



国家知识产权局

250014

山东省济南市历下区经十路 17703 号华特广场 B510 室 济南圣达知
识产权代理有限公司
周凌云(0531-68605722)

发文日:

2024年04月11日



申请号: 202410435319.0

发文序号: 2024041101247350

专利申请受理通知书

根据专利法第 28 条及其实施细则第 43 条、第 44 条的规定,申请人提出的专利申请已由国家知识产权局受理。现将确定的申请号、申请日等信息通知如下:

申请号: 2024104353190

申请日: 2024 年 04 月 11 日

申请人: 山东省计算中心(国家超级计算济南中心),齐鲁工业大学(山东省科学院)

发明人: 谭立状,隋宗瑞,王新航,董鑫,史慧玲,张玮

发明创造名称: CPU/FPGA 协同的网络遥测乱序重排方法及系统

经核实,国家知识产权局确认收到文件如下:

权利要求书 1 份 3 页,权利要求项数: 10 项

说明书 1 份 14 页

说明书附图 1 份 3 页

说明书摘要 1 份 1 页

发明专利请求书 1 份 5 页

实质审查请求书 文件份数: 1 份

申请方案卷号: 2024702048

提示:

1. 申请人收到专利申请受理通知书之后,认为其记载的内容与申请人所提交的相应内容不一致时,可以向国家知识产权局请求更正。

2. 申请人收到专利申请受理通知书之后,再向国家知识产权局办理各种手续时,均应当准确、清晰地写明申请号。

审查员: 自动受理

联系电话: 010-62356655

审查部门: 初审及流程管理部



200101
2023.03

纸件申请,回函请寄: 100088 北京市海淀区蓟门桥西土城路 6 号 国家知识产权局专利局受理处收
电子申请,应当通过专利业务办理系统以电子文件形式提交相关文件。除另有规定外,以纸件等其他形式提交的文件视为未提交。

权利要求书

1. CPU/FPGA 协同的网络遥测乱序重排方法，其特征在于，包括以下步骤：

CPU 将带内网络遥测报告拆分为多个待排序包和对应的信息包，构建内存缓冲区和动态指针管理区，将待排序包分别存储至对应的内存缓冲区中，将信息包存储至动态指针管理区中，所述待排序包中包括时间戳和指向其对应信息包的指针；

CPU 将多个待排序包传输至 FPGA 的片外存储资源 DDR 中；

FPGA 将 DDR 中的多个待排序包调入，并调用对应的多个加速排序内核并行进行待排序包的排序；

FPGA 排序完成后，将排序后的有序包传入 DDR，并将有序包从 DDR 传输至主机端的内存缓冲区；

CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复，得到一条排序完成的时序数据库记录。

2. 如权利要求 1 所述的 CPU/FPGA 协同的网络遥测乱序重排方法，其特征在于：

CPU 将带内网络遥测报告拆分为时间戳和时间戳对应的信息，将时间戳对应的信息作为信息包存储至动态指针管理区，所述时间戳对应的信息包括出口利用率和丢包统计信息；

构建数据指针，指向信息包在动态指针管理区中的存储地址，将时间戳和数据指针组合为待排序包，存储至对应的内存缓冲区中。

3. 如权利要求 1 所述的 CPU/FPGA 协同的网络遥测乱序重排方法，其特征在于：

权利要求书

加速排序内核读取 DDR 中对应的待排序包至 FPGA 片内缓存 BRAM 中；

加速排序内核读取参数，包括待排序包数组的首地址和待排序包数组的大小；

FPGA 启动多个加速排序内核，多个加速排序内核并行执行加速排序过程。

4. 如权利要求 3 所述的 CPU/FPGA 协同的网络遥测乱序重排方法，其特征在于，加速排序内核执行 8 路归并排序算法，对待排序包进行排序。

5. 如权利要求 3 所述的 CPU/FPGA 协同的网络遥测乱序重排方法，其特征在于，每个加速排序内核排序对应遥测节点的所有待排序包，每个内存缓冲区对应遥测路径中的一个节点。

6. 如权利要求 1 所述的 CPU/FPGA 协同的网络遥测乱序重排方法，其特征在于：

CPU 读取有序包至对应的内存缓冲区中；

CPU 根据有序包中的信息指针，读取指向动态指针管理区中的地址，找寻动态指针管理区中对应的信息包；

将有序包中的时间戳与对应的信息包重组组合，完成重组恢复。

7. 如权利要求 1 所述的 CPU/FPGA 协同的网络遥测乱序重排方法，其特征在于，内存缓冲区与 DDR 之间通过 PCIe 进行数据交互。

8. CPU/FPGA 协同的网络遥测乱序重排系统，其特征在于，包括：

预处理模块，被配置为：CPU 将带内网络遥测报告拆分为多个待排序包和对应的信息包，构建内存缓冲区和动态指针管理区，将待排序包分别

权利要求书

存储至对应的内存缓冲区中，将信息包存储至动态指针管理区中，所述待排序包中包括时间戳和指向其对应信息包的指针；

传输模块，被配置为：CPU 将多个待排序包传输至 FPGA 的片外存储资源 DDR 中；

并行排序模块，被配置为：FPGA 将 DDR 中的多个待排序包调入，并调用对应的多个加速排序内核并行进行待排序包的排序；

回传模块，被配置为：FPGA 排序完成后，将排序后的有序包传入 DDR，并将有序包从 DDR 传输至主机端的内存缓冲区；

重组模块，被配置为：CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复，得到一条排序完成的时序数据库记录。

