HPmon

Generated by Doxygen 1.8.14

Contents

Chapter 1

README

HPC and Scalable Monitor

This monitor processes different files in teh system to obtain updated information about the status of the system including:

- IP_Addr: Host ip.
- Mem: Total dynamic memory in GigaBytes.
- MemUsage: Percentage of dynamic memory used.
- NCPU: Number of CPUs per node.
- NCores: Number of cores per node.
- CPUBusy: percentage of cpu idle in the sampling interval.
- w: percentage of writes related to IO traffic in the sampling interval.
- tIO: Percentage of IO time in the sampling interval.
- Speed: Speed of the interface in Giga bits per second.
- NetUsage: Percentage of interface used in the sampling interval.
- CurrentTemp: Current temperature of the node on degrees Celsius.
- Temp(%): Percentage of temperature reached based on the maximum temperature.

Given in the following format:

```
IP_Addr Mem(GB) MemUsage(%) NCPU NCores CPUBusy(%) <for each device>(w(%) tIO(%)) <for each
       interface>(Speed(Gb/s) NetUsate(%)) <for each coretemp>CurrentTemp Temp(%)
```

Input Parameters and usage

```
./DaeMon -i <time interval (ms)> -s <samples to send packed (1 - 255)> -t <threshold 0-100> -p <number port
       of master server (1024 <= port <= 65535)> -a <address or name of master server>
```

- Time interval: Number of miliseconds in which the node will be sampled.
- Samples to be sent: Number of samples that will be included in the packet.
-

Packet composition

- 0-3 : IP-Address
- 4 : Memory GigaBytes
- 5 : Number of CPUs
- 6 : Number of cores per CPU
- 7 : Number of devices
- 8 : Number of interfaces
- 9 : Number of coretemps
- 10 : Number of GPUs
- 11 : Number of samples packed
- 12 : Memory usage (%)
- 13 : CPU Busy (%)
  - FOR EACH CPU, ENERGY SPENT IN THE INTERVAL
    - x : Energy in joules since computer starts.
  - FOR EACH DEVICE
    - x : Percentage of time doing write tasks
    - y : Percentage of time in the sampling interval doing I/O
  - FOR EACH NET INTERFACE
    - x : Speed of the interface in GigaBits per second
    - y : Percentage of net usage in the sampling interval
  - FOR EACH CORETEMP
    - x : Current temperatue
    - y : Percentage of current temperature over critical temperature
  - FOR EACH GPU
    - a : CUDA compatible GPU (1 compatible, -1 no comptable)!!!!!!!
    - b : Memory of GPU usage (%)
    - c : Gpu usage (%)
    - d : Temperature of GPU on degrees Celsius
    - e : Power of GPU on Watts

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 cpu_dev Struct Reference

Public Attributes

- string **model_name**
- Cpu_stats **stats** [2]

The documentation for this struct was generated from the following file:

- include/cpu_info.hpp

3.2 cpu_stats Struct Reference

Public Attributes

- unsigned long long **user**
- unsigned long long **nice**
- unsigned long long **sys**
- unsigned long long **idle**
- unsigned long long **iowait**
- unsigned long long **hardirq**
- unsigned long long **softirq**
- unsigned long long **steal**
- unsigned long long **guest**
- unsigned long long **guest_nice**

The documentation for this struct was generated from the following file:

- include/cpu_info.hpp

3.3 devices_stats Struct Reference

Public Attributes

- unsigned long **rd_ios**
- unsigned long **rd_merges**
- unsigned long **rd_sec**
- unsigned long int **rd_ticks**
- unsigned long **wr_ios**
- unsigned long **wr_merges**
- unsigned long **wr_sec**
- unsigned int **wr_ticks**
- unsigned int **ios_pgr**
- unsigned int **tot_ticks**
- unsigned int **rq_ticks**

The documentation for this struct was generated from the following file:

- include/devices_info.hpp

3.4 Dir_path Class Reference

Public Member Functions

- **Dir_path** (std::string a, std::vector< std::string > &b)
- std::vector< std::string > **read_directory** (const std::string &path)

