EAR: Energy management framework for HPC

EAR software is a management framework optimizing the energy and efficiency of a cluster of interconnected nodes. To improve the energy of the cluster, EAR provides energy control, accounting, monitoring and optimization of both the applications running on the cluster and of the overall global cluster.

At EAR's core is a monitoring tool which gathers data on the nodes and on the applications running on the cluster. Therefore, on top of optimizing the energy consumed by the applications running on the cluster and the overall global cluster, EAR reports system and application information.

The system information collected by EAR (called the system signature) reports the performance of the major components of each node (cpu, memory). It is used to optimize the cluster energy but also to report which components are not working to the level expected like, for example, the memory DIMMs in a node are not providing the bandwidth they should, or a cpu in a node is not running at the expected frequency. Therefore, EAR makes sure the performance efficiency of the cluster is kept to its maximum.

The application information collected by EAR (called the application signature) reports basic performance metrics of the application. It is used to optimize the application energy but it can also be used to determine that one application could perform better due to, for example, a high memory bandwidth or a low percentage of AVX512 instruction. Therefore, EAR helps to a better utilization of the system.

EAR is also robust and reliable as it is operational on SuperMUC-NG at Leibniz Supercomputing Centre (LRZ) in Garching near Munich since August 8 2019 (https://doku.lrz.de/display/PUBLIC/SuperMUC-NG) running on a cluster of 6480 nodes helping LRZ to meet its energy goals. At LRZ, EAR is transparently used through a SLURM plugin (https://doku.lrz.de/display/PUBLIC/Energy+Aware+Runtime).

EAR now has extensible offering of plugins for policies, power models, application tracing, and node energy readings.

With the latest version, EAR has reported energy savings of more than 10% on average across a set of applications evaluated in collaboration with SURF Open Innovation Lab (SOIL). An evaluation on a broader set of applications is under way.

EAR components are the EAR library (EARL), EAR DB manager (EARDBD), EAR Daemon (EARD), EAR slurm plugin (EARplug) and EAR Global Manager (EARGM). EAR offers a highly configurable and extensible infrastructure for energy management. Last version of EAR includes a plugin mechanism to dynamically load power policies, power and time models, energy readings and application traces generation. In order to offer a simple install&test approach, EAR includes default powerful plugins for all these features.

Energy accounting and power monitoring is provided by EARplug and EARD. EARD is a linux service running with privileges in computing nodes. This service is continuously monitoring power and other relevant nodes metrics such as temperature and average frequency and reporting them to the DB through EARDBD. Moreover, given EARD is continuously monitoring node metrics, hardware reliability tests are

periodically done. When a HW problem is detected, apart from trying to fix it, a notification is sent to syslogs and DB. These events could be automatically triggered by the system for an immediate reaction and/or analyzed. EARD uses an energy plugin to provide energy readings. By default, plugins for the Intel NodeManager and Lenovo SD650 used at LRZ are provided using the openipmi driver as well as a node energy estimation based on RAPL counters. This last plugin is practical to allow EAR to be installed and tested on any system with Intel CPUs.

EARplug contacts with EARD at application start/end to offer application accounting. This information is reported associated with the job id, being possible to join the EAR information with the one reported by the scheduler for later evaluation and analysis. The current EAR version only provides EARplug for SLURM.

EARL is a dynamic, transparent, and lightweight runtime library that optimizes and controls the energy consumed by mpi jobs without any application modification or user input. EARL guarantees the optimal utilization of system energy following system admin configuration. EARL includes by default two power policies: min_time_to_solution and min_energy_to_solution. Min_time_to_solution target is to boost the frequency of applications which performance scales with frequency according to a scaling factor defined by the sysadmin, while running the others at a default lower frequency and therefore reducing the energy of these applications without performance penalty Min_energy_to_solution targets to save energy by reducing the frequency up to a maximum performance degradation defined by the sysadmin. EARL dynamically identifies repetitive regions in parallel applications (outer loops) without adding any annotation or user input. The algorithm in charge of detecting these regions is called DynAIS. DynAIS is an innovative multi-level algorithm with very low overhead. EARL internals are DynAIS driven, being able to evaluate EARL decisions, one of the key differences between EARL and other solutions. Thanks to DynAIS, EARL dynamically computes the Application Signature, a very reduced set of metrics that characterize application behaviour: Cycles per Instructions (CPI), memory Transactions per Instruction (TPI), Time and Power, Percentage of SSE/AVX/AVX512 instructions. The Application Signature together with the System Signature are the inputs for the default power and time model used by EARL. The power policy API is executed at different points of the lifecycle of an application such as application start/end, loop start/end/per-iteration etc. In that way, EARL can be extended to offer new policies or simply variation of the already provided to better fit in data center requirements and workloads. Power model plugin allows the evaluation of new models and/or approaches for power and time projections such as neuronal networks without reinstalling EAR. These two plugins not only avoid having to reinstall EAR when a new policy or model wants to be considered, they are really very useful for research since specific policies or models can be enabled for specific users without affecting the system security. Metrics collected during application execution can be reported to the EAR DB for a post-mortem analysis.

EARL is compatible with the utilization of other instrumentation libraries such as Extrae (https://tools.bsc.es/extrae). A trace plugin for paraver (https://tools.bsc.es/paraver) traces is provided by default but additional plugins can be developed.

EAR configuration allows the sysadmin to configure the granularity and frequency of information reported to the DB for a fine grain control of the DB size. For large systems, EARDBD is responsible of guaranteeing system scalability.

EARGM monitors and controls the global energy and power consumed in the system. EARGM goal is to guarantee a maximum energy consumption for a given period. It can be configured to work only as a cluster monitoring tool, reporting global status to the DB, or it can be configured to be pro-active and automatically adapt system settings being coordinated with EARL and EARD. Since EARL is aware of application characteristics, it can react to the different EARGM warnings levels based on application characteristics and the energy efficiency measured. EARGM is a lightweight solution for global control since it only centralises the problem detection but specific actions are taken in a distributed way.

EAR components can be partially installed to be adapted to specific requirements. For instance, if only

energy accounting is needed, EARplug and EARD will be only installed. Depending on the cluster size EARDBD can be useful but it is not mandatory. If global monitoring of the cluster is needed, EARGM has to be installed and configured in monitoring model. EARL provides energy optimization, but only EARD and EARPlugin is needed to use it. However, the installation of all the components offers a flexible, extensible and powerful environment for energy management.

The Energy Aware Runtime software has developed in the framework of the BSC-Lenovo collaboration project

Barcelona Supercomputing Center - Centro Nacional de Supercomputación

---