9. 计算机可读存储介质，其上存储有程序，其特征在于，该程序被处理器执行时实现如权利要求 1-7 任一项所述的 CPU/FPGA 协同的网络遥测乱序重排方法中的步骤。

10. 电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的程序，其特征在于，所述处理器执行所述程序时实现如权利要求 1-7 任一项所述的 CPU/FPGA 协同的网络遥测乱序重排方法中的步骤。

说明书

CPU/FPGA 协同的网络遥测乱序重排方法及系统

技术领域

本发明属于网络遥测技术领域，尤其涉及 CPU/FPGA 协同的网络遥测乱序重排方法及系统。

背景技术

网络测量（遥测）是网络操作维护管理的重要手段，现有传统 SNMP 方案普遍面临性能低（snmp_interval 默认 30s）、开销大（头部封装占比大）等突出问题。随着软件定义网络和可编程数据平面技术的出现，带内网络遥测通过转发路径交换节点对数据包依次插入网络测量元数据完成网络状态采集，已经成为一种新型网络遥测方案。相比较 SNMP 等带外网络遥测方案，带内网络遥测将网络管理通道与网络数据通道绑定，显著降低网络遥测开销，支持逐包、逐路径的高精度细粒度网络遥测。

发明人发现，在带内网络遥测中，由于不同遥测路径可能存在节点重叠，且逐跳网络延迟不可控，遥测服务器处理遥测报告时面临严重的乱序问题。遥测服务器在处理乱序遥测报告，提取元数据放到对应的时序数据库表时，缺少较好的处理方法。如何兼顾带内网络遥测数据写入性能与查询性能，缓解数据乱序对遥测系统效率影响，是制约遥测系统的关键性能瓶颈问题。

发明内容

为克服上述现有技术的不足，本发明提供了 CPU/FPGA 协同的网络遥测乱序重排方法及系统，使得 CPU 和 FPGA 协同合作，实现了编程模式中的数据并行，显著提升了遥测服务器处理遥测数据的速度，降低了 CPU 的占用率和处理延迟。

为实现上述目的，本发明的一个或多个实施例提供了如下技术方案：

本发明第一方面提供了 CPU/FPGA 协同的网络遥测乱序重排方法。

说明书

CPU/FPGA 协同的网络遥测乱序重排方法，包括以下步骤：

CPU 将带内网络遥测报告拆分为多个待排序包和对应的信息包，构建内存缓冲区和动态指针管理区，将待排序包分别存储至对应的内存缓冲区中，将信息包存储至动态指针管理区中，所述待排序包中包括时间戳和指向其对应信息包的指针；

CPU 将多个待排序包传输至 FPGA 的片外存储资源 DDR 中；

FPGA 将 DDR 中的多个待排序包调入，并调用对应的多个加速排序内核并行进行待排序包的排序；

FPGA 排序完成后，将排序后的有序包传入 DDR，并将有序包从 DDR 传输至主机端的内存缓冲区；

CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复，得到一条排序完成的时序数据库记录。

本发明第二方面提供了 CPU/FPGA 协同的网络遥测乱序重排系统。

CPU/FPGA 协同的网络遥测乱序重排系统，包括：

预处理模块，被配置为：CPU 将带内网络遥测报告拆分为多个待排序包和对应的信息包，构建内存缓冲区和动态指针管理区，将待排序包分别存储至对应的内存缓冲区中，将信息包存储至动态指针管理区中，所述待排序包中包括时间戳和指向其对应信息包的指针；

传输模块，被配置为：CPU 将多个待排序包传输至 FPGA 的片外存储资源 DDR 中；

并行排序模块，被配置为：FPGA 将 DDR 中的多个待排序包调入，并调用对应的多个加速排序内核并行进行待排序包的排序；

回传模块，被配置为：FPGA 排序完成后，将排序后的有序包传入 DDR，并将有序包从 DDR 传输至主机端的内存缓冲区；

重组模块，被配置为：CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复，得到一条排序完成的时序数据库

记录。

本发明第三方面提供了计算机可读存储介质，其上存储有程序，该程序被处理器执行时实现如本发明第一方面所述的 CPU/FPGA 协同的网络遥测乱序重排方法中的步骤。

本发明第四方面提供了电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的程序，所述处理器执行所述程序时实现如本发明第一方面所述的 CPU/FPGA 协同的网络遥测乱序重排方法中的步骤。

以上一个或多个技术方案存在以下有益效果：

1、本发明提供了一种 CPU/FPGA 协同的网络遥测乱序重排方法及系统，在遥测服务器中，通过引入 FPGA 来加速硬件卸载带内网络遥测乱序重排任务，CPU 和 FPGA 协同合作，实现了编程模式中的数据并行，显著提升了遥测服务器处理遥测数据的速度，降低了 CPU 的占用率和处理延迟。

2、CPU 构建了多个内存缓冲区用来分别存储对应的待排序包，并把待排序包放入其遥测节点对应的数据块中，为这些数据块分配多个加速排序内核进行并行排序，进一步提高了大数据量的排序速度。

3、CPU 构建了一块动态指针管理区来存放信息包，包括出口利用率和丢包统计等信息，最后排序完成后，将有序包的时间戳与其对应的出口利用率和丢包统计重组组合，构成一条时序数据库的记录，这种构造动态指针管理区的方式，抽离了与重排序过程无关的信息，缓解了 CPU 与 FPGA 之间数据传输的压力，并且提高了 FPGA 重排序的效率。