Public Attributes

- string **path**
- std::vector< std::string > **dfiles**

The documentation for this class was generated from the following files:

- include/dir_path.hpp
- src/dir_path.cpp

3.5 Gpu_accu Struct Reference

Public Attributes

- unsigned char **memUsage**
- unsigned char **gpuUsage**
- unsigned char **temperature**
- unsigned char **powerUsage**

The documentation for this struct was generated from the following file:

- include/gpu_info.hpp

3.6 Gpu_dev Struct Reference

Public Attributes

- int **cudacomp**
- string **model_name**
- size_t **memTotal**
- size_t **memFree**
- size_t **memUsage**
- unsigned int **gpuUsage**
- unsigned int **temperature**
- unsigned int **powerUsage**
- int **dev_id**
- unsigned char **capability**

The documentation for this struct was generated from the following file:

- include/gpu_info.hpp

3.7 handle_args Struct Reference

Public Attributes

- char **client_IP** [30]
- unsigned char * **buffer**
- ssize_t **size**

The documentation for this struct was generated from the following file:

- src/servidor_monitor.cpp

3.8 hw_conf Struct Reference

Public Attributes

- string **ip_addr_s**
- string **mac_addr**
- string **hostname**
- int **mem_total**
- int **n_devices_io**
- int **n_cpu**
- int **n_cores**
- int **n_gpu**
- int **n_interfaces**
- vector< Net_dev > **net_interfaces**
- vector< Cpu_dev > **cpus**
- int **path_dir**
- int **GPU_DEVICES_COMPATIBLE**

- vector< Gpu_dev > **gpus**
- void * **cuLib**
- void * **nvmlLib**
- vector< IO_dev > **io_dev**
- vector< Temp_features > **temp_features**
- int **n_core_temps**
- vector< std::string > **vcore_path**
- vector< std::string > **files_input**
- vector< std::string > **files_max**
- double **max_temp**
- vector< Power_cpu > **pwcpu_features**

The documentation for this struct was generated from the following file:

- include/system_features.hpp

3.9 io_dev Struct Reference

Public Attributes

- std::string **dev_name**
- Devices_stats **stats** [2]

The documentation for this struct was generated from the following file:

- include/devices_info.hpp

3.10 net_dev Struct Reference

Public Attributes

- std::string **net_name**
- unsigned int **speed**
- Net_stats **stats** [2]

The documentation for this struct was generated from the following file:

- include/net_info.hpp

3.11 net_stats Struct Reference

Public Attributes

- unsigned long long **rx_bytes**
- unsigned long long **tx_bytes**

The documentation for this struct was generated from the following file:

- include/net_info.hpp

3.12 Packed_sample Class Reference

Public Member Functions

- **Packed_sample** (Hw_conf hwconf, unsigned int interval, int n_samples, int threshold)
- void **set_n_samples** (int n_samples)
- void **set_n_devices_io** (int n_devices_io)
- void **set_n_interfaces** (int n_interfaces)
- void **pack_features** ()
- void pack_monitoring (int mem_usa, int cpu_usa, int ∗devices_usa[2], int ∗net_usa, int ∗temp)
- void pack_sample_s (std::string sample)
- vector< string > split_log (std::string str)
- vector< string > parse_log (std::string sample)
- void calculate_sample_size ()
- int calculate_bit_maps_bytes ()
- void to_bit_map ()
- void codificar_cuartil (int cuartil)

