Adjust\_platform(& in\_policy)

위에서의 정책을 기반으로 플랫폼 설정을 조정합니다.

In\_policy : Settings for each control in the policy

In\_policy : 정책에 따른 각 컨트롤에 대한 설정

Aggregate\_sample(& in\_sample, & out\_sample) or + (% agg\_func) 사용안함x

Next level tree로 전송하기 위한 children의 샘플 수집.

In\_sample : 각 자식당의 한 개의 샘플 벡터

Out\_sample : Parent로 전송할 집계 표본값

+

집계 함수의 벡터가 주어지면 일반적으로 샘플의 벡터를 집계합니다. 이 Helper method는 자식 샘플에 대해 집계하는 동안 각 샘플 요소에 다른 집계 함수를 적용합니다.

In\_sample : 각 자녀로부터 받은 샘플 벡터의 자녀에 대한 벡터입니다.

Agg\_func: 적용된 집계 함수의 에이전트 샘플에 대한 벡터입니다.

Out\_sample : 자식 샘플에 집계를 적용하여 생성된 샘플 벡터입니다.

Do\_send\_policy()

새로운 Policy 값을 트리 아래로 에이전트의 자식에게 보내야 하는지 결정하기 위해 컨트롤러에 의해 호출된다.

마지막 호출 이후 정책이 업데이트된 경우 Return값은 true이다.

Do\_send\_sample()

마지막 호출 이후 새 샘플을 읽은 경우 Return값은 true이다.

Do\_write\_batch()

플랫폼 컨트롤을 업데이트 하기위해 write\_batch() 을 호출할지 여부를 결정하기 위해 컨트롤러에서 호출된다.

플랫폼에 조정되면 Return값은 true이고, 그렇지 않은 경우에는 false이다.

Enforce\_policy(&policy)

PlatformIO::write\_control()을 사용하여 정책을 한번 시행한다. 컨트롤러가 없을 때 정적 정책을 시행하기 위해 호출된다.

Init(level, &fan\_in, is\_level\_root)

이 에이전트가 활성화된 레벨을 설정하고 해당 레벨에 대한 Signal/controls를 Push 한다.

Level : 이 에이전트가 활성화된 트리의 레벨. 레벨 0의 에이전트만 sample\_platform(), adjust\_platform()을 실행한다.

Fan\_in : Vector over level 에서 작동하는 각 루트 에이전트에 보고하는 에이전트 수

Is\_level\_root : 에이전트가 레벨의 루트 역할을 하는 경우에 해당. 루트 에이전트만이 ascend(), descend()를 호출한다.

Make\_dictionary(&policy\_names, &sample\_names)

에이전트가 Factory에 등록될 때 에이전트에 대해 올바르게 포맷된 dictionary를 만드는데 사용된다. 구체적인 에이전트 클래스는 policy\_names(), sample\_names() 메소드를 제공하여 이 메소드에 전달할 벡터를 제공 할 수 있다.

Num\_policy(& dictionary)

특정 에이전트에 대해 트리 아래로 전송된 정책 벡터의 값 수를 조회하는데 사용. 관심있는 에이전트에 대해 Agent\_factory().dictionary(agent\_name)에서 리턴한 dictionary으로 호출해야 한다.

Num\_sample(& dictionary)

특정 에이전트에 대해 트리 위로 전송된 샘플의 값 수를 조회하는데 사용. 관심있는 에이전트에 대해 Agent\_factory().dictionary(agent\_name)에서 리턴한 dictionary으로 호출해야 한다.

Policy\_names(& dictionary)

특정 에이전트에 대해 트리 아래로 전송된 정책 벡터의 값 이름을 조회하는데 사용. 관심있는 에이전트에 대해 Agent\_factory().dictionary(agent\_name)에서 리턴한 dictionary으로 호출해야 한다.

Report\_header()

이 에이전트를 사용할 때 보고서 머리글에 추가 될 사용자 정의 필드

Report\_host()

보고서 Host 섹션에 대한 사용자 정의 필드

Report\_region()

보고서 각 지역에 대한 사용자 정의 필드

Sample\_names(& dictionary)

특정 에이전트에 대해 트리 위로 전송된 샘플의 값 이름를 조회하는데 사용. 관심있는 에이전트에 대해 Agent\_factory().dictionary(agent\_name)에서 리턴한 dictionary으로 호출해야 한다.

Sample\_platform(& out\_sample)

플랫폼에서 신호를 읽고 이 신호를 해석/집계 하여 트리로 보낼 수 있는 샘플을 만든다.

Out\_sample : 트리로 전송될 에이전트 특정 샘플 값의 샘플 벡터

Split\_policy(& in\_policy, & out\_policy)

다음 레벨의 하위트리에서 정책을 분할하기 위해 콘트롤러에 의해 호출된다.

In\_policy : 부모의 정책 값

Out\_policy : 각 자식에게 보내질 정책 벡터

Trace\_formats()

추적에 추가된 각 열의 형식 문자열을 리턴한다.

Trace\_names()

추적에 추가할 열 헤더

Trace\_value(& value)

추적에 추가할 최신 값을 얻기위해 콘트롤러에 의해 호출된다.

Validate\_policy(& policy)

들어오는 정책 값을 확인하고 들어오는 정책에서 요청한 기본값을 구성하기 위해 콘트롤러에 의해 호출된다. 정책 발신자는 정책 벡터에서 원하는 오프셋에 ‘NaN’을 사용하여 기본값을 요청할 수 있습니다. 반환된 정책에는 ‘NaN’이 포함되어서는 안되며 descend 및 adjust\_platform에 의해 사용 가능해야 한다.

Policy : 정책은 NaN값을 기본값으로 대채한다. 정책값이 NaN이 아니지만 에이전트에서 해당값이 지원되지 않는 경우 메소드는 geopm::Exception을 통해 GEOPM\_ERROR\_INVALID 값을 출력한다.

Wait()

샘플 기간이 경과할 때까지 컨트롤러에서 호출한다. 컨트롤러 메인 루프의 상태를 제어한다.

Plugin\_name()

해당 플러그인 에이전트의 이름 리턴

Make\_plugin()

해당 플러그인을 정의( geopm::make\_unique() 사용 )

Update\_policy(&in\_policy)

Margin값 수정, freq\_min, freq\_max 수정(m\_perf\_margin, set\_frequency\_bounds() 사용)