4、CPU 为多个加速排序内核分配不同的 DDR 偏移地址，多个加速排序内核可以并行读取数据、排序数据、返回数据，从而实现了 FPGA 的并行化，加快了整体排序速率，提高了 FPGA 的排序效率。

本发明附加方面的优点将在下面的描述中部分给出，部分将从下面的描述中变得明显，或通过本发明的实践了解到。

附图说明

构成本发明的一部分的说明书附图用来提供对本发明的进一步理解，本发明的示意性实施例及其说明用于解释本发明，并不构成对本发明的不当限定。

图 1 为本发明实施例一整体方法流程图。

图 2 为本发明遥测报告的格式结构及乱序解释图。

图 3 为本发明 CPU 预处理带内网络遥测报告的示意图。

图 4 为本发明 CPU/FPGA 协同处理的系统架构图。

图 5 为本发明 FPGA 实现 8 路归并排序的结构图。

具体实施方式

应该指出，以下详细说明都是示例性的，旨在对本发明提供进一步的说明。除非另有指明，本文使用的所有技术和科学术语具有与本发明所属技术领域的普通技术人员通常理解的含义。

需要注意的是，这里所使用的术语仅是为了描述具体实施方式，而非意图限制根据本发明的示例性实施方式。

在不冲突的情况下，本发明中的实施例及实施例中的特征可以相互组合。

实施例一

本实施例公开了 CPU/FPGA 协同的网络遥测乱序重排方法。

如图 1 所示，CPU/FPGA 协同的网络遥测乱序重排方法，包括以下步骤：

CPU 将带内网络遥测报告拆分为多个待排序包和对应的信息包，构建内存缓冲区和动态指针管理区，将待排序包分别存储至对应的内存缓冲区中，将信息包存储至动态指针管理区中，所述待排序包中包括时间戳和指向其对应信息包的指针；

CPU 将多个待排序包传输至 FPGA 的片外存储资源 DDR 中；

说明书

FPGA 将 DDR 中的多个待排序包调入，并调用对应的多个加速排序内核并行进行待排序包的排序；

FPGA 排序完成后，将排序后的有序包传入 DDR，并将有序包从 DDR 传输至主机端的内存缓冲区；

CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复，得到一条排序完成的时序数据库记录。

CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复，得到一条排序完成的时序数据库记录。

本实施例公开了一种 CPU/FPGA 协同的网络遥测乱序重排方法。在整个流程中，使用 CPU 作为控制层面，负责程序调度和拆分解析，使用 FPGA(Field-Programmable Gate Array, 即现场可编程门阵列)作为数据层面，负责乱序重排任务。

具体包括以下步骤：

步骤 1：CPU 拆分并且分析带内网络遥测报告，构建内存缓冲区和动态指针管理区，提取时间戳及其信息，然后分析节点信息，接着将排序包和信息包分别存放入对应内存缓冲区和动态指针管理区，完成数据预处理。

步骤 2：内存缓冲区中的数据通过 PCIe 被存放至 FPGA 的片外存储资源 DDR 中，为排序做准备。

步骤 3：FPGA 将 DDR 中的数据调入片内存储资源 BRAM 中，接着调用多个加速排序内核。每一个内核负责排序特定遥测节点的排序包，多个内核并行执行排序过程。

步骤 4：FPGA 排序完成后，将 BRAM 中排序后的数据传入 DDR，并且通知 CPU 读取有序数据。排序后的数据通过 PCIe 传输至主机端的内存缓冲区，然后 CPU 读取有序包中指针（指针指向内存缓冲区中的二元组<

说明书

出口利用率, 丢包统计>), 接着对其重组恢复。

步骤 1 包括以下步骤:

步骤 1-1, 主机端创建 n 块内存缓冲区 (n 对应遥测路径中节点个数)。

步骤 1-2, CPU 拆分带内网络遥测报告, 提取 METADATA 中毫秒时间戳以及出口利用率和丢包统计等信息。

步骤 1-3, CPU 构建一块动态指针管理区, 其存放出口利用率和丢包统计等信息, 以缓解 CPU 与 FPGA 之间数据传输的压力并且提高 FPGA 排序的效率。

步骤 1-4, CPU 读取并且分析 METADATA 中 DeviceID 值, 分配待排序包给对应的内存缓冲区。待排序包由毫秒时间戳和信息指针两部分构成, 信息指针指向出口利用率和丢包统计等信息存放在主机端动态指针管理区的地址。

步骤 2 包括以下步骤:

步骤 2-1, CPU 将主机端的 n 个内存缓冲区中的待排序包通过 PCIe 传输至 FPGA 的片外存储资源 DDR 中。

步骤 3 包括以下步骤:

步骤 3-1, 排序内核读取 FPGA 片外存储资源 DDR 中的数据至片内缓存 BRAM 中。

步骤 3-2, 排序内核读取参数, 包括待排序包数组的首地址和待排序包数组的大小。

步骤 3-3, FPGA 启动多个归并排序内核, 多个排序内核并行执行加速排序过程, 一个归并排序内核负责排序一个遥测节点的乱序数据。

步骤 3-4, 加速排序内核开始执行 8 路归并排序算法, 此算法使用 HLS 中 dataflow 指令, 加速了归并排序过程, 实现了归并排序的并行化, 减少了迭代次数, 提高了归并排序效率。