Public Attributes

- unsigned char **packed_buffer** [SAMPLES_BUFFER_SIZE]
- unsigned char **n_samples**
- unsigned int **sample_size**
- unsigned int **interval**
- int **memtotal**
- int **samples_packed**
- int **packed_ptr**
- int **packed_bytes**
- int **sample_pt**
- int **n_devices_io**
- int **n_interfaces**
- int **threshold**
- int **n_core_temps**
- int **n_cpu**
- int **n_cores**
- int **n_gpus**
- time_t **time_sample**
- std::string **ip_addr_s**
- std::vector< std::string > **ip_v**
- unsigned char **bit_map** [ ]
- int **datos_cuartil**
- int **bytes_bitmap**
- int **bit_map_ptr**
- int **byte_bit_map**
- unsigned char **memory_avg**
- unsigned char **cpu_avg**
- vector< unsigned char > **pw_cpu**
- vector< unsigned char > **dev_avg**
- vector< unsigned char > **net_avg**
- vector< unsigned char > **temp_avg**
- vector< unsigned char > **gpu_avg**
- vector< Gpu_accu > **gpu_accu**

3.12.1 Member Function Documentation

3.12.1.1 calculate_bit_maps_bytes()

```
int Packed_sample::calculate_bit_maps_bytes ( )
```

Calculate size of bit map on bytes

Returns

Size of bit map on bytes

n_cpu = Number of cpus mem_cpu = n_interfaces = number of interfaces n_devices_io = number of I/O devices n_core_temps = number of coretemps

n_gpus = number of gpus for each gpu 4 bytes: memoryUsage, gpuUsage, temperature and powerUsage

3.12.1.2 calculate_sample_size()

```
void Packed_sample::calculate_sample_size ( )
```

Calculate the size of a sample. n_cpu = Number of cpus mem_cpu = n_interfaces = number of interfaces n_↩ devices_io = number of I/O devices n_core_temps = number of coretemps

n_gpus = number of gpus for each gpu 4 bytes: memoryUsage, gpuUsage, temperature and powerUsage

3.12.1.3 codificar_cuartil()

```
void Packed_sample::codificar_cuartil (
            int cuartil )
```

Encodes the value passed by parameters in the bit map

Parameters

in	*cuartil*	number to be coded

3.12.1.4 pack_monitoring()

```
void Packed_sample::pack_monitoring (
            int mem_usa,
            int cpu_usa,
            int * devices_usa[2],
```

```
            int * net_usa,
            int * temp )
```

Pack sample into buffer to be sent.

Parameters

in	*sample*	String to be packed in the buffer.

3.12.1.5 pack_sample_s()

```
void Packed_sample::pack_sample_s (
            std::string sample )
```

Pack sample into buffer to be sent.

Parameters

in	*sample*	String to be packed in the buffer.

3.12.1.6 parse_log()

```
vector< string > Packed_sample::parse_log (
            std::string sample )
```

Parse sample to include on the buffer

Parameters

in	*sample*	String including sample to be parsed.

Returns

Vector of strings including the parsed contents.

3.12.1.7 split_log()

```
vector< string > Packed_sample::split_log (
            std::string str )
```

Splits the log

Parameters

in	*str*	String representing the log content.

Returns

Vector of strings with the log splitted

3.12.1.8 to_bit_map()

```
void Packed_sample::to_bit_map ( )
```

Calculate size of bit map on bytes

Returns

Size of bit map on bytes

The documentation for this class was generated from the following files:

- include/Packed_sample.hpp
- src/Packed_sample.cpp

3.13 packet Struct Reference

Public Attributes

- char **title** [256]
- char **xlabel** [256]
- char **ylabel** [256]
- char **style** [8]
- int **filesize**

The documentation for this struct was generated from the following file:

- include/cliente_monitor.hpp

3.14 power_cpu Struct Reference

Public Attributes

- std::string **power_path**
- std::string **energy_path**
- unsigned long long **energy_value**
- double **diff_energy_value**

The documentation for this struct was generated from the following file:

- include/power_cpu_info.hpp

3.15 req_inf Struct Reference

Public Attributes

- int **socket**
- char **address** [16]
- int **port**

The documentation for this struct was generated from the following file:

- include/servidor_monitor.hpp

3.16 temp_features Struct Reference

Public Attributes

- std::string **temp_path**
- std::vector< std::string > **files_input**
- std::vector< std::string > **files_max**
- double **max_temp**

The documentation for this struct was generated from the following file:

- include/temp_info.hpp

Index