Birrell, A., and Fetterly, D.

Dryad: distributed data-parallel programs from sequential building blocks.

In *EuroSys '07: Proceedings of the ACM SIGOPS/EuroSys European Conference on Computer Systems 2007* (New York, NY, USA, 2007), ACM, pp. 59–72.

KFS.

Kosmos distributed file system.

`http://kosmosfs.sourceforge.net/`, 2007.

MCSTL.

Mcstl: The multi-core standard template library.

Bibliography V

Patterson, D. A.
Technical perspective: the data center is the computer.
Commun. ACM 51, 1 (2008), 105–105.

Pike, R., Dorward, S., Griesemer, R., and Quinlan, S.
Interpreting the data: Parallel analysis with sawzall.
Scientific Programming 13, 4 (2005), 277–298.

Sanjay Ghemawat, H. G., and Leung, S.-T.
The google file system.
In *SOSP' 03* (2003).
SOSP' 03, October 1922, 2003, Bolton Landing, New York, USA.h.

Bibliography VI

Singler, J., Sanders, P., and Putze, F.
Mcstl: The multi-core standard template library.
In *Euro-Par* (2007), A.-M. Kermarrec, L. Bougé, and T. Priol, Eds., vol. 4641 of *Lecture Notes in Computer Science*, Springer, pp. 682–694.

STXXL.
Stxxl : Standard template library for extra large data sets.

Wolfe, J., Haghighi, A., and Klein, D.
Fully distributed em for very large datasets.
Tech. rep., Computer Science Division, UC Berkeley, 2007.

Bibliography VII

Xin-jie, Z., Cheng-rong, Z., and Qi-bang, X.
Design and implementation of distributed ray tracing using mapreduce.
Computer Engineering 33 (22) (2007), 83–85.
Dept. of Computer Science and Technology, Tongji University, Shanghai 200331.