说明书

步骤 4 包括以下步骤：

步骤 4-1, FPGA 将 BRAM 中的数据突发传输至 DDR 中, 并且通知 CPU 取回排序后的遥测数据。

步骤 4-2, 经由 PCIe, DDR 中的有序遥测数据传输至主机端的内存缓冲区中。

步骤 4-3, CPU 分析有序包中指针, 读取指向动态指针管理区中的地址, 找寻动态指针管理区中时间戳对应的出口利用率和丢包统计等信息, 将时间戳与其对应的出口利用率和丢包统计重组组合, 构成一条时序数据库的记录。

更为详细的: 本实施例首先对遥测数据进行预处理, 创建了内存缓冲区和动态指针管理区, 其中内存缓冲区用来存放时间戳和其对应信息的指针, 动态指针管理区用来存放时间戳对应信息的具体数据。

如图 2 所示, 一份遥测数据报中存在多条 METADATA, 一条 METADATA 对应遥测路径上的一个节点的具体信息。由于不同遥测路径可能存在节点重叠, 所以顺序到达的遥测数据报会出现乱序的问题。METADATA 中存在序列信息 (DeviceID, EgressPortRxDropCount, RxTimeStampSeconds, EgressPortUtilization), 其中 RxTimeStampSeconds 属性对应值会出现乱序到达的问题。

例如: 对于 Device ID 等于 3 的两份 METADATA, 图中先到达的遥测数据报中的毫秒时间戳却晚于后到达的遥测数据报中的毫秒时间戳。CPU 将 EgressPortRxDropCount 与 EgressPortUtilization 组合为一个信息包 (EgressPortRxDropCount, EgressPortUtilization), 并且放入动态指针管理区中; 接着构建一个指针 ptr 且使其指向信息包存放在动态指针管理区中的地址, 然后 CPU 分析序列信息中的 DeviceID 值, 根据 DeviceID 值将排序包 (RxTimeStampSeconds, ptr) 放入对应的内存缓冲区中。

说明书

如图 3 所示, 遥测服务器接收到遥测报告后, 接着 CPU 读取 INT Header, 获取到 METADATA 的数量。CPU 拆分 METADATA 0, 首先取出 EgressPortRxDropCount(丢包统计) 与 EgressPortUtilization(出口利用率), 接着将其组合为信息包 (EgressPortRxDropCount , EgressPortUtilization) 并放入动态指针管理区, 然后创建指针 ptr 指向信息包在动态指针管理区中的地址, 然后分析序列信息中 DeviceID 值, 将排序包(RxTimeStampSeconds, ptr) 放入内存缓冲区 1; 接着 CPU 拆分 METADATA 1, 重复上述内存缓冲区和动态指针管理区相关操作, 分析 DeviceID 值, 将排序包放入内存缓冲区 2; 然后 CPU 继续拆分下一个 METADATA, 直至所有的 METADATA 拆分分析完毕。需要说明的是, 遥测服务器下达遥测任务后, 会接收到多个 INT Header。

需要说明的是内存缓冲区中的个数是根据所有遥测路径上节点的总个数设定的, 一个内存缓冲区对应遥测路径的一个具体节点, 用户可以根据节点的个数来进行设置内存缓冲区的个数。在此模式下, CPU 根据不同 DeviceID 值把数据划分到不同的数据块中, 并且后续为这些数据块分配多个加速排序内核进行并行排序, 进一步提高了大数据量的排序速度, 而且这种构造动态指针管理区的方式, 抽离了与重排序过程无关的信息, 缓解了数据传输和 FPGA 存储的压力, 并且提高了 FPGA 重排序的效率。

如图 4 所示, 本实施例公开了一种 CPU/FPGA 协同处理带内网络遥测数据乱序的系统架构。当遥测数据报到达时, 主机端的调度区调用拆分区, 其拆分分析遥测数据报, 将拆分的数据分别放入内存中的内存缓冲区和动态指针管理区中, 放置在内存缓冲区中的数据通过 PCIe(peripheral component interconnect express, 高速串行计算机扩展总线标准)传输至 FPGA 的 DDR 中, 接着 FPGA 启动多个加速排序内核, 加速排序内核会获取 CPU 发送的控制参数; 其中, 所述控制参数包括开始信号、DDR 的偏移

说明书

地址和数据量大小，然后加速排序内核加载 DDR 中待排序数据至片内高速缓存 BRAM 中，一个加速排序内核处理遥测路径中相应节点的乱序数据。

需要说明的是，多个加速排序内核之间数据无关联相互独立，并且 CPU 为多个加速排序内核分配不同的 DDR 偏移地址，多个加速排序内核可以并行读取数据、排序数据、返回数据，从而实现了 FPGA 的并行化，加快了整体排序速率，提高了 FPGA 的排序效率。

此外，通过 PCIe，数据可实现在主机端内存与 DDR 内存之间的数据零拷贝，并不会因数据传输而性能受影响。

每一个加速排序内核使用 8 路归并排序算法，此算法使用 HLS 语言中 dataflow 指令，实现了数据流操作，改善了吞吐量，降低了延迟，提高了排序效率：

如图 5 所示，8 路归并排序创建了 8 个输入数据流，8 个输入数据流并行执行第一阶段归并操作（两两归并），归并后的结果写入结果数据流中，结果数据流会作为第二阶段归并的输入数据流；

FPGA 检查第二阶段 4 个输入数据流，若某个输入数据流满，则通知输入数据流(第一阶段的输出数据流)对应第一阶段的归并操作暂停;若某个归并操作的两个输入数据流全空，则通知输入数据流对应第二阶段的归并操作暂停;若某个归并操作的一个输入数据流空，另一个输入数据流非空，则此归并可以继续执行；

当执行第三阶段的归并操作时，FPGA 将提前检查第三阶段的 2 个输入数据流，若输入数据流满，则通知第二阶段相应的归并操作暂停，直至上述输入数据流有缓存空间，若两个输入数据流全空，则需检查第一阶段与第二阶段的归并操作状态，若所有的归并操作均暂停，则此轮排序结束。

需要说明的是:乱序数据需要经过多轮排序才可以完全有序，每一轮后数据可以达到部分有序，所有阶段的所有归并操作可以并行执行(在符合上述归并操作执行的前提下)。经过一次 8 路归并排序，n 个乱序数据可以被

说明书

排序为 $n/8$ 个有序序列。N 个乱序数据经过 $\log_8 n$ 次排序即可完成排序。一般来说，排序数据的规模会设置为 8 的倍数。

下面我们将以实例说明。

假设我们所需排序的乱序数据为 1 至 64 的倒序数据。64 个乱序数据中的前 8 位 (64, 63, 62, 61, 60, 59, 58, 57) 分别对应进入 8 个数据流中 (数据流 1-1, 数据流 1-2,, 数据流 1-8), 8 个数据流同时进行归并操作, 64 与 63 归并, 62 与 61 归并, 60 与 59 归并, 58 与 57 归并, 上述 4 个归并操作是同时进行的, 此时结果数据流 (2-1, 2-2, 2-3, 2-4) 分别为 ({63, 64}, {61, 62}, {59, 60}, {57, 58}) 这些结果数据流会作为第二阶段的输入数据流;

接着进行第二阶段的归并操作, 2 个归并操作并行执行, 数据流 2-1 与数据流 2-2 归并, 数据流 2-3 与数据流 2-4 进行归并, 此时结果数据流 (3-1, 3-2) 分别为 ({61}, {57}), 此时数据流 (2-1, 2-2, 2-3, 2-4) 的状态分别为 ({63, 64}, {62}, {59, 60}, {58});

然后进行第三阶段的归并操作, 结果数据流接收到结果 57, 同时第二阶段进行归并操作, 第二阶段的 2 个归并操作是并行执行的, 结果数据流 (3-1, 3-2) 分别为 ({61, 62}, {58}), 此时数据流 (2-1, 2-2, 2-3, 2-4) 的状态分别为 ({63, 64}, {}, {59, 60}, {}), 在这个时段中, 第二阶段与第三阶段共 5 个归并操作可以并行执行; FPGA 再次并行执行第二阶段与第三阶段的归并操作, 多次执行上述操作后, 结果数据流为 (57, 58, 59, 60, 61, 62, 63, 64), 完成了第一轮排序的分排序。

接着乱序数据中的下 8 位分别对应进入至 8 个数据流中, 数据流 (数据流 1-1, 数据流 1-2,, 数据流 1-8) 状态分别为 ({56}, {55}, {54}, {53}, {52}, {51}, {50}, {49}), 再次执行上述过程, 此时的结果数据流为 (57, 58, 59, 60, 61, 62, 63, 64, 49, 50, 51, 52, 53, 54, 55, 56)。再经过此 6 次分排序后, 结果数据流为 (57, 58, 59, 60, 61, 62,

说明书

63, 64, 49, 50, 51, 52, 53, 54, 55, 56,, 1, 2, 3, 4, 5, 6, 7, 8), 此结果数据流为局部有序的(前 8 位有序, 8-16 位有序,, 最后 8 位有序)。

接着 FPGA 进行第二轮排序, 8 个数据流(数据流 1-1, 数据流 1-2,, 数据流 1-8)分别对应读取第一轮结果数据流的相应局部有序数据(前 8 位, 8-16 位,, 最后 8 位), 此时 8 个数据流(数据流 1-1, 数据流 1-2,, 数据流 1-8)的状态分别为 ({57, 58, 59, 60, 61, 62, 63, 64}, {49, 50, 51, 52, 53, 54, 55, 56},, {1, 2, 3, 4, 5, 6, 7, 8})。

FPGA 执行第二轮排序的第一阶段归并排序, 4 个归并操作并行执行, 结果数据流 (2-1, 2-2, 2-3, 2-4) 分别为 ({49}, {33}, {17}, {1})。

紧接着 FPGA 执行第二阶段归并排序, 49 与 33 归并, 17 与 1 归并, 结果数据流 (3-1, 3-2) 分别为 ({33}, {1}), 此瞬间结果数据流 (2-1, 2-2, 2-3, 2-4) 分别为 ({49}, {}, {17}, {}), 同时第一阶段的所有归并排序并行执行, 第二阶段的输入数据流 (2-1, 2-2, 2-3, 2-4) 分别为 ({49, 50}, {34}, {17, 18}, {2}), 当程序运行至此状态时, 则会出现三个阶段的归并排序操作并行的状态(即 7 个归并操作并行): 对于第三阶段的归并, 数据流 3-1 与 3-2 归并, 结果数据流得到 1, 结果数据流 (3-1, 3-2) 的此瞬间为 ({33}, {}); 对于第二阶段的归并, 数据流 2-1 与数据流 2-2 归并, 数据流 2-3 与数据流 2-4 归并, 结果数据流 (3-1, 3-2) 分别为 ({33, 34}, {2}); 对于第一阶段的归并, 数据流 1-1 与数据流 1-2 归并, 数据流 1-3 与数据流 1-4 归并,, 数据流 1-7 与数据流 1-8 归并, 结果数据流 (2-1, 2-2, 2-3, 2-4) 分别为 ({49, 50}, {35}, {17, 18, 19}, {3}), 输入数据流 (数据流 1-1, 数据流 1-2,, 数据流 1-8) 分别为 ({57, 58, 59, 60, 61, 62, 63, 64}, {51, 52, 53, 54, 55, 56},, {4, 5, 6, 7, 8})。

FPGA 接着并行三个阶段中的归并操作(前提需要满足上述归并条件),

说明书

当除所有阶段的输入流（输入流 1-1，.....，输入流 3-1，输入流 3-2）都空时，此排序结束，最终乱序数据被排序为完全有序数据（1， 2， 3， 4， 5， 6， 7， 8， 9， 10，, 61， 62， 63， 64）。

需要说明的是，若数据规模特别大，若我们设置的数据流空间无法一次性装载下整个有序序列，则可以分批次读取有序序列进行排序。例如：数据规模为 32768，最后一次归并排序为将 8 个 4096 的有序序列进行归并操作，但 8 个数据流（数据流 1-1，数据流 1-2，.....，数据流 1-8）设置的缓存空间都为 512，因此每个数据流读取对应有序序列的前 512 位，当数据流空间空闲时，再读剩下的数据。

例如：8 个数据流都分别读取对应有序序列的前 512 位，随着排序的进行，数据流 1-2 的数据流的缓存空间空闲，则数据流 1-2 则会读取对应有序序列的下 512 位，直至对应有序序列的 4096 个数据全部读取完毕，其他数据流也遵守此规定。

排序完成后，FPGA 把 BRAM 中的数据运输至 DDR 中，并且向 CPU 传输排序完成信号；CPU 接收到排序完成信号后，利用 PCIe 从 DDR 中读取已排序数据至内存缓冲区。接着主机端的调度区调用重组区进行重组恢复，重组区读取内存缓冲区中（RxTimeStampSeconds， ptr），由 ptr 找到动态指针管理区中的信息包（EgressPortRxDropCount， EgressPortUtilization），重组得到记录（ RxTimeStampSeconds ， EgressPortUtilization ， EgressPortRxDropCount）。

需要说明的是，数据中心可以根据遥测路径中节点的个数来确定加速排序内核的数量，例如遥测路径中包含 8 个节点，则数据中心则会选择调用 8 个加速排序内核来对此乱序重排过程进行加速。这种 CPU/FPGA 协同处理的系统架构，加速了排序过程，提高了整体性能，并且由于排序过程不在 CPU 内进行，还降低了主机端 CPU 和内存的负载。

实施例二

说明书

本实施例公开了 CPU/FPGA 协同的网络遥测乱序重排系统。

CPU/FPGA 协同的网络遥测乱序重排系统，包括：

预处理模块，被配置为：CPU 将带内网络遥测报告拆分为多个待排序包和对应的信息包，构建内存缓冲区和动态指针管理区，将待排序包分别存储至对应的内存缓冲区中，将信息包存储至动态指针管理区中，所述待排序包中包括时间戳和指向其对应信息包的指针；

传输模块，被配置为：CPU 将多个待排序包传输至 FPGA 的片外存储资源 DDR 中；

并行排序模块，被配置为：FPGA 将 DDR 中的多个待排序包调入，并调用对应的多个加速排序内核并行进行待排序包的排序；

回传模块，被配置为：FPGA 排序完成后，将排序后的有序包传入 DDR，并将有序包从 DDR 传输至主机端的内存缓冲区；

重组模块，被配置为：CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复，得到一条排序完成的时序数据库记录。

实施例三

本实施例的目的是提供计算机可读存储介质。

计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现如本公开实施例 1 所述的 CPU/FPGA 协同的网络遥测乱序重排方法中的步骤。

实施例四

本实施例的目的是提供电子设备。

电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的程序，所述处理器执行所述程序时实现如本公开实施例 1 所述的 CPU/FPGA 协同的网络遥测乱序重排方法中的步骤。

说明书

以上实施例二、三和四的装置中涉及的各步骤与方法实施例一相对应，具体实施方式可参见实施例一的相关说明部分。术语“计算机可读存储介质”应该理解为包括一个或多个指令集的单个介质或多个介质；还应当被理解为包括任何介质，所述任何介质能够存储、编码或承载用于由处理器执行的指令集并使处理器执行本发明中的任一方法。

本领域技术人员应该明白，上述本发明的各模块或各步骤可以用通用的计算机装置来实现，可选地，它们可以用计算装置可执行的程序代码来实现，从而，可以将它们存储在存储装置中由计算装置来执行，或者将它们分别制作成各个集成电路模块，或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。本发明不限制于任何特定的硬件和软件的结合。

上述虽然结合附图对本发明的具体实施方式进行了描述，但并非对本发明保护范围的限制，所属领域技术人员应该明白，在本发明的技术方案的基础上，本领域技术人员不需要付出创造性劳动即可做出的各种修改或变形仍在本发明的保护范围以内。

说明书附图

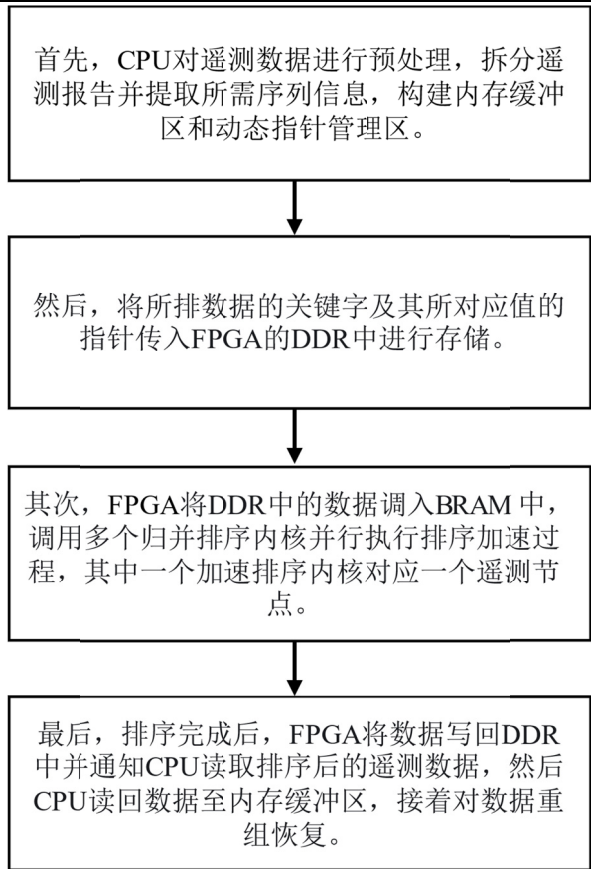


图 1

INT Header	INT Header
####[METADATA 0]#### DeviceID: 1 EgressPortRxDropCount: 1 RxTimeStampSeconds : 100100040125 RxTimeNanoStampSecondsUpper : 3077 EgressPortUtilization : 1	####[METADATA 0]#### DeviceID: 2 EgressPortRxDropCount: 3 RxTimeStampSeconds : 1700100040025 RxTimeNanoStampSecondsUpper : 3077 EgressPortUtilization : 0
####[METADATA 1]#### DeviceID: 3 EgressPortRxDropCount: 1 RxTimeStampSeconds : 1700100040225 RxTimeNanoStampSecondsUpper : 3077 EgressPortUtilization : 0	####[METADATA 1]#### DeviceID: 3 EgressPortRxDropCount: 2 RxTimeStampSeconds : 1700100040125 RxTimeNanoStampSecondsUpper : 3077 EgressPortUtilization : 0
####[METADATA 2]#### DeviceID: 5 EgressPortRxDropCount: 3 RxTimeStampSeconds : 1700100040725 RxTimeNanoStampSecondsUpper : 3077 EgressPortUtilization : 1	####[METADATA 2]#### DeviceID: 5 EgressPortRxDropCount: 2 RxTimeStampSeconds : 1700100040525 RxTimeNanoStampSecondsUpper : 3077 EgressPortUtilization : 0

(1) 先到达的遥测数据包

(2) 后到达的遥测数据包

图 2

说明书附图

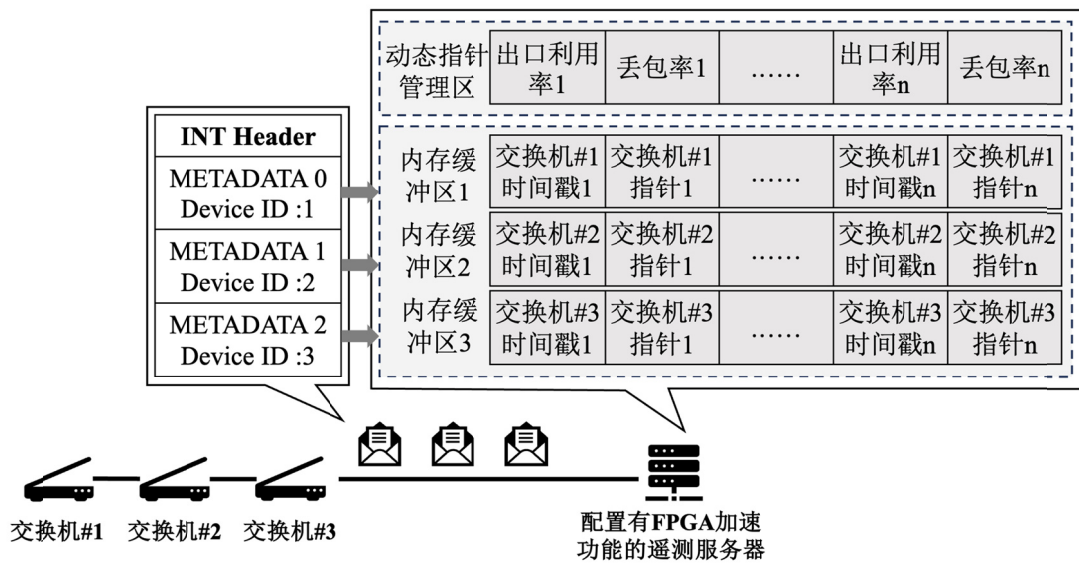


图 3

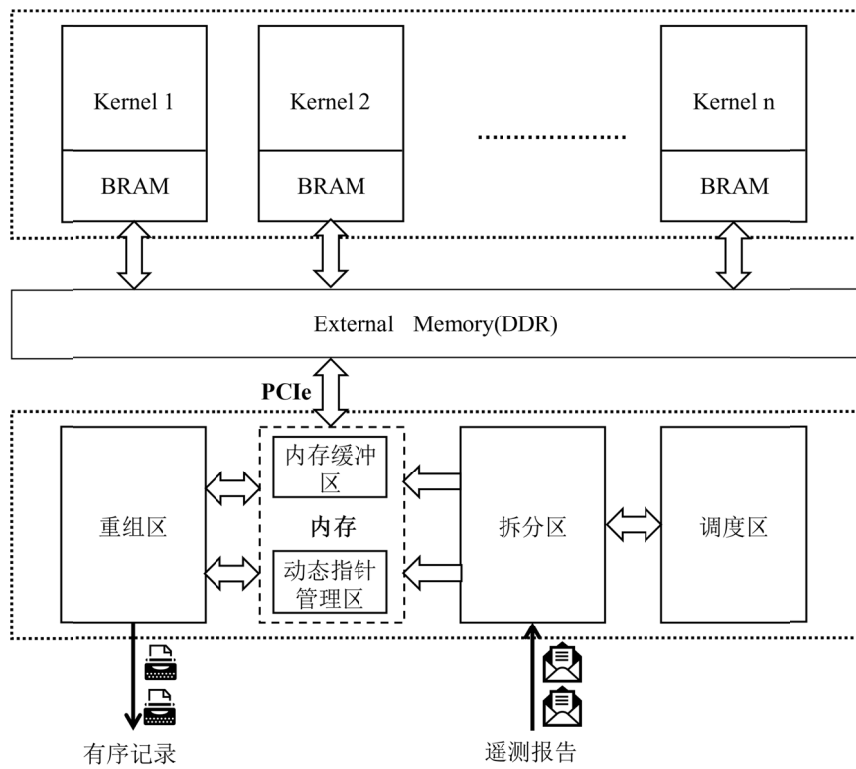


图 4

说明书附图

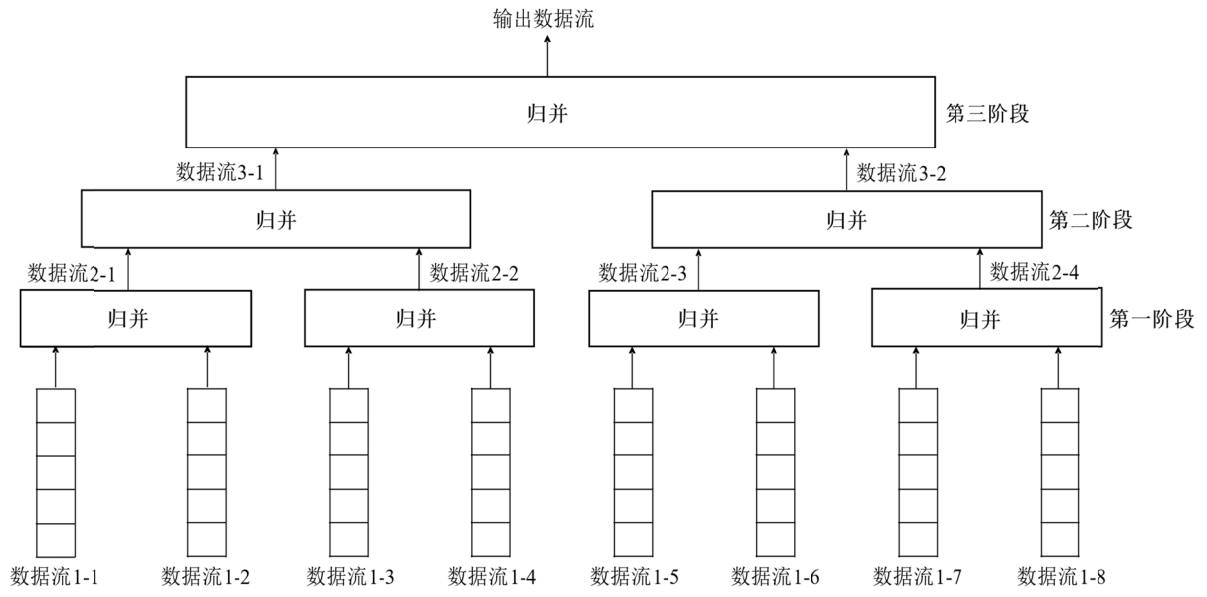


图 5

说明书摘要

本发明提出 CPU/FPGA 协同的网络遥测乱序重排方法及系统，涉及网络遥测技术领域。包括 CPU 将带内网络遥测报告拆分为待排序包和信息包，构建内存缓冲区和动态指针管理区，将待排序包存储至对应的内存缓冲区中，将信息包存储至动态指针管理区中；CPU 将多个待排序包传输至 FPGA 的片外存储资源 DDR 中；FPGA 调用多个加速排序内核并行进行待排序包的排序；FPGA 排序完成后，将排序后的有序包传入 DDR；CPU 读取有序包，并从动态指针管理区中提取出有序包对应的信息包，进行重组恢复。本发明 CPU 和 FPGA 协同合作，提升了处理遥测数据的速度，降低了 CPU 的占用率和处理延迟。